From Boltzmann to Diffusion

Statistical Physics of Machine Learning

Tristan Bereau

2026-09-14

Table of contents

Course information

Winter Semester 2026/27. Lecture period: October 12, 2026 to February 6, 2027 (no lectures December 21, 2026 to January 6, 2027)

Lectures: Mon & Wed 9-11; **Tutorial**: Wed 16-18

Lectures: Ph12 gHS; **Tutorial**: Ph12 106

Lecturer: Prof. Dr. Tristan Bereau, Institute for Theoretical Physics, Heidelberg University

Tutors: Sander Hummerich, Institute for Theoretical Physics & Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University; Gerrit Gerhartz, Institute for Theoretical Physics, Heidelberg University

8 credit points

Course links

- HeiCO
- Physik Übungsgruppen

Course registration runs through HeiCO; exercise-group allocation runs through Physik Übungsgruppen; there is no Moodle. Warm-ups, stuck-point cards, and exam admission are all handled in the Physik Übungsgruppen platform.

Course description

Modern generative machine learning is not merely *inspired by* statistical physics — it is, structurally, nineteenth- and twentieth-century statistical mechanics rediscovered. The Boltzmann distribution $p \propto e^{-\beta E}$, the softmax layer $p_i = e^{z_i}/\sum_j e^{z_j}$, and the score $\nabla_x \log p$ that a diffusion model learns are the same equation in three costumes; the third is the first differentiated, which is exactly how the partition function disappears. This course traces that pedigree — the derivation, not the analogy.

A single object runs through the whole course: the partition function Z, the normalizer of an energy-defined probability distribution. You can almost always write a model down; you can almost never normalize it. Z is *raised* as the obstruction that blocks training a Boltzmann machine (Module 1), *approximated* by variational and mean-field methods (Module 2), *sampled around* by Monte Carlo and Langevin dynamics (Module 3), and finally *eliminated* by score matching, where $\nabla_x \log Z = 0$ (Module 4). Along the way, learning becomes the shaping of an energy landscape,

training dynamics become a stochastic process with their own Fokker–Planck equation, and the generative process of a diffusion model is revealed as nonequilibrium thermodynamics run in reverse.

This course is the structural-derivation complement to the companion *Machine Learning and Physics* course: that course teaches the working methods (how to run DDPM, VAEs, MCMC); this one derives *why they work* from statistical mechanics.

Prerequisites

- **Required.** *Theoretical Statistical Physics* or equivalent — ensembles, entropy, the free energy, the Boltzmann distribution, Gaussian integrals, Laplace's method. Every module rests on it continuously.
- **Assumed familiarity.** *Machine Learning and Physics* or equivalent. The course derives why methods you have already run work, and is written for readers who have trained a variational autoencoder and a diffusion model and run a Markov-chain sampler; Week 12 opens on the diffusion loss you minimized there. The derivations are self-contained regardless, so a reader arriving without that background meets these models here for the first time. Mehta et al. (2019) §I–II, the Week 1 tutorial reading, covers the machine-learning vocabulary the early lectures assume.

Assessment

- **Written final exam (100% of the grade).** The exam is built directly from the tutorial problems and tests computational *reasoning*, not live coding. It is closed-book except for **one hand-written aid sheet** that you compile from your own tutorial solutions.
- **Exam admission (ungraded *Studienleistung*, *Zulassungsvoraussetzung*).** Submit the weekly stuck-point card for at least 8 of the 12 tutorial warm-ups (Weeks 2–13; the Week 1 tutorial has no warm-up).

Tutorials run every week (weeks 1–13, plus a week-14 review) and use a productive-failure, peer-instruction, just-in-time format: a light home warm-up whose stuck-point card feeds the in-class triage, then the hard problem is revealed and cracked *in class* — on paper first, and on the computation-led weeks with one laptop per group for a bounded stretch after the prediction has been written down. There is no separate graded problem-set track.

Course outline

The course spans 14 weeks and five modules, organized around the Z spine.

- **Week 1 — Introduction.** Maximum entropy the Boltzmann/Gibbs distribution the partition function Z; the softmax layer *is* the Boltzmann distribution; the 2^N normalization wall.

- **Module 1 (Weeks 2–4) — The energy-based view.** Hopfield networks → Boltzmann machines / RBMs and contrastive divergence → modern energy-based models → modern Hopfield networks attention. *Z is raised as the training obstruction.*
- **Module 2 (Weeks 5–7) — Variational & mean-field methods.** Mean-field theory and the variational free energy → TAP / belief propagation / approximate message passing → Hubbard–Stratonovich transformations and variational autoencoders. *Z is approximated.*
- **Module 3 (Weeks 8–9) — Sampling & the Fokker–Planck of training.** MCMC and Langevin dynamics, simulated annealing → SGD-as-Langevin, flat minima, and entropy-SGD. *Z is sampled around.*
- **Module 4 (Weeks 10–12) — Nonequilibrium thermodynamics of generative modeling.** Score matching → Jarzynski / Crooks work relations and annealed importance sampling → DDPM and score-based SDEs. *Z is eliminated ($\nabla_x \log Z = 0$).*
- **Module 5 (Weeks 13–14) — Information, free energy & bridges.** The information bottleneck → phase transitions in inference → synthesis and exam preparation. *Z becomes the object of study rather than the obstruction.*

Main references

- Mehta, P., Bukov, M., Wang, C.-H., Day, A. G. R., Richardson, C., Fisher, C. K., & Schwab, D. J. (2019). A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810, 1–124. https://doi.org/10.1016/j.physrep.2019.03.001
- MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. https://www.inference.org.uk/mackay/itila/
- Sethna, J. P. (2021). *Statistical Mechanics: Entropy, Order Parameters, and Complexity* (2nd ed.). Oxford University Press.
- Welling, M., Lu, S., & Holdijk, L. (2026). *Generative AI and Stochastic Thermodynamics: A Tale of Free Energies*. Cambridge University Press (forthcoming). https://www.cambridge.org/9781009709064 — companion reading for the back half of the course (Modules 3–5): the free-energy view of generative modeling and stochastic thermodynamics.
- Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4), 620–630. https://doi.org/10.1103/PhysRev.106.620
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554
- Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *JMLR*, 6, 695–709.
- Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *NeurIPS 33*. https://arxiv.org/abs/2006.11239
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *ICLR*. https://arxiv.org/abs/2011.13456

Acknowledgments

These lecture notes, exercises, and supporting code were prepared with the assistance of Claude (Anthropic). All material has been reviewed, edited, and validated by the instructor.

Foreword

These notes are the written companion to a course taught at the blackboard. They exist for the reasons any script does: to correct the errors I will inevitably make while writing at speed, to carry the material for which ninety minutes leaves no room, to give the longer derivations the coherent form a lecture cannot, and to be usable in the week before the exam. They document one path through this material, not the completeness of a textbook, and where the path was a choice I have tried to say so.

The course they accompany is a graduate course in the physics department, and I have written for that reader: someone who has had a master's course in statistical mechanics and is comfortable with ensembles, entropy, the free energy, and Gaussian integrals. In practice a course like this also draws bachelor students who are ahead of themselves, and people from the machine-learning side who arrive with the algorithms but without the thermodynamics. To the first group: the statistical mechanics is genuinely assumed, and Weeks 2 and 11 will be uncomfortable without it. To the second: the derivations here are self-contained, and what you lose by arriving without the companion course *Machine Learning and Physics* is the experience of having already trained the models we take apart. That is a real loss, since much of the pedagogical force of this course comes from explaining why something you have run actually works, but it is not a barrier to following the argument.

The books behind this one

Five books and one review shaped these notes, and it is worth saying what each contributes, since no single one of them covers the course.

Mehta and coauthors' *A high-bias, low-variance introduction to machine learning for physicists* (Mehta et al. 2019) supplies the vocabulary the whole course speaks — the correspondence between energy and loss, between the partition function and the normalizer of a likelihood — and it is the reference text for the tutorials. MacKay's *Information Theory, Inference, and Learning Algorithms* (MacKay 2003) is where I go for entropy and inference approached from the information-theoretic side, and its treatment of the relationship between coding, inference, and free energy is closer to this course's spirit than any physics text. Sethna's *Statistical Mechanics* (Sethna 2021) is the source of the habit of thinking in probabilities rather than in thermodynamic potentials, which is what makes the translation to machine learning read as translation and not as metaphor. Mézard and Montanari's *Information, Physics, and Computation* (Mézard and Montanari 2009) carries the disordered-systems spine of Modules 1 and 2 — mean field, the cavity method, belief propagation — and no machine-learning text I know covers that material at all; the second module would not exist without it. Welling, Lu, and Holdijk's *Generative AI and Stochastic Thermodynamics* (Welling, Lu, and Holdijk 2026) is the closest published match to the

back half of the course, and I recommend it as parallel reading from Module 3 onward: it organizes the same territory around the free energy where I organize it around the partition function, which is the same choice made from the other side of $F = -T \log Z$. Finally, Ulrich Schwarz's Heidelberg script *Theoretical Statistical Physics* (Schwarz 2023) is the structural model for this document, and readers who want their statistical mechanics refreshed to the level assumed here should start with its early chapters.

Guiding principles

Four decisions shaped the course, and knowing them in advance will make the choices in the chapters that follow easier to read.

First, derivation rather than analogy. That statistical mechanics and machine learning resemble each other is remarked upon constantly and demonstrated rarely. This course makes a stronger and narrower claim: in a specific and enumerable set of cases the two fields are manipulating the same equation, and the identity can be proven. Where I can prove it, I do. Where the connection is genuinely a suggestive parallel rather than an identity, I say that instead.

Second, one object tracked through the whole semester. The partition function Z runs through all fourteen weeks. It is raised as the obstruction that blocks training a Boltzmann machine, approximated by variational and mean-field methods, sampled around by Monte Carlo and Langevin dynamics, and finally eliminated by score matching, where $\nabla_x \log Z = 0$. A course organized as a tour of techniques would cover more ground; this organization gives the reader a thread to hold, and it means that Week 10 resolves a difficulty the reader met personally in Week 3.

Third, every method arrives as an explanation of something already run. This course is the structural-derivation complement to *Machine Learning and Physics*: that course teaches how to train a variational autoencoder, a restricted Boltzmann machine, a diffusion model; this one derives why those methods work. Week 12 accordingly opens on the mean-squared error you already minimized and takes it apart into objects the preceding three weeks built.

Fourth, complete derivations in writing and one derivation at the board. The script shows the algebra in full, with annotated multi-line displays. The lecture does one derivation properly — the engine of the day — and states the rest with a pointer to the chapter. The two documents are therefore not interchangeable, and the details live here.

The arc

We begin with the maximum-entropy argument, following Jaynes, and derive the Boltzmann distribution and its normalizer from it; the same lecture proves that the softmax layer terminating essentially every classifier trained today *is* that distribution, and the first week closes on the 2^N wall that makes Z intractable. Module 1 then builds the energy-based view: Hopfield networks and the capacity transition, Boltzmann machines and the intractable gradient, contrastive divergence as the standard evasion, and modern Hopfield networks, whose update rule turns out

to be transformer attention. Module 2 approximates what Module 1 could not compute, moving from mean-field theory and the variational free energy through the TAP equations and belief propagation to the Hubbard–Stratonovich transformation and the variational autoencoder. Module 3 stops approximating and starts sampling: Markov-chain Monte Carlo, Langevin dynamics, simulated annealing, and then the observation that stochastic gradient descent is itself a Langevin process, with the learning rate setting a temperature and flat minima favored for entropic reasons. Module 4 removes the obstruction rather than working around it, deriving score matching, the reverse-time stochastic differential equation, the Jarzynski and Crooks work relations and annealed importance sampling, and finally the denoising diffusion model, whose training loss we identify as a nonequilibrium free-energy estimator. Module 5 turns the machinery back on learning itself, through the information bottleneck and the phase transitions of inference, and the last week is a synthesis.

What is not here

A single semester leaves a great deal out, and the omissions below are deliberate.

I do not develop rigorous large-deviation theory, and I do not give the renormalization group its due; Week 5 computes mean-field exponents and states where they fail, but computing them correctly below the critical dimension is a different course. The replica method is used but not derived: Week 2 quotes the Amit–Gutfreund–Sompolinsky capacity $\alpha_c \approx 0.138$ and states precisely what the replica calculation supplies that our tools cannot, which is the average of $\log Z$ over quenched disorder. On the generative side, flow matching appears only as a pointer in Week 12, and Schrödinger bridges, the optimal-transport and JKO view of diffusion, and the thermodynamics of computation are absent entirely; for all of these the Welling, Lu, and Holdijk text is the natural continuation, and I have said so where each would have gone. The hardware and systems side of machine learning appears nowhere. Neither does any serious treatment of transformers beyond the attention identity of Week 4, which is a statement about one architecture's update rule and not a theory of language models.

The tutorial problems are not reproduced here. They are revealed and worked in the room, a decision about the tutorials and not about these notes, whose reasoning is given in the course description.

The tutorials are run by Sander Hummerich and Gerrit Gerhartz. Errors that survive in these notes are mine, and I would be glad to hear about them.

Heidelberg, Winter Semester 2026/27

Tristan Bereau

1 Introduction: the shared language

Recommended reading

Core (~1 h). Jaynes (1957), §§1–4 — the origin of the maximum-entropy argument we carry out below: inference as the least-committal distribution consistent with what is known. This is the single source the lecture is built on. Alongside it, the introduction and §II of Mehta et al. (2019), which set up the physics ML vocabulary — energy, loss, likelihood — that the whole course speaks; this review is also the reference text for the tutorials.
Optional background. The probability-and-entropy chapters of Schwarz (2023), for anyone who wants their master's statistical mechanics refreshed to the level this course assumes; MacKay (2003), Ch. 2–4, for entropy and inference from the information-theory side; Sethna (2021), Ch. 5–6, for entropy and free energy with the "thinking in probabilities" emphasis we lean on. The historical anchors — Shannon (1948), Hopfield (1982), Ackley, Hinton, and Sejnowski (1985) — return as primary sources in Weeks 2–3, so a first pass now pays off later.
Prerequisite reminder. The Boltzmann distribution, entropy, and free energy are assumed from master's statistical mechanics. From the companion *Machine Learning and Physics* course we assume only fluency with the softmax layer, which the identification in §*The softmax is a Boltzmann distribution* rests on; a reader arriving without it should read Mehta et al. (2019) §II first. What is new here is the framing of these objects as the shared substrate of generative machine learning — not the objects themselves.

1.1 Why a nineteenth-century equation keeps reappearing

We write three objects side by side. The Boltzmann distribution of statistical mechanics,

$$p(x) \propto e^{-\beta E(x)},$$

the softmax layer that terminates essentially every classifier trained today,

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}},$$

and the *score* of a probability distribution,

$$\nabla_x \log p(x),$$

the vector field a diffusion model learns in order to turn noise into images. The first two are the same equation; the third is that equation differentiated, which is precisely what removes the normalizer, since for $p \propto e^{-\beta E}$ the score is $-\beta \nabla_x E$ — the force of the energy landscape, with Z gone. The resemblance is not an analogy but a pedigree. By the end of this lecture we will have proven that the first two are identical; Week 10 establishes the third, and Week 12 identifies the score with the noise-prediction target ε_θ that industrial diffusion models actually minimize.

The claim rests on a precise intellectual lineage, traced in Figure 1.1, in which physics and what we now call machine learning have exchanged the same mathematical object for a century and a half. Boltzmann gave entropy a statistical meaning in the 1870s — $S = k \log W$, the equation on his tombstone in Vienna (the constant-bearing form is Planck's): entropy counts the microstates consistent with a macrostate, and probability enters physics. Gibbs (1902) systematized the idea into *ensembles* and the canonical distribution $p \propto e^{-\beta E}$, together with its normalizer, the *partition function* Z — from the German *Zustandssumme*, "sum over states." The object this whole course revolves around is already on the page in 1902.

The pivot comes at mid-century. Shannon (1948) defined the information entropy $H = -\sum_i p_i \log p_i$ — the same functional Gibbs wrote down, but now measuring uncertainty in a message rather than heat in a gas. Jaynes (1957) then turned the logic around: the equilibrium distribution of statistical mechanics is nothing but the *maximum-entropy* distribution consistent with what one knows, so that the usual computational rules of the field, beginning with the determination of the partition function, "are an immediate consequence of the maximum-entropy principle." With Jaynes, the Boltzmann distribution stops being a statement about molecules and becomes a principle of *inference* — and inference is what learning machines do. This is the derivation we carry out in full in Section 1.4.

The machine-learning lane takes over in 1982. Hopfield (1982) imported the Ising energy function into computer science: a recurrent network of binary units whose dynamics roll downhill on an energy landscape, with memories stored as its minima — learning becomes the *shaping of a landscape*. Ackley, Hinton, and Sejnowski (1985) made the construction stochastic and trainable, the Boltzmann machine, and immediately hit the wall this course is organized around: the training gradient requires an expectation under the model, and that expectation requires Z. Amit, Gutfreund, and Sompolinsky (1985) answered a quantitative question with spin-glass methods — how many patterns can a Hopfield network store? — obtaining the celebrated capacity $\alpha_c \approx 0.138$ patterns per neuron (Amit, Gutfreund, and Sompolinsky 1987), the first time the statistical mechanics of disordered systems yielded a quantitative, closed-form theory of a learning system. Hinton (2002) supplied a practical dodge: contrastive divergence approximates the intractable model expectation with a short Gibbs chain — cheap, biased, and enormously influential.

The modern turn reunites the two lanes. Hyvärinen (2005) showed that an unnormalized model can be fit by matching the *score* $\nabla_x \log p$, in which Z drops out entirely because $\nabla_x \log Z = 0$ — the clean elimination, where before there had only been evasions. In parallel, nonequilibrium statistical mechanics produced the work relations of Jarzynski (1997) and Crooks (1999), which recover equilibrium free-energy differences — ratios of partition functions — from finite-time, driven trajectories. Sohl-Dickstein et al. (2015) read that literature and built a generative model from it explicitly: destroy structure with a forward diffusion, learn the reverse process,

and generation *is* nonequilibrium thermodynamics run backwards. Denoising diffusion probabilistic models (Ho, Jain, and Abbeel 2020) made the construction state of the art, and Song, Sohl-Dickstein, et al. (2021) unified the whole family as a forward stochastic differential equation plus a reverse-time SDE driven by the score $\nabla_x \log p_t$. Every reappearance in this story is the same normalization, met and evaded in a new form.

The mathematics of counting configurations weighted by energy is the same mathematics whether the configurations are gas microstates, stored memories, or images — and the recurring hard part is always the same normalization.

This also fixes the course's position relative to its companion. *Machine Learning and Physics* teaches the working methods — how to run a DDPM, a VAE, an MCMC chain. This course is its structural-derivation complement: we derive why those methods work, from statistical mechanics — reverse-time SDEs, the Jarzynski and Crooks relations, the Fokker–Planck equation of training, mean-field and replica theory.

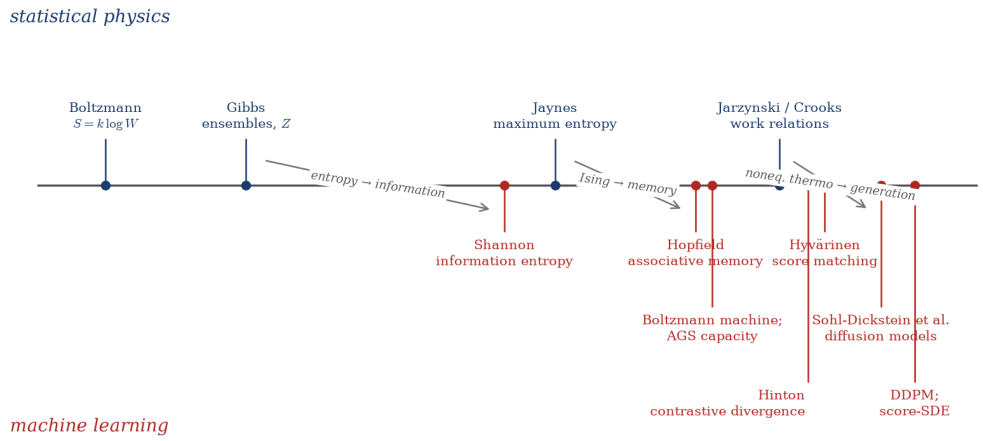

Figure 1.1: The pedigree this course traces. Two lanes — statistical physics above the axis, machine learning below — keep handing the same object back and forth: Gibbs's partition function becomes Shannon's entropy, Jaynes's inference principle, Hopfield's landscape, the Boltzmann machine's training obstruction, and finally the diffusion model's generative engine. Gray arrows mark the crossings.

> **Take-home 1**
>
> Statistical physics and generative machine learning are two dialects of one language: *probability distributions defined by an energy.* This course traces the derivation, not the analogy.

1.2 The bridge, in two directions

The traffic across this bridge runs both ways; the two directions organize all fourteen weeks (Figure 1.2).

In the **physics → ML** direction, an energy function defines a probability distribution; *learning* becomes shaping an energy landscape so that the data sit in its

basins, and *generation* becomes sampling from the shaped landscape. Modules 1, 3 and 4 live on this arrow. In the **ML → physics** direction, the tools of statistical mechanics are turned onto learning itself: training under stochastic gradients is a *stochastic process* with its own Fokker–Planck equation and stationary measure, and inference is *variational free-energy minimization*. Modules 2 and 3 live on this arrow, and so does Module 5, aimed at a different target: Modules 2 and 3 turn the machinery on learning *algorithms* and ask what a given method computes and how accurately, while Module 5 turns it on learning *problems* and asks which of them admit a solution at all. Week 13's detectability threshold is a statement about a task, not about any particular method for it.

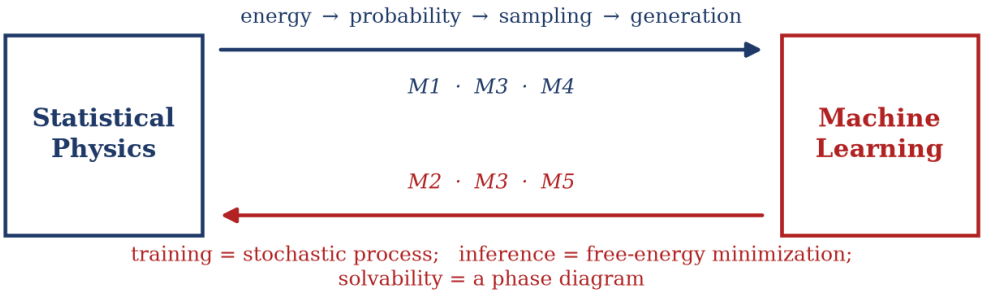

Figure 1.2: The two-way bridge, and the filing cabinet for the semester. Physics → ML: an energy defines a probability, learning shapes the landscape, generation samples it (Modules 1, 3, 4). ML → physics: training is a stochastic process, inference is free-energy minimization, and solvability itself has a phase diagram (Modules 2, 3, 5).

Beneath both directions sits one operational tension that every week of the course responds to: for essentially every interesting model, the unnormalized weight $e^{-E(x)}$ is trivial to write down and to evaluate at any single configuration — while its normalizer

$$Z = \sum_x e^{-E(x)}$$

is intractable. Everything hard in this course is a strategy for living with, approximating, or eliminating Z.

The course map follows directly. We open with the shared foundations (this week). Module 1 (Weeks 2–4) develops the *energy-based view* — Hopfield networks, Boltzmann machines and contrastive divergence, modern energy-based models, and the identification of modern Hopfield networks with attention — and it is here that Z is **raised** as the training obstruction. Module 2 (Weeks 5–7) develops the *variational and mean-field* toolbox — the variational free energy, TAP equations, belief propagation and approximate message passing, Hubbard–Stratonovich transformations and variational autoencoders — where Z is **approximated**. Module 3 (Weeks 8–9) treats *sampling and the Fokker–Planck equation of training* — Markov chain Monte Carlo, Langevin dynamics, simulated annealing, SGD-as-Langevin and flat

minima — where Z is **sampled around**. Module 4 (Weeks 10–12) is the resolution: the *nonequilibrium thermodynamics of generative modeling* — score matching, the Jarzynski and Crooks relations, annealed importance sampling, DDPM and score-based SDEs — where Z is **eliminated**. Module 5 (Week 13) turns the same machinery on representation and on inference itself: the information bottleneck, where Z normalizes an encoder and compression is **traded** against relevance at an inverse temperature, and the phase transitions of statistical inference, where a partition function over hypotheses decides which problems are solvable at all. Week 14 synthesizes.

The course's spine — and the backbone of the exam — runs from Week 3, *"the Boltzmann machine cannot be trained because* $\nabla_\theta \log Z$ *is intractable,"* to Week 10, *"score matching removes* Z *because* $\nabla_x \log Z = 0$*."* Note the subscripts: the obstruction is a gradient with respect to the *parameters* θ; the escape is a gradient with respect to the *configuration* x. These are different derivatives of the same offending object, and the precision matters — we return to it in Week 10.

> Take-home 2
>
> *You can always write the model down; you can almost never normalize it.* The partition function Z is the through-line of the course — raised as an obstruction in Module 1, resolved in Module 4.

1.3 The shared vocabulary

Students arrive here with statistical mechanics already in hand, so we do not re-derive; we state and cite, and use the space for the machine-learning reframing. The following panel fixes the notation for all fourteen weeks.

Symbol	Meaning	Notes
x	*configuration / microstate*	a gas microstate, a spin configuration, a stored pattern, an image — deliberately overloaded
$E(x)$	energy / cost	physics Hamiltonian = ML loss, score, negative log-likelihood
β	inverse temperature $1/T$	clashes with ML's β (e.g. β-VAE, Adam's $\beta_{1,2}$) — we always mean $1/T$
Z	*partition function* $\sum_x e^{-\beta E(x)}$	the course's through-line
F	(Helmholtz) *free energy* $-T \log Z$	$= \langle E \rangle - TS$
S	entropy $-\sum_x p(x) \log p(x)$	a functional of a distribution, not of one state

Symbol	Meaning	Notes
$\langle\cdot\rangle$	expectation	subscript the distribution: $\langle\cdot\rangle_{\text{data}}$, $\langle\cdot\rangle_{\text{model}}$
$p(x),\ q(x)$	model / variational (or data) distributions	fixed per lecture
θ	model parameters	gradients ∇_θ
∇_x	gradient w.r.t. the configuration	distinct from ∇_θ — the score-matching distinction hinges on this

We fix three conventions once and use them throughout. First, we set $k_B = 1$: temperature is measured in energy units and entropy is dimensionless. Second, log means the natural logarithm throughout, so entropy is measured in *nats* — with that choice the information-theoretic and thermodynamic entropies are literally the same object, not proportional ones; where a base-2 aside is needed we will write $\log_2$ explicitly. Third, energies always enter as $e^{-\beta E}$: lower energy means higher probability, and on the ML side the logits will accordingly be *negative* energies.

Two of the entries in the panel call for a comment. The energy $E(x)$ is, in physics, a Hamiltonian one *measures*; in machine learning it is a loss or score one *designs* — same mathematical role, opposite provenance, and the whole of Module 1 lives in that shift. The free energy $F = -T \log Z = \langle E \rangle - TS$ is the logarithm of the very object we cannot compute; when Module 2 constructs variational bounds, it is F they bound.

> **Trap**
>
> Entropy in this course is a functional of a *distribution*, not a property of a single configuration. A configuration has an energy; only an ensemble has an entropy. Students arriving from a thermodynamics-first background routinely conflate the two; keep them separate.

1.4 Maximum entropy and the Boltzmann distribution

We now carry out the derivation in full. The question it answers is an inference question, and posing it that way is Jaynes's insight (Jaynes 1957).

Suppose we have an energy function $E(x)$ over configurations, and suppose the only thing we know about the system is its average energy $\langle E \rangle = U$. What probability distribution $p(x)$ should we assign? We refuse to smuggle in any assumption beyond what we actually know — and that refusal can be formalized: among all distributions consistent with the constraints, choose the one of *maximum entropy*,

$$S[p] = -\sum_x p(x) \log p(x),$$

because any distribution of lower entropy encodes information we do not possess. The constraints are ① normalization, $\sum_x p(x) = 1$, and ② the known mean energy, $\sum_x p(x)\,E(x) = U$.

The derivation. We maximize $S[p]$ subject to ① and ② with Lagrange multipliers α and β:

$$\mathcal{L}[p] = -\sum_x p(x)\log p(x) \;-\; \alpha\Big(\sum_x p(x) - 1\Big) \;-\; \beta\Big(\sum_x p(x)E(x) - U\Big).$$

Stationarity with respect to each $p(x)$ — the derivative of $-p\log p$ is $-\log p - 1$ — gives

$$\frac{\partial \mathcal{L}}{\partial p(x)} = -\log p(x) - 1 - \alpha - \beta E(x) = 0$$

$$\Rightarrow \quad p(x) = \underbrace{e^{-1-\alpha}}_{\text{independent of } x}\, e^{-\beta E(x)}.$$

The underbraced prefactor carries no x-dependence; it is fixed entirely by normalization ①. Imposing it defines the normalizer the whole course revolves around,

$$\boxed{Z(\beta) = \sum_x e^{-\beta E(x)}} \tag{1.1}$$

partition function

and the maximum-entropy distribution takes its final form,

$$\boxed{p(x) = \frac{e^{-\beta E(x)}}{Z(\beta)}} \tag{1.2}$$

Boltzmann / Gibbs distribution

The maximum-entropy distribution consistent with a known mean energy is exponential in the energy — nothing else is assumed.

What is β? Mechanically, it is the multiplier that enforces the energy constraint, tuned so that $\langle E\rangle = U$; physically, we identify it as the *inverse temperature*, $\beta = 1/T$. The two limits make the identification vivid, and Figure 1.3 draws them: at low β (hot), the exponential is nearly flat and probability spreads almost uniformly over configurations; at high β (cold), the mass collapses onto the ground state — the global minimum of the landscape. In one line: *temperature is the exchange rate between entropy and energy.* An ML reader already knows this dial — it is the "temperature" setting on a language model's sampler, and it is the same β.

A second lens. There is more in the same result. $Z(\beta)$ is not merely a normalizer; it is a *generating function*. Differentiating $\log Z$ produces the observables: $\langle E\rangle = -\partial \log Z/\partial\beta$, and the second derivative gives the energy fluctuations and hence the heat capacity (both derived in the next lecture, in Chapter 2, and they return quantitatively in Module 5). Knowing Z is knowing everything about the equilibrium ensemble — which is exactly why its intractability is so costly.

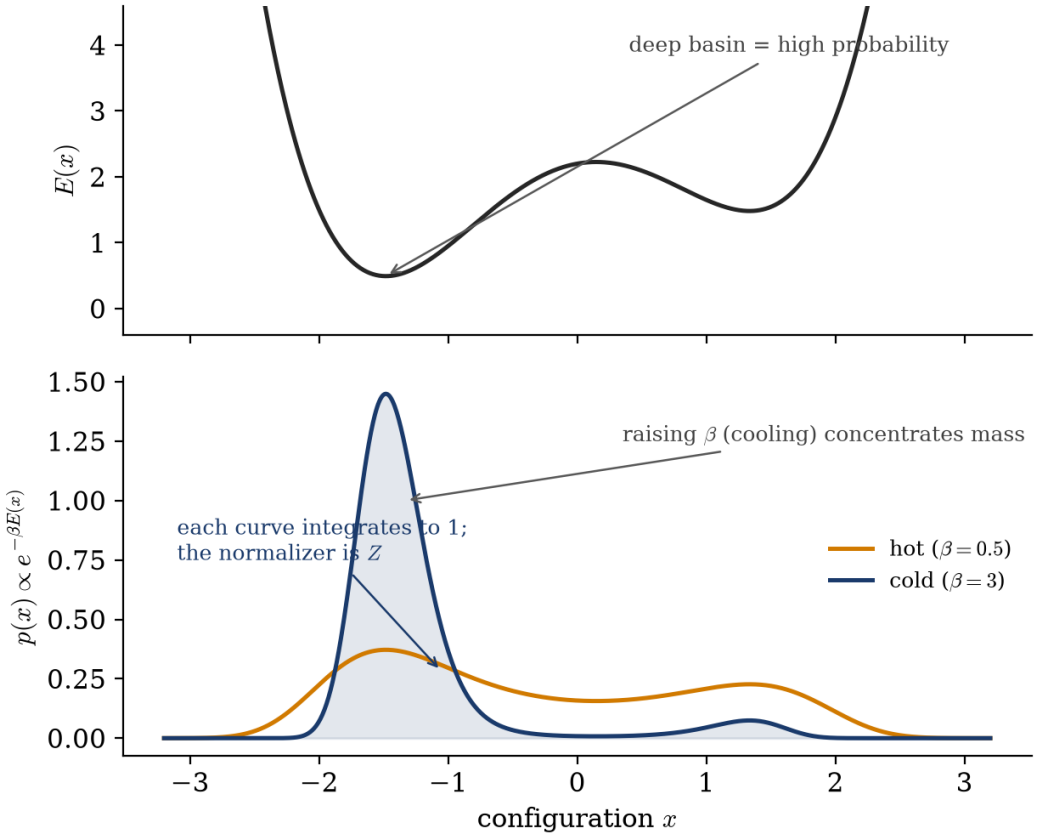

Figure 1.3: The central picture of the course. Top: an energy landscape $E(x)$ with basins of unequal depth. Bottom: the Boltzmann densities $p(x) \propto e^{-\beta E(x)}$ it induces at $\beta = 0.5$ (hot — nearly flat) and $\beta = 3$ (cold — mass concentrated in the deep basin). Each curve integrates to one, and the normalizer that enforces this is Z; temperature tunes exploration against exploitation.

Trap

Z is *not* a constant you may ignore. Change any parameter of E and Z changes with it; in a learned model, $\nabla_\theta \log Z$ is exactly the term that blocks training. The habit of "dropping the normalization" — harmless in a homework problem — is a reflex to unlearn before this course.

1.5 The softmax is a Boltzmann distribution

The corollary takes four lines. Let the configurations be class labels, $x \in \{1, \dots, K\}$, and let a network's last layer output the logits $z_1, \dots, z_K$. Define the energy of class i to be $E_i = -z_i$ and set $\beta = 1$. Then Equation 1.2 reads

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1.3}$$

softmax

The softmax layer is a Boltzmann distribution over labels, and its denominator is a partition function over K states.

Every reader of these notes has been computing partition functions daily, under an assumed name. The reason Z has never hurt: a classifier sums over $K \approx 10$–10^3 classes, and a sum with a thousand terms is free. The moment the configurations are not labels but high-dimensional objects — images, spin configurations, molecules — the same sum acquires exponentially many terms, and the field's difficulty begins. The next section makes that quantitative.

Two further identifications are flagged now and derived later. The cross-entropy loss is $-\log p_{\text{correct}}$ — a free-energy difference — and minimizing it is maximum-likelihood estimation on a Boltzmann model (shown in Week 3). And the relation between F, the Kullback–Leibler divergence, and variational bounds is the engine of Module 2. Neither is derived today.

1.6 Example: two states, then the 2^N wall

A toy that works. Take the smallest possible system: one binary unit, $x \in \{0, 1\}$, with $E(0) = 0$ and $E(1) = \Delta E > 0$, so that state 0 is the ground state. The partition function has exactly two terms, $Z = 1 + e^{-\beta \Delta E}$, and the occupation of the excited state is

$$p(1) = \frac{e^{-\beta \Delta E}}{1 + e^{-\beta \Delta E}} = \frac{1}{1 + e^{\Delta E / T}}.$$

Measuring temperature in units of the gap, $\tau \equiv T/\Delta E$, the numbers behave as follows:

$\beta \Delta E$	$\tau = T/\Delta E$	$p(1)$	reading
0	∞	0.500	hot: both states equally likely

$\beta\Delta E$	$\tau = T/\Delta E$	$p(1)$	reading
$\ln 2 \approx 0.693$	1.44	0.333	excited : ground $= 1 : 2$
1	1.00	0.269	thermal energy equals the gap
2	0.50	0.119	—
5	0.20	0.0067	cold: mass collapses onto the ground state
$\to \infty$	$\to 0$	$\to 0$	only the ground state survives

As a function of the gap at fixed temperature, the same occupation reads $p(1) = \sigma(-\beta\Delta E)$ with $\sigma(u) = 1/(1 + e^{-u})$ — the *logistic function*. The sigmoid of every neural-network textbook *is* the two-state Boltzmann distribution, the second such identification of the day. The left panel of Figure 1.4 draws the occupation against temperature instead, reading off the two limits of the table.

The wall. Now let the system grow: N binary units — an Ising layer, or the visible layer of a small restricted Boltzmann machine. The partition function

$$Z = \sum_{x \in \{0,1\}^N} e^{-\beta E(x)}$$

has 2^N terms, and the right panel of Figure 1.4 shows what that exponential means in practice. For $N = 10$, Z has 1,024 terms — trivial. For $N = 300$, it has $2^{300} \approx 2 \times 10^{90}$ terms, roughly *ten orders of magnitude more than the* $\sim 10^{80}$ *atoms in the observable universe*. For $N = 784$ — the pixels of one small MNIST image — it has $2^{784} \approx 10^{236}$.

An order-of-magnitude check makes the cost concrete. Grant an optimistic 10^9 energy evaluations per second. Brute-forcing the $N = 300$ partition function then takes $\sim 10^{90}/10^9 = 10^{81}$ seconds, against a universe whose age is $\approx 4.4 \times 10^{17}$ seconds — about 10^{63} ages of the universe for one number. And yet evaluating $e^{-\beta E(x)}$ at any *single* configuration costs a microsecond.

> Evaluating one configuration is free; summing over all of them is impossible — and that single asymmetry is why Modules 1–4 exist. *You can always write the model down; you can almost never normalize it.*

In the Week 3 tutorial you will hit this 2^N wall yourselves, on a real Boltzmann machine. The resolution comes in Week 10.

Take-home 3

The cost of Z is exponential in system size: computable for a softmax over K labels, hopeless for a distribution over configurations. The entire course is strategies for that gap.

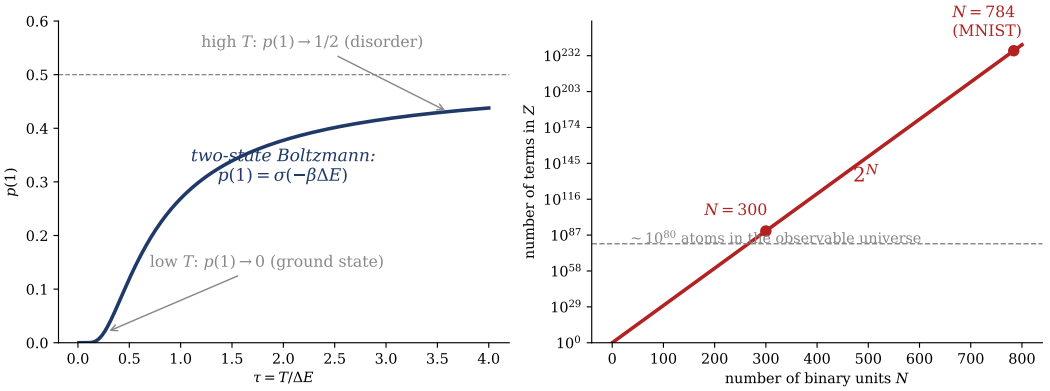

Figure 1.4: Left: the excited-state occupation $p(1)$ of the two-state system ($E(0) = 0$, $E(1) = \Delta E$) against reduced temperature $\tau = T/\Delta E$, with $p(1) \to 1/2$ at high temperature and $p(1) \to 0$ as the mass collapses onto the ground state; as a function of the gap at fixed temperature, the same occupation is the logistic $\sigma(-\beta\Delta E)$. Right: the number of terms in Z, equal to 2^N, against the number of binary units N; the dashed line marks the $\sim 10^{80}$ atoms in the observable universe, crossed already at $N \approx 266$, with $N = 300$ and $N = 784$ (one MNIST image) indicated.

> **Trap**
>
> "Softmax has no partition-function problem" is true only because K is small. The same object over configurations — not labels — is intractable. Do not let the benign case set your intuition for the general one.

1.7 Course mechanics and the contract

The practical arrangements are stated once, here, and kept current on the course-information page. Fourteen weeks, five modules, two 90-minute lectures per week (Lectures 1 and 2, 9–11) plus an afternoon tutorial (16–18); 8 credit points. The grade is a **written final exam, 100%** — no graded problem sets. The exam is built directly from the tutorials' in-class problems and is organized along the Z spine, so preparing for the exam *is* doing the tutorials; it is closed-book except for **one hand-written aid sheet** that you compile from your own tutorial solutions. Exam admission is an ungraded *Studienleistung*: submit the weekly stuck-point card for at least 8 of the 12 tutorial warm-ups, which run from Week 2.

The tutorials run every week and follow a productive-failure format: a light warm-up at home (a short reading and a small priming task, whose stuck-point card feeds the session's triage), then the hard problem — revealed only in the room and cracked in groups. Come having attempted the crux; meeting the wall in class is part of the design. On the weeks whose problem is computational, bring a laptop if you have one: after a silent individual attempt and a written group prediction, each group opens one machine for about twenty minutes and fills in a small number of marked lines in a prepared notebook, and the closing round runs on the projector for the room. One machine per group of three or four is enough, so not everyone needs to carry one. On the weeks whose problem is a derivation there are no machines at all. The exam tests computational reasoning rather than live coding.

This course does not develop rigorous large-deviation theory, does not give the renormalization group its full due, and does not treat the hardware and systems side of machine learning; each would be its own course, and the reading list points to where they are done well. What this course does claim is the *derivational* road set out at the start of this chapter: not how to run the methods, but why they work — with the partition function as the recurring obstruction.

1.8 Outlook

The next lecture assembles the quantitative foundations we lean on all semester: ensembles, the free-energy identities used as working tools rather than slogans, maximum entropy as an inference engine, and the Gaussian and saddle-point machinery that recurs from Module 2 onward. Week 2 then opens Module 1 with the Hopfield model — the Ising energy repurposed as an associative memory, and the first time in the course that $E(x)$ is something we *design* rather than measure.

All three take-home points reduce to one: **you can always write the model down; you can almost never normalize it.**

2 Foundations: the saddle-point toolkit

Recommended reading

Core (~1 h). Schwarz (2023), the free-energy and saddle-point sections together with the mean-field treatment of the Ising model, where the free-energy-density saddle is carried out on a real Hamiltonian — this lecture is, in structure, a compression of exactly that material into the tools we will reuse all semester. Alongside it, the statistics and Gaussians sections of Mehta et al. (2019), for the machine-learning vocabulary — Gaussians, log-likelihoods — that the saddle-point tools attach to in Modules 2 and 4.
Optional background. Sethna (2021), the free-energy chapter and the appendix on Laplace's approximation, in the "thinking in probabilities" spirit this course leans on; MacKay (2003), Ch. 2, for Laplace's method and Stirling's formula from the inference side (freely available online). Students who want the complex-analytic version — the method of steepest descent proper — will find it in any graduate mathematical-methods text; we will not need it.
Prerequisite reminder. Ensembles, the free energy, and the Gaussian integral are assumed from master's statistical mechanics. What is genuinely new here is the packaging: the *saddle-point method* elevated to the recurring computational strategy of the entire course, and the free energy reframed as the object that strategy computes.

2.1 The sum versus its largest term

The previous lecture ended at a wall. We derived the Boltzmann distribution from maximum entropy, identified the softmax as a partition function over labels, and then saw the same object become intractable: for N binary units, $Z = \sum_x e^{-\beta E(x)}$ has 2^N terms, and already at $N = 300$ that is $\sim 2 \times 10^{90}$ — ten orders of magnitude more terms than there are atoms in the observable universe (Section 1.6). We left the number on the board as a provocation.

And yet physicists have computed with Z — phase diagrams, heat capacities, critical temperatures — for a century and a half. The question, then: *if we can never sum it, how does anyone ever compute anything with Z?*

The resolution is not cleverness about the sum; it is a change of target. We give up on Z itself and compute its logarithmic *density*, $\frac{1}{N} \log Z$, in the limit where physics and machine learning both live — N large: 10^{23} atoms, 10^6 pixels, 10^9 weights. In that limit a simplification takes over: when the exponent of a sum is *extensive* — proportional to N — the sum is dominated by its largest term, and $\frac{1}{N} \log Z$ can be read off from that single term alone. The central dichotomy — for this lecture and for the course — is *the sum versus its largest term*: the sum is hopeless, the largest term is a one-dimensional maximization.

The technology that makes this precise is *Laplace's method*, known to physicists as the *saddle-point approximation*, and it is built out of one integral — the Gaussian. The chain runs: Gaussian integral, then Laplace's method on top of it, then Stirling's formula as its canonical corollary, and finally the free energy as a saddle — mean-field theory in embryo and the seed of all of Module 2. Before the new machinery, we spend a few compressed pages placing Z properly: on the ladder of ensembles, and as a generating function — the corollaries the previous lecture deferred.

Everything below is stated for sums or for integrals as convenient. A sum over configurations and an integral over configurations differ by a measure that contributes nothing at the exponential order we work in, so $\sum_x \leftrightarrow \int \mathrm{d}x$ throughout; we lead discrete this week, per the course convention, and switch to continuous from Module 3 on.

2.2 The ensemble ladder, in one panel

In the previous lecture, the canonical distribution arrived by inference: maximum entropy under a mean-energy constraint. The physics tradition arrives at the same object through a different door — as one rung on a ladder of *ensembles*, the systematization owed to Gibbs (1902) — and we spend one compressed panel placing the course on that ladder, because the choice of rung is a choice of *what is held fixed*. This is master's material; we state it and move on.

Ensemble	Held fixed	Distribution	Thermodynamic potential
microcanonical	$E,\ N$	$p(x) = 1/\Omega(E)$ (uniform on the shell)	$S(E) = \log \Omega(E)$
canonical	$T,\ N$	$p(x) = e^{-\beta E(x)}/Z$	$F(T) = -T \log Z$
grand-canonical	$T,\ \mu$	$p(x) \propto e^{-\beta[E(x)-\mu N(x)]}$	$\Omega_G(T,\mu) = -T \log Z_G$

The microcanonical rung is the previous lecture's tombstone relation $S = \log W$ dressed as an ensemble: fix the energy exactly, count the microstates, take the log. It is the only rung with no free dial — every other ensemble is reached from it by a *Legendre transform*, trading a conserved quantity for its conjugate field: $E \leftrightarrow \beta$ takes microcanonical to canonical, $N \leftrightarrow \mu$ takes canonical to grand-canonical, and the potentials in the last column are the transform partners. The general algebra is standard and we do not reproduce it (Schwarz 2023) — with one exception, because the course lives on the rung it lands on.

One rung, derived: the heat bath. The physics tradition's route to the canonical distribution — the counterpart of the previous lecture's inference route — is a three-line marginalization, and it is worth having both doors on record (Welling, Lu, and Holdijk 2026, ch. 3 on the canonical ensemble). Embed the system of interest in a much larger *bath* with which it exchanges energy, and let the pair be isolated at total energy E_{tot}. The combined system sits on the microcanonical rung — uniform on its shell — and the system's own distribution follows by tracing the bath out: when the system holds configuration x, the bath is left with energy $E_{\mathrm{tot}} - E(x)$, and the probability of x is proportional to the number of ways the bath can accommodate that,

$$p(x) \;\propto\; \Omega_{\text{bath}}(E_{\text{tot}} - E(x)) \;=\; \exp\Big[S_{\text{bath}}(E_{\text{tot}} - E(x))\Big],$$

using the first rung's own relation $S = \log\Omega$. Now use that the bath is large, $E(x) \ll E_{\text{tot}}$, and expand its *entropy* — not its microstate count — to first order:

$$S_{\text{bath}}(E_{\text{tot}} - E(x)) \;=\; S_{\text{bath}}(E_{\text{tot}}) \;-\; E(x)\,\underbrace{\frac{\partial S_{\text{bath}}}{\partial E}}_{\equiv\, 1/T} + \mathcal{O}(E(x)^2\,\partial_E^2 S_{\text{bath}}),$$

the canonical rung, from the microcanonical one

where $1/T \equiv \partial S/\partial E$ is the microcanonical definition of temperature — the slope of the bath's entropy against energy, the one dial the bath presents to anything coupled to it. The leading term is independent of x and absorbs into the normalization, and what remains is $p(x) = e^{-E(x)/T}/Z$: the canonical rung, derived rather than listed. The reading to keep: the Boltzmann factor *counts bath microstates* — a configuration is penalized by exactly the entropy the bath must surrender to fund its energy, $e^{-\Delta S_{\text{bath}}}$. The dropped curvature term is where the assumption of a much larger bath enters quantitatively: $\partial_E^2 S_{\text{bath}} = -1/(T^2 C_{\text{bath}})$ vanishes as the bath's heat capacity grows, which is the precise content of "the bath's temperature is unaffected by the exchange." And the two doors now visibly open onto one room: The previous lecture manufactured β as the Lagrange multiplier of a mean-energy constraint; here the same β arrives as a physical slope, the bath's entropy price per unit of energy. That the multiplier *is* the slope is the Legendre duality of the ladder, met in its most concrete instance.

What matters for us is where this course lives: on the canonical rung, almost without exception, from here to Module 5. The canonical ensemble is the natural home of learning problems — a model at fixed "temperature" (noise level, regularization strength) rather than fixed energy — and its potential $F = -T\log Z$ is the object every module manipulates. The grand-canonical rung is listed for completeness; it will not reappear unless particle-number-like quantities do. (A symbol warning: the grand potential Ω_G shares its letter with the microstate count $\Omega(E)$ one row above — unrelated objects, distinguished by the subscript, and we will need only the latter.)

> Trap
>
> Ensemble *equivalence* — the statement that all rungs give the same physics — holds for mean values in the thermodynamic limit, but it **fails for fluctuations and at phase transitions**. The moment Module 2 studies broken symmetry, the distinction starts to matter.

2.3 The partition function as a generating function

Last time we noted that Z is "not a bookkeeping constant." Differentiate $\log Z(\beta)$ with respect to β:

$$\frac{\partial \log Z}{\partial \beta} = \frac{1}{Z}\sum_x (-E(x))\, e^{-\beta E(x)} = -\sum_x E(x)\,\underbrace{\frac{e^{-\beta E(x)}}{Z}}_{=\,p(x)} = -\langle E\rangle,$$

so that

$$\langle E \rangle = -\frac{\partial \log Z}{\partial \beta}$$

mean energy from $\log Z$

Differentiating the log-partition function produces the observables — no additional summation over configurations required.

Differentiating again, the same product rule that produced $-\langle E \rangle$ now produces both $\langle E^2 \rangle$ and $\langle E \rangle^2$:

$$\frac{\partial^2 \log Z}{\partial \beta^2} = \langle E^2 \rangle - \langle E \rangle^2 = \mathrm{Var}(E) \;\geq\; 0,$$

and since $\partial/\partial T = -\beta^2\, \partial/\partial\beta$, the heat capacity follows immediately:

$$C \;=\; \frac{\partial \langle E \rangle}{\partial T} \;=\; \beta^2\, \mathrm{Var}(E).$$

The pattern has a name: $\log Z$ is a *cumulant generating function* — its β-derivatives are the cumulants of the energy — and the second identity says that a *fluctuation* (the energy variance, a property of the equilibrium ensemble sitting still) equals a *response* (the heat capacity, how the system reacts when you turn the temperature dial). This fluctuation–response pattern is not a curiosity of the energy: it is the seed of the susceptibility and fluctuation–dissipation results of Module 2, and of the information-theoretic identities of Module 5. As a bonus, $\mathrm{Var}(E) \geq 0$ says $\log Z$ is convex in β — a fact the Legendre transforms of the previous section quietly rely on.

The free energy ties the two derivatives together: $F = -T \log Z = \langle E \rangle - TS$, the previous lecture's identity, now backed by explicit derivatives. F is the object the next two sections learn to *compute* — and the object the variational methods of Module 2 learn to *bound*, via $D_{\mathrm{KL}} \geq 0$.

Take-home 1

Z is a generating function: $\langle E \rangle = -\partial \log Z/\partial\beta$ and $\mathrm{Var}(E) = \partial^2 \log Z/\partial\beta^2$, so a *fluctuation* (energy variance) equals a *response* (heat capacity), $C = \beta^2\, \mathrm{Var}(E)$. Derivatives of $\log Z$ are physics — which is why we always want $\log Z$, never Z.

Trap

$\langle E \rangle = -\partial \log Z/\partial\beta$ holds at **fixed couplings**: the derivative moves β and nothing else. If the energy itself depends on β (or on T, as effective Hamiltonians often do), differentiating silently adds terms. State what is held fixed before differentiating — always.

2.4 The engine: Gaussian integral, Laplace's method, Stirling's formula

The derivation builds the engine the rest of the course runs on. The problem is the dichotomy of the opening section: we cannot sum 2^N terms, but we can find the largest one, and we need to know *how good* "keep only the largest term" is. Laplace's method answers that question quantitatively, and the Gaussian integral is the piece that makes the answer computable.

The Gaussian integral. We start with the atom, *the* recurring integral of the course:

$$I(a) = \int_{-\infty}^{\infty} \mathrm{d}x\; e^{-ax^2/2}, \qquad a > 0.$$

The integrand has no elementary antiderivative; the classic trick — which we carry out once in full and then invoke freely — is to compute the *square* of the integral and pass to polar coordinates, where the measure $r\,\mathrm{d}r$ supplies exactly the missing factor:

$$\begin{aligned} I^2(a) &= \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{\infty} \mathrm{d}y\; e^{-a(x^2+y^2)/2} \\ &= \int_0^{2\pi} \mathrm{d}\varphi \int_0^{\infty} r\,\mathrm{d}r\; e^{-ar^2/2} = 2\pi \left[-\frac{e^{-ar^2/2}}{a} \right]_0^{\infty} = \frac{2\pi}{a}, \end{aligned}$$

and therefore

$$\int_{-\infty}^{\infty} \mathrm{d}x\; e^{-ax^2/2} = \sqrt{\frac{2\pi}{a}} \tag{2.1}$$

Gaussian integral

Figure 2.1 draws the statement: a bell curve of width $\sim 1/\sqrt{a}$ enclosing an area $\sqrt{2\pi/a}$ — sharper curvature, narrower peak, smaller area. For later use we state, result-first, the multivariate extension with a linear term (proved by completing the square and diagonalizing):

$$\int \mathrm{d}^n x\; e^{-\frac{1}{2}x^\top A\, x + b^\top x} = \sqrt{\frac{(2\pi)^n}{\det A}}\; e^{\frac{1}{2}b^\top A^{-1} b}, \qquad A \succ 0.$$

Every Gaussian you will ever integrate in this course is Equation 2.1 plus a change of variables. This one labeled result already covers ① the Hubbard–Stratonovich transformation of Module 2, ② the Laplace approximation behind variational autoencoders, also Module 2, and ③ the transition kernel of every single step of a diffusion model, Module 4.

Laplace's method. The picture comes first, because the picture *is* the method. Consider an integral of the form

$$I_N = \int \mathrm{d}x\; e^{Nf(x)},$$

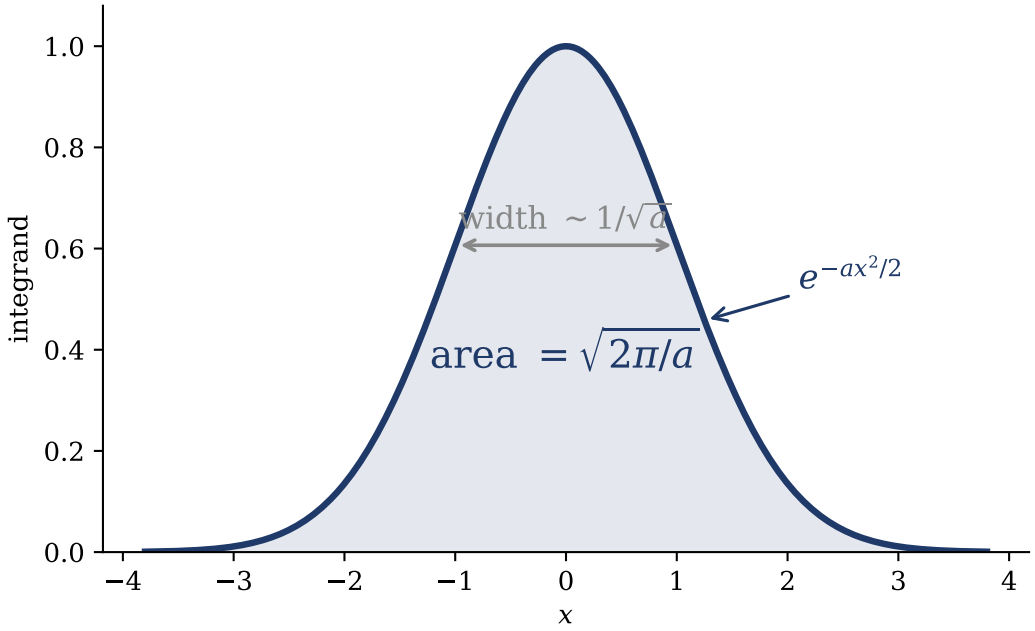

Figure 2.1: The atom of the course: the Gaussian integrand $e^{-ax^2/2}$ (drawn for $a = 1$) encloses an area of exactly $\sqrt{2\pi/a}$, with characteristic width $1/\sqrt{a}$ — sharper curvature a means a narrower, smaller-area peak. Every Gaussian integral from here to Module 4 — Hubbard–Stratonovich, the VAE's Laplace approximation, each diffusion kernel — is this one integral after a change of variables.

where N is large and the exponent is *extensive* — the factor N multiplies a smooth, N-independent function $f(x)$ with ① a unique interior maximum at x^*, so $f'(x^*) = 0$ and $f''(x^*) < 0$. (A warning on symbols: $f(x)$ here is a generic exponent, not the free-energy density $f(m)$ of the next section — the clash is standard in the literature, and we flag it rather than pretend it away.) Figure 2.2 shows what the factor N does to the integrand: $e^{Nf(x)}$ is the function $e^{f(x)}$ raised to the N-th power, so every ratio between the peak and the rest gets amplified N-fold in the exponent, and the integrand collapses onto a shrinking neighborhood of x^*. The integral is a sum of contributions; as N grows, one contribution swallows all the others.

The calculation formalizes the picture in three moves. Taylor-expand the exponent about x^* — about the *maximum*, precisely because the linear term vanishes there and the quadratic term is the leading survivor:

$$f(x) = f(x^*) + \tfrac{1}{2} f''(x^*)\,(x - x^*)^2 + \mathcal{O}\big((x - x^*)^3\big)\,.$$

Substitute, pull the constant out of the integral, and recognize what remains:

$$I_N \;\approx\; e^{Nf(x^*)} \int \mathrm{d}x\; e^{-\frac{1}{2}N|f''(x^*)|\,(x-x^*)^2},$$

which is Equation 2.1 with curvature $a = N|f''(x^*)|$. Hence

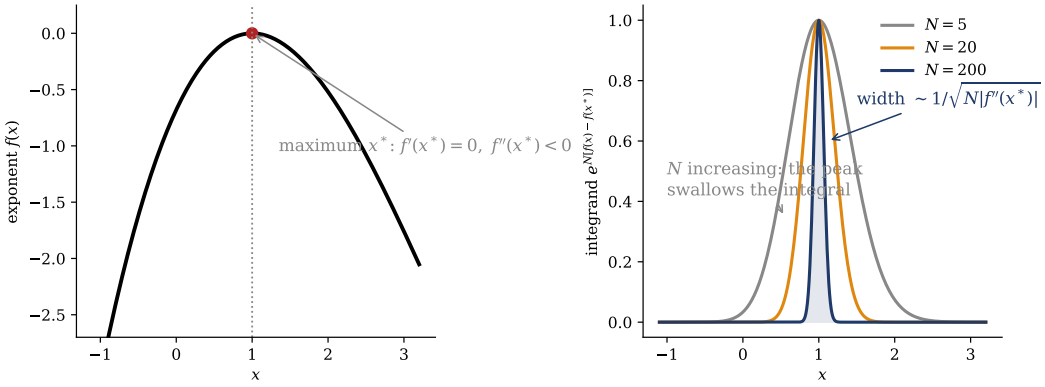

Figure 2.2: The central picture of the lecture. Left: a smooth exponent $f(x)$ with a unique maximum at x^*, where $f'(x^*) = 0$ and $f''(x^*) < 0$. Right: the integrand $e^{N[f(x)-f(x^*)]}$ for $N = 5$, 20, and 200 — each increase in N sharpens the peak, whose width shrinks as $1/\sqrt{N|f''(x^*)|}$, until the neighborhood of x^* carries essentially the entire integral. Laplace's method is the statement that this peak swallows the whole integral as $N \to \infty$.

$$\boxed{I_N = \int \mathrm{d}x\; e^{Nf(x)} \;\approx\; e^{Nf(x^*)} \sqrt{\frac{2\pi}{N\,|f''(x^*)|}}} \tag{2.2}$$

Laplace's method

Taking the logarithm and dividing by N gives the form we will actually use, over and over:

$$\frac{1}{N}\log I_N = f(x^*) + \mathcal{O}\!\left(\frac{\log N}{N}\right) \;\longrightarrow\; f(x^*) \quad (N \to \infty).$$

To leading order in N, an integral — or a sum — with an extensive exponent equals its single largest contribution; the Gaussian fluctuations around it are a $1/\sqrt{N}$ correction that survives only in the subleading log *term. The 2^N wall is not climbed; it is bypassed.*

A remark on names, for recognition rather than use. The same idea for complex or oscillatory exponents — deform the integration contour through the point where $f'(z) = 0$, a *saddle point* of the complex landscape — is the *method of steepest descent*, associated with Riemann and worked out by Debye (1909). Physicists say "saddle point" even in the real case, and so will we; when the replica literature of Module 2 says it, this is what it means. We will only ever need the real, Laplace version (MacKay 2003; Sethna 2021).

Stirling's formula. The engine's canonical corollary is Stirling's formula — published in 1730, a century before Laplace's method made it a one-liner — and it is the bridge back to the previous lecture's entropy counting. Start from the Gamma-function representation of the factorial and bring it to the Laplace template by substituting $x = ny$:

$$n! = \int_0^\infty \mathrm{d}x\; x^n e^{-x} \overset{x=ny}{=} n^{\,n+1} \int_0^\infty \mathrm{d}y\; e^{\,n(\log y - y)}.$$

The exponent $f(y) = \log y - y$ has its maximum at $f'(y^*) = 1/y^* - 1 = 0$, i.e. $y^* = 1$, with $f(1) = -1$ and $f''(1) = -1$. Applying Equation 2.2,

$$n! \;\approx\; n^{n+1}\, e^{-n} \sqrt{\frac{2\pi}{n}} = \sqrt{2\pi n}\, \left(\frac{n}{e}\right)^n,$$

or in the logarithmic form we will use:

$$\log n! = n \log n - n + \tfrac{1}{2} \log(2\pi n) + \mathcal{O}(1/n) \tag{2.3}$$

Stirling's formula

Stirling is Laplace applied to a factorial — and it is precisely the tool that turns the previous lecture's entropy counting $S = \log W$, with W a product of factorials, into a smooth, differentiable entropy density that calculus can grip. That conversion — counting into calculus — is what the next section puts to work.

Trap

The saddle point is a **large-N** statement — exact only as $N \to \infty$, and only when the exponent is **extensive** ($\propto N$). At finite N the Gaussian prefactor $\sqrt{2\pi/N|f''(x^*)|}$ is a real, quantitative correction, not decoration (the worked example below puts numbers on it). Writing $\int e^{Nf} \approx e^{Nf(x^*)}$ with no conditions attached is a frequent abuse in the literature — attach the regime, every time.

Take-home 2

Laplace's method: $\frac{1}{N} \log \int \mathrm{d}x\; e^{Nf(x)} \to f(x^*)$, where x^* maximizes f. In the large-N limit a sum or integral equals its largest term; the width of the peak around it is $\sim 1/\sqrt{N}$, and the Gaussian integral $\sqrt{2\pi/a}$ computes the correction. This is the universal computational strategy of the course.

2.5 The free energy as a saddle

Applying the engine to Z itself yields mean-field theory — the preview of Module 2.

The move that makes a 2^N-term sum tractable is a *reorganization*. Take N binary spins $x_i = \pm 1$ and define the *order parameter* $m = \frac{1}{N}\sum_i x_i$, the magnetization — one number summarizing a configuration. Instead of summing over configurations one by one, group them by their value of m. Two facts make the grouped sum exponential in form. First, the number of configurations with magnetization m is a binomial coefficient, $\binom{N}{N(1+m)/2}$, and Stirling's formula — this is exactly where Equation 2.3 does its work — turns it into $e^{Ns(m)}$ with the *entropy density*

$$s(m) = \log 2 - \tfrac{1}{2}\left[(1+m)\log(1+m) + (1-m)\log(1-m)\right].$$

Second, when the energy depends on the configuration only through m — true for all-to-all (*mean-field*) couplings, e.g. $E = -\frac{J}{2N}\left(\sum_i x_i\right)^2 = N\varepsilon(m)$ with $\varepsilon(m) = -\frac{J}{2}m^2$ — each grouped term carries the weight $e^{-\beta N \varepsilon(m)}$. The partition function collapses to a sum over the $N+1$ values of m:

$$Z = \sum_m e^{N[s(m) - \beta\,\varepsilon(m)]}.$$

This is precisely the Laplace template — an extensive exponent, a sum we may read as an integral — so Equation 2.2 applies verbatim, and the *free-energy density* $f = F/N$ becomes

$$\boxed{f \;=\; -\frac{1}{\beta N}\log Z \;\longrightarrow\; \min_m \left[\,\varepsilon(m) - T\,s(m)\,\right] \quad (N \to \infty)} \tag{2.4}$$

free-energy density as a saddle

The free energy is the minimum, over the order parameter, of "energy minus temperature times entropy": a 2^N-term sum has collapsed to a one-dimensional minimization, and the minimizer m^* is the equilibrium value of the order parameter. (The maximum of $s - \beta\varepsilon$ becomes a minimum of $\varepsilon - Ts$ under the overall minus sign — hence "minimize the free energy," the phrasing every physics course drills.) Note the symbols doing double duty, as flagged above: $f(m)$ is now a free-energy density and $s(m)$ an entropy density — lowercase for *per-spin* quantities, reserving F and S for totals.

Everything thermodynamically interesting now lives in the *shape* of $f(m) = \varepsilon(m) - Ts(m)$, and the temperature is the dial that deforms it. Figure 2.3 draws the competition for the mean-field energy above. At high T the entropy term dominates: $f(m)$ has a single well at $m^* = 0$, the disordered state — maximal entropy, no net magnetization. At low T the energy term wins: the single well splits into two symmetric wells at $m^* = \pm m_0$, and the system must settle into one of them. That splitting is *spontaneous symmetry breaking*, and the calculation we just sketched — group by an order parameter, Stirling the count, saddle the sum — *is* mean-field theory. We have done it here for the one energy where it is exact; doing it in full on the Hopfield network, and repairing it when it is not exact, is the business of Module 2 (Weeks 5–7).

> Take-home 3
>
> The saddle-point method turns the free energy into a minimization: $f = \min_m [\varepsilon(m) - Ts(m)]$ — the competition between energy and entropy resolved by one dominant order-parameter value. This *is* mean-field theory: planted here, solved properly in Module 2.

2.6 Example: how good is the largest term?

The approximation needs a reality check with actual numbers — first on Stirling, then on the concentration claim itself.

Stirling's accuracy. Compare $\log n!$ computed exactly against the leading form $n\log n - n$ and against the corrected form that keeps the Gaussian prefactor, $n\log n - n + \frac{1}{2}\log(2\pi n)$:

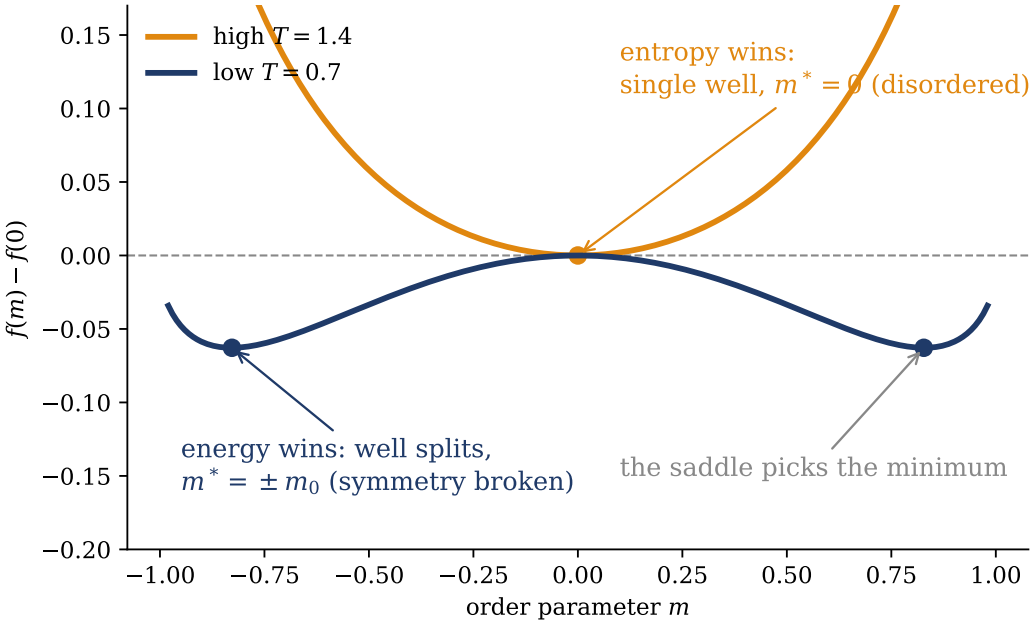

Figure 2.3: The free-energy saddle, drawn for the all-to-all ferromagnet $\varepsilon(m) = -m^2/2$ (so $T_c = 1$), as $f(m) - f(0)$ against the order parameter m. At high temperature ($T = 1.4$, orange) entropy dominates and the free energy has a single well at $m^* = 0$: the disordered state. At low temperature ($T = 0.7$, navy) energy dominates and the well splits into two symmetric minima at $m^* = \pm m_0 \approx \pm 0.83$: the saddle must pick one — spontaneous symmetry breaking, and mean-field theory in embryo. Module 2 performs this calculation on real energies.

n	$\log n!$ (exact)	$n\log n - n$	$+\frac{1}{2}\log(2\pi n)$	rel. err. (leading)	rel. err. (corrected)
2	0.693	−0.614	0.652	—	6×10^{-2}
5	4.787	3.047	4.771	0.36	3×10^{-3}
10	15.104	13.026	15.096	0.14	5×10^{-4}
100	363.739	360.517	363.739	0.9%	2×10^{-6}

The left panel of Figure 2.4 plots the trend. Two readings. First, the leading form — the one we will routinely use — is within 1% by $n = 100$ and improves steadily; for the $n \sim 10^2$–10^{23} of our applications it is quantitatively excellent. Second, and more instructive: the error of the leading form *is* the $\frac{1}{2}\log(2\pi n)$ term — literally the Gaussian prefactor of Equation 2.2 — and restoring it nails the answer to 5×10^{-4} already at $n = 10$. The correction we usually drop is not mysterious; it is the width of the peak, and we know its formula.

Binomial concentration — the 2^N wall revisited. Now the concentration claim, on the simplest possible system: N unbiased binary units, every configuration equally likely. The fraction of configurations with mean activity m is governed by an exponential of the entropy deficit, $P(m) \propto e^{-ND(m\|1/2)}$ — in the language of *large deviations*, $D(m\|1/2)$ is the *rate function*, here used informally — peaked at $m = 1/2$ with standard deviation

$$\sigma_m = \frac{1}{2\sqrt{N}}.$$

Put numbers on it: for $N = 10^2$, 10^4, and 10^6, the width is $\sigma_m = 0.05$, 0.005, and 5×10^{-4}, respectively.

The right panel of Figure 2.4 draws the collapse. For $N = 100$ the sum has $2^{100} \approx 1.3\times10^{30}$ terms — already unsummable — yet the distribution over the order parameter is a bump of width 0.05; at $N = 10^4$ it is a spike. All but a vanishing fraction of the 2^N configurations are crowded into a sliver of width $1/\sqrt{N}$ around the dominant value of m — which is exactly why keeping only the largest term is legitimate, and why its error is the Gaussian factor we computed rather than something unknown.

> *We never compute Z; we compute $\frac{1}{N}\log Z$ — and one term is enough.* The 2^N terms are all still there, but almost every one of them sits in a sliver of width $1/\sqrt{N}$ around the saddle. Evaluating the saddle is $\mathcal{O}(1)$; the other $2^N - 1$ terms are a correction we compute analytically and never enumerate.

For the mean-field energy above there is a single saddle. For real energies — short-range, structured, disordered — the landscape $f(m)$ grows *many* saddles, and deciding which ones to trust, and how to correct them, is the central question of Module 2: the TAP equations, belief propagation and message passing, and the replica method are three answers to that one question.

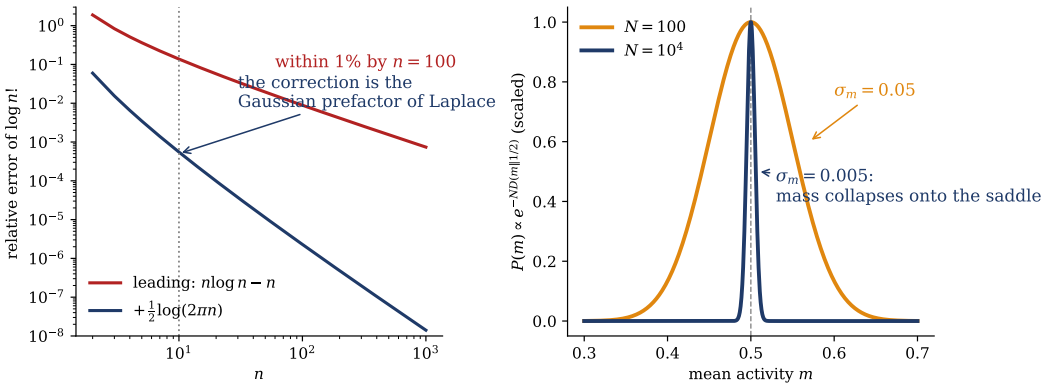

Figure 2.4: Left: relative error of $\log n!$ for the leading Stirling form $n\log n - n$ (red) and the corrected form with the $\frac{1}{2}\log(2\pi n)$ term (navy) — the leading form is within 1% by $n = 100$, and the correction, which is precisely the Gaussian prefactor of Laplace's method, is accurate to 5×10^{-4} already at $n = 10$. Right: the distribution $P(m) \propto e^{-ND(m\|1/2)}$ of the mean activity of N unbiased binary units, for $N = 100$ ($\sigma_m = 0.05$, orange) and $N = 10^4$ ($\sigma_m = 0.005$, navy): the mass collapses as $1/\sqrt{N}$ onto the saddle.

2.7 Loose ends and scope

Two loose ends from the previous lecture, then the scope of what this lecture skipped.

Maximum entropy, generalized. The previous lecture's derivation used a single constraint — the mean energy — and produced a single multiplier, β. The generalization is immediate and worth stating result-first: maximizing entropy subject to K constraints $\langle f_k(x)\rangle = c_k$ yields the *exponential family*

$$p(x) \;\propto\; \exp\Big(-\sum_{k=1}^{K} \lambda_k f_k(x)\Big),$$

one Lagrange multiplier per constraint, by the same variational calculation run in parallel (Jaynes 1957). Energy-based models, logistic regression, and Markov random fields are all the same object under different constraint sets — a unification Module 1 uses from Week 2 on. We do not re-run the derivation; the previous lecture's single-constraint version plus "one multiplier per constraint" is the whole proof idea.

Scope. Three things this lecture deliberately did not do. We used the language of large deviations — rate function, concentration — informally, without the rigorous theory (Cramér, Varadhan); the informal version is all we need. We stayed with the real-variable Laplace method and never deformed a contour; the complex-analytic steepest descent is named so that you recognize it in the literature, not developed. And we asserted ensemble equivalence for mean values without proof, while flagging where it fails. Each of these has its rightful depth elsewhere; the corrections to the saddle point itself — the cases where the largest term is *not* enough — return as the spine of Module 2.

2.8 Outlook

Week 1 is complete, and its two lectures divide cleanly. The previous lecture gave the object and the obstruction: probability from energy, $p = e^{-\beta E}/Z$, and the 2^N wall that makes Z intractable. Today gave the toolkit: $\log Z$ as the generating function whose derivatives are physics, and the saddle-point chain — Gaussian integral, Laplace's method, Stirling's formula — that computes $\frac{1}{N}\log Z$ asymptotically, collapsing the free energy onto a minimization over one order parameter. One idea, stated three ways: *the largest term wins* — in counting (Stirling), in integrals (Laplace), and in the free energy (the saddle).

Week 2 opens Module 1 with the Hopfield model: the Ising energy repurposed as an associative memory, and the first time in this course that $E(x)$ is something we *design* rather than measure. Designed energies raise quantitative questions — how many patterns can a network of N neurons store? — and when Week 2 quotes the answer ($\alpha_c \approx 0.138$, as promised in the previous lecture) and Module 2 equips you with its logic, it will rest on exactly the calculation you saw today: write the free energy, find the saddle. Figure 2.3 reappears throughout the course in progressively richer forms.

The practitioner's summary: **when you see $e^{N\cdot(\text{something})}$, do not sum — find where the exponent is largest; and when you see a Gaussian, you can integrate it.**

Part I

Module 1 — The energy-based view

3 The Hopfield model: memory as attractors

Recommended reading

Core (~1 h). Hopfield (1982) — the founding paper, five pages, and remarkably readable. Read all of it: the content-addressable-memory framing, the storage prescription, the update rule, and even a numerical capacity estimate are already there. This is also the source for this week's tutorial warm-up, so the hour is spent twice. Alongside it, MacKay (2003), Ch. 42 — the inference-side telling of the same model, associative memory as an error-correcting code, with the convergence argument done cleanly (freely available online).
Optional background. Hertz, Krogh, and Palmer (1991), Ch. 2–3, *the* textbook treatment of the Hopfield model, whose logic this chapter follows closely; Amit (1989), Ch. 1–2, for the attractor picture developed at length by one of the people who solved the model; Nishimori (2001), Ch. 7, for associative memory in the spin-glass language Module 2 will need; and the energy-based-model sections of Mehta et al. (2019) for the machine-learning vocabulary this module now makes precise.
Prerequisite reminder. The Ising model — Hamiltonian, ferromagnetic ordering, single-spin-flip dynamics — is assumed from master's statistical mechanics. What is new is not the spin system; it is the *inversion of the problem*: choosing the couplings so that the minima sit where we want them, and reading relaxation as computation.

3.1 How can a physical system remember?

Start from an everyday observation so familiar it goes unnoticed. You recognize a friend from a quarter of a face in bad light. You complete "to be or not to —" without consulting an index. In both cases a *fragment of the content* retrieves the whole — no address, no lookup table, no exact match required. Whatever implements this in the brain, it is nothing like the random-access memory in a computer, where the address *is* the mechanism and a single corrupted address bit fetches garbage. The question this week answers is how a physical system — neurons, spins, anything with an energy — can remember in the first sense. The question carries weight: the 2024 Nobel Prize in Physics went to John Hopfield and Geoffrey Hinton "for foundational discoveries and inventions that enable machine learning with artificial neural networks," and the two objects the citation points at are precisely the subjects of Weeks 2 and 3, honored *as physics*. Today we meet the first of them, and we will find that it is an Ising model put to a new purpose.

The idea has a lineage, and we trace it because Hopfield's contribution was a *fusion*, not an invention from nothing — three ingredients from three fields that lay

separately for decades. McCulloch and Pitts (1943) modeled the neuron as a binary threshold unit and showed that networks of such units can compute logical functions: the *unit* of neural computation existed by 1943, but the treatment was logical, with nothing collective in it — no robustness, no storage, no dynamics. Hebb (1949) supplied a storage prescription in prose: neurons that fire together strengthen their connection — a learning rule stated decades before it had a mathematical home. Little (1974) made the physics mapping explicit, casting neural firing as an Ising system with synchronous, parallel dynamics. And the missing ingredient arrived from condensed matter, from a direction that had nothing to do with brains: the spin-glass theory of Edwards and Anderson (1975) and Sherrington and Kirkpatrick (1975) made *disordered couplings* a first-class object of statistical mechanics. Rugged energy landscapes with many metastable minima stopped being a pathology to average away and became a structure to study — and a landscape with many minima is exactly what a memory needs, one minimum per stored item.

Hopfield (1982) fused the three. Take McCulloch–Pitts threshold units; couple them with an Ising energy whose couplings are chosen by Hebb's rule from the data; update them *asynchronously*, one at a time, with symmetric couplings — and, as we prove below, the dynamics rolls strictly downhill on an energy landscape whose minima are the stored patterns. Memory becomes an attractor; recall becomes relaxation. Three years later, Amit, Gutfreund, and Sompolinsky (1985) did to this model what physicists do to any model — computed its phase diagram — and found that associative memory survives up to a sharp critical storage load. That calculation is the next lecture; the spin-glass machinery inside it, this course meets through its modern mean-field descendants in Module 2.

A memory is not a stored record but a dynamical attractor — and the same mathematics that makes a ferromagnet remember its magnetization direction can be engineered to remember a photograph.

Through Week 1, the energy $E(x)$ was *given*, and everything flowed forward from it: given E, maximum entropy delivered $p = e^{-\beta E}/Z$ (Section 1.4), and the saddle-point toolkit computed with it (Chapter 2). Today, for the first time, the direction inverts — we *choose* E — and one absence is worth noticing: nothing in this lecture mentions Z. The network below runs deterministically, at zero temperature; there are no probabilities to normalize. This is the first of only two stretches in the course — this week and Week 4 — where the partition function plays no role. The absence is brief: Week 3 heats this same network back up, and Z returns as the training obstruction the whole course pivots on.

> Take-home 1
>
> Associative memory is an energy landscape used as a computer: stored patterns are engineered minima, cues are initial conditions, retrieval is relaxation. The Hopfield network is the Ising model with its couplings chosen by data — the first energy-based model.

3.2 From lookup to attractor

Two dichotomies organize the week.

The first is **address versus attractor**. A conventional memory implements the map *address* → *contents*; it is exact, fast, and brittle — corrupt one address bit and you retrieve an unrelated word. An *associative* (or *content-addressable*) memory implements the map *partial contents* → *full contents*; the query is a corrupted fragment of the thing itself, and the corruption is not fatal but *repaired by the dynamics*. There is an engineering reading of this that MacKay develops and we only flag: an associative memory is an error-correcting code, with the basin of attraction playing the role of the code's correction radius (MacKay 2003).

The second dichotomy is the one that defines Module 1: **measured versus designed energy**. Physics traditionally receives its Hamiltonian from nature and asks what states it produces. Hopfield inverts the question: *given* the desired states $\{\xi^\mu\}$ — the memories — what energy has them as its minima? This inverse problem is precisely what Week 1 promised under the slogan "learning is shaping an energy landscape," and everything in this module is a progressively better answer to it: Hebb's one-shot rule today, gradient-trained Boltzmann machines in Week 3, modern Hopfield networks in Week 4.

Figure 3.1 draws the central picture, one the course keeps returning to: an energy landscape over configuration space, where each stored pattern sits in an engineered basin, a corrupted cue starts partway up a basin wall, and retrieval is the roll downhill. The picture rests on three concrete requirements: ① a dynamics that "rolls downhill" even though our variables are discrete spins with no gradient; ② a proof that this dynamics reaches a minimum and halts (the derivation of Section 3.4); ③ a recipe for placing the minima at the data — Hebb's rule. The figure also shows a minimum nobody ordered, reflecting a general property of designed landscapes rather than a drawing error.

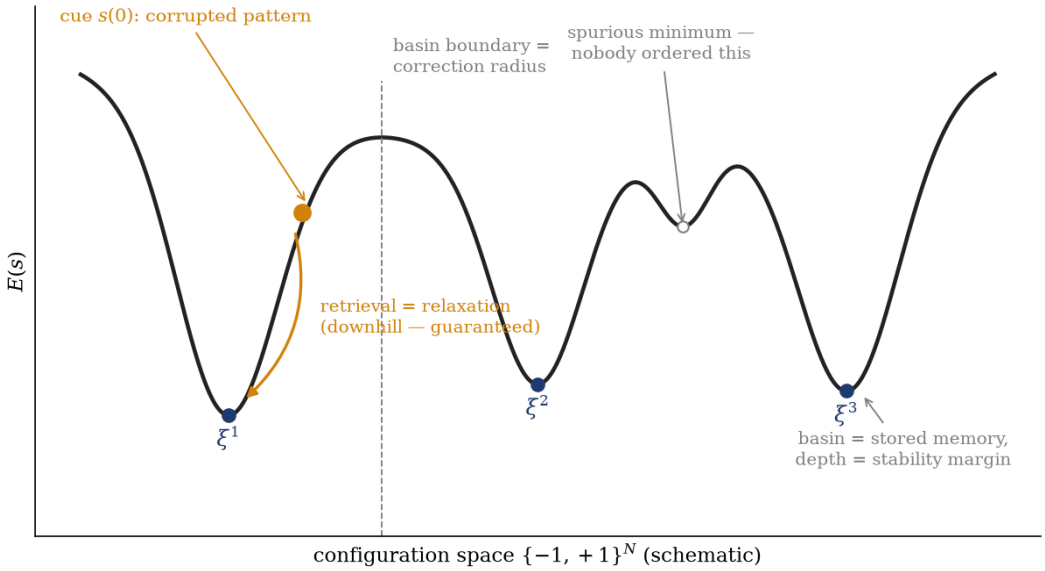

Figure 3.1: The central picture of Module 1: memory as an energy landscape. Stored patterns ξ^1, ξ^2, ξ^3 are engineered minima; a corrupted cue $s(0)$ is an initial condition partway up a basin wall; retrieval is relaxation to the basin's bottom, and the basin boundary sets how much corruption is correctable. The dynamics is guaranteed to run downhill (the Lyapunov theorem of Section 3.4) — but the landscape also contains *spurious* minima that nobody ordered, a point we return to at the end of the chapter.

3.3 The model: neurons as spins

Each object below carries two names — one from physics, one from machine learning — since the course works at their interface.

Take N binary *neurons* $s_i = \pm 1$, firing or quiet — which we read, with no modification whatsoever, as N Ising spins, up or down. The network state $s = (s_1, \dots, s_N)$ is a corner of the hypercube $\{-1, +1\}^N$: a *configuration* in exactly the sense of Week 1's symbol table, $x \leftrightarrow s$, and we will use s whenever the configuration is a neural state. Each neuron feels a *local field* — in machine-learning language, a *pre-activation* —

$$h_i = \sum_j J_{ij}\, s_j,$$

where the couplings J_{ij} are, in the other dialect, the *weights* of the network (w_{ij} in most of the ML literature; we keep J and the physics conventions). The dynamics is the simplest thing the local field permits: pick a neuron — *asynchronously*, one at a time, in random order — and align it with its field,

$$s_i \;\leftarrow\; \operatorname{sign}\Big(\sum_j J_{ij}\, s_j\Big) \tag{3.1}$$

asynchronous threshold dynamics

Thresholds are set to zero throughout — for the symmetric ± 1 patterns we store, they buy nothing — which also spares us a notation clash, since the course reserves θ for model parameters.

Two structural assumptions complete the model, and we state them as deliberate choices rather than habits, because the next section shows exactly what each one buys: the couplings are **symmetric**, $J_{ij} = J_{ji}$, and carry **no self-coupling**, $J_{ii} = 0$. Under these assumptions we associate to the network the energy

$$\boxed{E(s) = -\tfrac{1}{2} \sum_{i \neq j} J_{ij}\, s_i s_j} \tag{3.2}$$

Hopfield energy

— an infinite-range Ising Hamiltonian with heterogeneous couplings. As an *object*, nothing here is new to you; what is new is who chooses J. Note also what is absent: no β, no $p(x)$, no Z. The update rule Equation 3.1 is single-spin-flip dynamics at zero temperature — each move is the locally greedy choice — and the next section proves that these greedy local choices are, collectively, going somewhere definite.

Figure 3.2 shows the machine in action before we prove anything about it: three 8×8 patterns stored in a network of $N = 64$ neurons, one of them corrupted by flipping 30% of its pixels, and the corrupted cue handed to the dynamics Equation 3.1. The network repairs it completely.

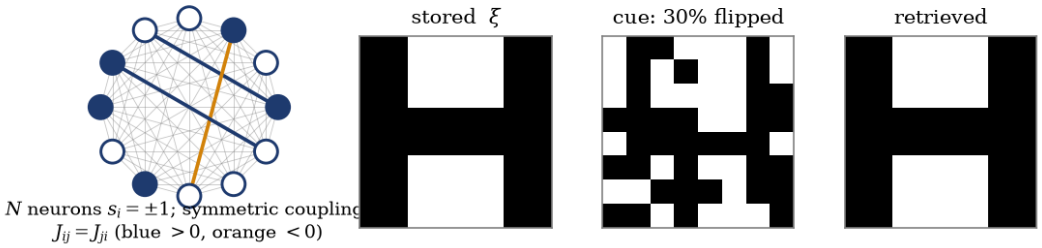

Figure 3.2: A Hopfield network doing its job. Left: the model — N fully connected binary neurons $s_i = \pm 1$ with symmetric couplings J_{ij} (spin = neuron, coupling = weight, local field = pre-activation). Right: one of three stored 8×8 patterns ($N = 64$, $P = 3$); the same pattern with 30% of its pixels flipped (overlap $m_0 \approx 0.41$); and the fixed point reached by the asynchronous dynamics Equation 3.1 — the memory, recovered exactly ($m = 1$).

3.4 The energy never increases

We owe the landscape cartoon a theorem. For continuous variables, "downhill" means gradient flow and $\dot{E} \leq 0$; for discrete spins there is no gradient, so what, precisely, guarantees descent? The derivation is short.

Let neuron k update by Equation 3.1: $s_k \to s_k' = \operatorname{sign}(h_k)$, all other spins held fixed. In the energy Equation 3.2, collect the terms containing s_k. The variable s_k appears twice in the double sum — once as $i = k$ and once as $j = k$ — and the *symmetry* $J_{ij} = J_{ji}$ merges the two appearances into a single term:

$$E(s) = -\,s_k \underbrace{\sum_{j \neq k} J_{kj}\, s_j}_{= h_k} + (\text{terms without } s_k),$$

where we used $J_{kk} = 0$ so that the sum defining h_k is the full local field — and, crucially, h_k *does not depend on* s_k *itself*: it is the same number before and after the update. Since the spectator terms cancel in the difference, the energy change of the single update is exactly

$$\Delta E \;=\; E' - E \;=\; -\,(s_k' - s_k)\, h_k.$$

Two cases exhaust the possibilities. ① If s_k is already aligned with its field, $s_k = \operatorname{sign}(h_k)$, the update does nothing: $s_k' = s_k$ and $\Delta E = 0$. ② If it is misaligned, the update flips it, $s_k' = -s_k = \operatorname{sign}(h_k)$; then $s_k' - s_k = 2s_k'$ and $s_k' h_k = |h_k|$, so

$$\Delta E = -2\,|h_k| \;<\; 0.$$

In both cases,

$$\Delta E \leq 0 \quad \text{under asynchronous updates with symmetric } J, \;\; J_{ii} = 0, \tag{3.3}$$

Lyapunov property

with equality precisely when the chosen spin was already aligned.

Convergence follows at once. The energy is bounded below — crudely, $E \geq -\frac{1}{2}\sum_{i \neq j} |J_{ij}|$ — and the configuration space is finite, so E takes finitely many values; every strict move lowers E by a positive amount, so no configuration can ever recur after a strict move. The dynamics must therefore reach, in finite time, a state from which *no* single-spin update changes anything: a **fixed point**, satisfying $s_i = \text{sign}(h_i)$ for every i — a local minimum of E against all single flips.

The cartoon is now a theorem: the network state flows monotonically downhill on E and halts at a local minimum. E is a Lyapunov function *for the dynamics — the dynamicist's certificate that a system computes something definite. If we can place the minima, we have built a content-addressable memory.*

Both assumptions are essential. *Symmetry* is what made the energy's s_k-terms collapse onto the same h_k that the update rule consults — with asymmetric couplings, the quantity the update aligns with and the quantity the energy penalizes are different objects, no energy function exists at all, and the dynamics is free to cycle or wander chaotically. *Asynchrony* ensured that only one term changes at a time; the synchronous variant — all spins updated in parallel, which is Little's model (Little 1974) — loses the guarantee and admits stable period-2 oscillations, in which two configurations toss the network back and forth forever. And $J_{ii} = 0$ is not cosmetic either: a self-coupling would put a $J_{ii}s_i$ term into the field $h_i = \sum_j J_{ij}s_j$ that spin i consults, biasing the spin toward its own current value — the update would then depend on s_i itself, breaking the very alignment of field and energy gradient that the descent argument above relied on. In the extreme of strong positive J_{ii}, every configuration freezes into a fixed point — which is to say, no memory at all.

Figure 3.3 shows the theorem at work on a real network — $N = 1000$ neurons, twenty stored patterns, a cue with 30% of its bits flipped: the energy descends as a monotone staircase, one small drop per accepted flip, and locks onto a plateau within a single sweep while the overlap with the stored pattern climbs to one.

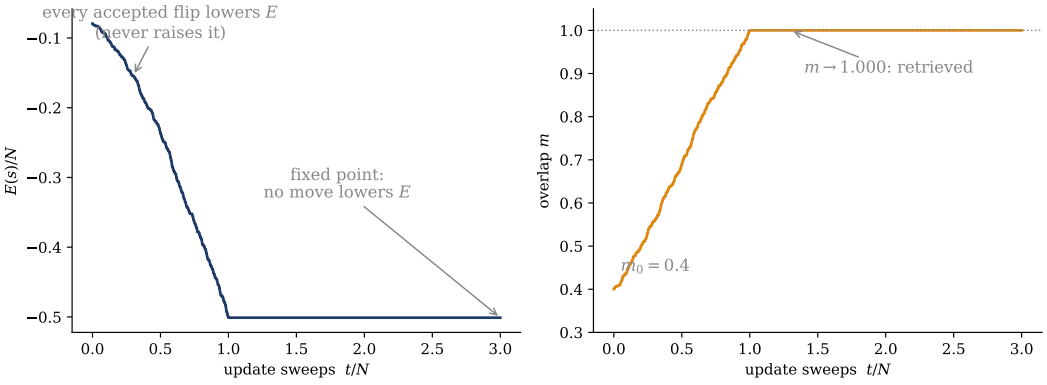

Figure 3.3: The Lyapunov theorem at work: a Hopfield network with $N = 1000$ neurons and $P = 20$ stored patterns ($\alpha = 0.02$), cued with a stored pattern corrupted in 30% of its bits ($m_0 = 0.4$). Left: the energy per neuron under asynchronous updates — a monotone staircase, $\Delta E \leq 0$ at every single step, reaching a fixed point within one sweep (here $E/N \to -0.50$). Right: the overlap m with the cued memory climbing from 0.4 to 1.000 — exact retrieval.

Trap

Symmetry and asynchrony are **load-bearing, not stylistic**. Drop symmetry and no energy function exists — the dynamics can cycle or run chaotically. Keep symmetry but update synchronously and period-2 limit cycles survive. The energy-landscape picture of neural computation rests on these two assumptions — and note the scope: real cortical synapses are asymmetric, so biology sits outside the theorem.

3.5 Hebbian storage

The theorem guarantees arrival at *a* minimum; we now place the minima. Given P patterns $\xi^\mu \in \{-1,+1\}^N$, $\mu = 1,\dots,P$ — the memories, drawn for the theory below as independent unbiased random signs — choose the couplings as a sum of outer products:

$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{P} \xi_i^\mu \, \xi_j^\mu \quad (i \neq j), \qquad J_{ii} = 0. \tag{3.4}$$

Hebb rule

This is Hebb's postulate (Hebb 1949) — neurons that fire together, wire together — finally given an equation: every pair of units that agrees in a pattern gets its coupling strengthened, every pair that disagrees gets it weakened, pattern by pattern, and the $1/N$ keeps the local fields of order one. The rule is symmetric by construction, so the theorem of Section 3.4 applies for free. And note what the rule is in machine-learning terms: *one-shot learning*. The weights are written down from the data in a single pass — no loss function, no gradient, no iteration. In Week 3, learning becomes gradient *descent* on a likelihood — and the gradient is precisely where Z enters.

Does it work? One pattern, verified. Store a single pattern ($P = 1$) and evaluate the local field at the pattern itself, $s = \xi$:

$$h_i \;=\; \frac{1}{N}\sum_{j\neq i} \xi_i \, \xi_j \, \xi_j \;=\; \frac{N-1}{N}\,\xi_i \;\approx\; \xi_i.$$

Every spin sits aligned with its field, with a uniform stability margin of order one: ξ is a fixed point of Equation 3.1, hence by Section 3.4 a local minimum of the energy. (So, immediately, is the flipped pattern $-\xi$: the energy Equation 3.2 is even under a global flip $s \to -s$ — our first minimum that nobody ordered. We collect the others below.)

More is true, and it introduces the week's order parameter. Define the *overlap* of the network state with a pattern,

$$m \;=\; \frac{1}{N}\sum_i \xi_i \, s_i \;\in\; [-1,1],$$

overlap: the retrieval order parameter

which counts agreement: $m = 1$ is perfect retrieval, $m = 0$ is chance, and $m_0 = 1-2f$ for a cue corrupted in a random fraction f of its bits. For $P = 1$ the local field at *any* state collapses onto the overlap,

$$h_i \;=\; \frac{1}{N}\,\xi_i \sum_{j\neq i} \xi_j s_j \;=\; \xi_i\Big(m - \frac{s_i\xi_i}{N}\Big) \;\approx\; \xi_i\, m,$$

so a single sweep of Equation 3.1 sets every spin to $\operatorname{sign}(\xi_i m) = \xi_i$ whenever $m > 0$: *any* cue with positive initial overlap retrieves the memory in one sweep, and the basin of ξ is the entire $m > 0$ half of the hypercube — 2^{N-1} configurations — with the mirror image $-\xi$ owning the other half. You have seen this m before: it is exactly the magnetization that indexed the free-energy saddle of Section 2.5, now reinterpreted as *how retrieved* a memory is. The order-parameter language of Week 1 carries over directly, and the next lecture makes quantitative use of it.

Many patterns: signal and crosstalk. With P patterns stored, evaluate the aligned local field at one of them, $s = \xi^\mu$, splitting the Hebb sum into the term $\nu = \mu$ and the rest:

$$h_i\,\xi_i^\mu \;=\; \underbrace{\frac{N-1}{N}}_{\text{signal}\,\approx 1} \;+\; \underbrace{\frac{1}{N}\sum_{\nu\neq\mu}\sum_{j\neq i} \xi_i^\mu\,\xi_i^\nu\,\xi_j^\nu\,\xi_j^\mu}_{\text{crosstalk}\,\equiv\, C_i}.$$

The stored pattern contributes a clean unit signal — that is the $P = 1$ calculation again. Everything else contributes *crosstalk*: a sum of $(P-1)(N-1)$ terms, each $\pm 1/N$ with a pseudo-random sign, since the patterns are independent. By the central limit theorem, C_i is approximately Gaussian with mean zero and variance

$$\operatorname{Var}(C_i) \;=\; \frac{(P-1)(N-1)}{N^2} \;\approx\; \frac{P}{N} \;\equiv\; \alpha,$$

where α is the *storage load*, the number of patterns per neuron. The entire question of memory capacity is now visible in one line: a bit of the stored pattern is stable as long as its signal beats its noise, and the competition is between 1 and a Gaussian of width $\sqrt{\alpha}$. How that competition ends — and why it ends in a sudden collapse rather than a graceful fade — is the next lecture; the example below is the first look at the numbers.

> Take-home 2
>
> The Hebb rule $J_{ij} = \frac{1}{N}\sum_\mu \xi_i^\mu \xi_j^\mu$ writes the memories directly into the couplings — learning in one shot, no gradient. Each stored pattern feels a unit signal plus Gaussian crosstalk of width $\sqrt{P/N}$ from all the others: memory works until the stored patterns start to interfere.

3.6 Example: retrieval, then interference

Retrieval from heavy corruption ($P = 1$). Store one pattern in $N = 1000$ neurons and corrupt it heavily: flip a random 40% of the bits, leaving the initial overlap $m_0 = 1 - 2 \times 0.4 = 0.2$. By the field formula of Section 3.5, every neuron

feels $h_i \approx 0.2\,\xi_i$ — a weak field, but pointing the *right way* at every single site, because with one pattern there is no noise to point it wrong. One asynchronous sweep restores $m = 1$ exactly. Running the actual dynamics confirms the table:

fraction flipped f	$m_0 = 1 - 2f$	after one sweep	reading
0.10	0.80	$m = 1$	easy repair
0.25	0.50	$m = 1$	half the bits suspect — fine
0.40	0.20	$m = 1$	the case of Figure 3.2 and Figure 3.3
0.49	0.02	$m = 1$	still inside the basin
0.50	0.00	knife-edge	no signal at all
0.60	−0.20	$m = -1$	converges to the anti-pattern $-\xi$

The stored pattern sits at energy $E(\xi)/N = -(N-1)/2N \approx -1/2$ per neuron — extensively deep. The lesson of the table calibrates what comes next: with a single stored pattern, retrieval is robust — the basin is half of configuration space — so the dynamics is never what limits an associative memory; interference is.

Interference: the first-sweep error rate. Now store P patterns and read the aligned field on one of them: signal 1, Gaussian crosstalk of width $\sqrt{\alpha}$. A bit comes out wrong on the first sweep when the crosstalk overwhelms its signal, $C_i < -1$, which happens with probability

$$P_{\text{err}} \;=\; \Pr\left[C_i < -1\right] \;=\; \tfrac{1}{2}\,\operatorname{erfc}\!\Big(\frac{1}{\sqrt{2\alpha}}\Big).$$

The Gaussian tail sweeps this quantity across twenty orders of magnitude over the loads in the table:

$\alpha = P/N$	P at $N = 1000$	P_{err}	wrong bits per retrieved pattern	reading
0.01	10	$\sim 10^{-23}$	0	interference invisible
0.05	50	4×10^{-6}	0.004	still essentially perfect
0.10	100	8×10^{-4}	0.8	first wrong bits — and errors feed back
0.138	138	3.6×10^{-3}	3.6	the edge (why *here*? — next lecture)
0.20	200	1.3×10^{-2}	13	avalanche territory

Figure 3.4 shows the same story measured rather than estimated: the empirical distribution of the aligned local field across the sites of a stored pattern, in a network

of $N = 2000$ neurons, at two loads. At $\alpha = 0.05$ the Gaussian sits comfortably to the right of zero and, in the simulation, *not one* of the 2000 bits is unstable. At $\alpha = 0.2$ a visible tail has leaked past zero — 1.6% of bits in this realization, against the Gaussian estimate of 1.3% — and each of those bits, once flipped, adds noise to the fields of all the others.

That feedback is what the table leaves out. The erfc column is a *first-sweep, no-feedback* estimate; it does not by itself locate the storage limit. Flipped bits degrade the fields, which flip more bits — and whether the avalanche self-limits or runs away is a self-consistency question, not a one-line Gaussian one. Its answer is one of the celebrated results of the field: retrieval survives up to a sharp critical load $\alpha_c \approx 0.138$ and then collapses discontinuously — memory dies as a phase transition, not a fade (Amit, Gutfreund, and Sompolinsky 1985, 1987). The next lecture builds the self-consistent argument; in the tutorial you will cross α_c yourself and watch it happen. (For calibration: Hopfield (1982) already reported, numerically, that recall degrades severely beyond $P \approx 0.15\,N$ — close to the sharp answer computed three years later.)

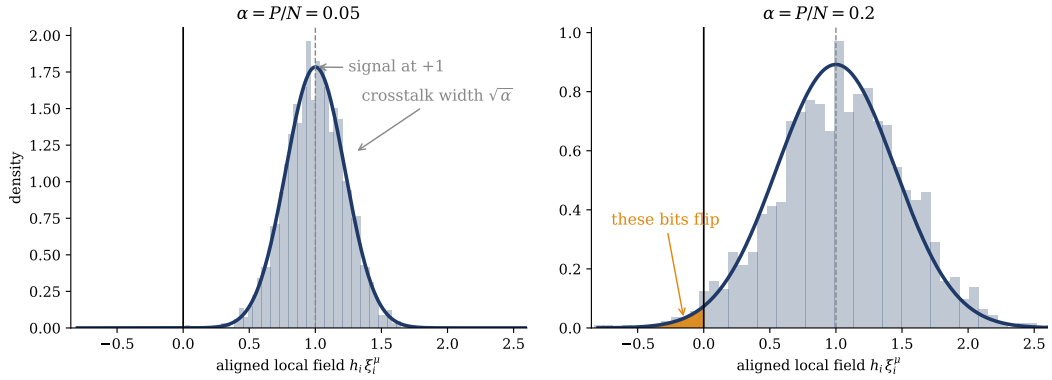

Figure 3.4: Signal versus crosstalk, measured. Histograms of the aligned local field $h_i \xi_i^\mu$ over the $N = 2000$ sites of a stored pattern, with the Gaussian $\mathcal{N}(1, \alpha)$ overlaid. Left, load $\alpha = P/N = 0.05$: the crosstalk is narrow, the entire distribution sits right of zero, and no bit is unstable. Right, $\alpha = 0.2$: the width $\sqrt{\alpha}$ has grown until a tail (shaded) leaks past zero — those bits flip on the first sweep (1.6% measured here versus 1.3% from the Gaussian estimate), and their errors feed back into every other field. Where this avalanche becomes fatal is the next lecture's calculation.

Energy stops being something we measure and becomes something we design. That is the banner of Module 1 — and everything that follows in this course, through the Boltzmann machine to the diffusion model, inherits it.

One practical pointer, because this week's tutorial is built on exactly this material: the warm-up card (due the evening before the next lecture) works from this lecture and from Hopfield (1982), and it primes the field decomposition above — signal against crosstalk. The hard problem in the tutorial builds on it.

Take-home 3

One stored pattern owns half of configuration space, so dynamics never limits an associative memory — interference does. Capacity is a competition between

a unit signal and crosstalk of width $\sqrt{P/N}$: quantified in the next lecture, experienced in the tutorial.

3.7 What the landscape contains besides your data

The landscape contains more minima than we stored, and the extras matter for what follows.

Three families of uninvited minima populate the landscape. ① The **flipped patterns** $-\xi^\mu$, exactly degenerate with the originals by the global symmetry $s \to -s$ of Equation 3.2 — harmless in practice (fix one bit's meaning and the ambiguity is gone) but a genuine doubling of the attractor count. ② **Mixture states**: symmetric blends of odd numbers of patterns, the simplest being the three-pattern majority $s_i = \mathrm{sign}(\xi_i^1 + \xi_i^2 + \xi_i^3)$, which for random patterns has overlap $1/2$ with each of its three parents and is a genuine local minimum — the network's version of a confused recollection that merges three memories. ③ At larger loads, **spin-glass states**: minima essentially uncorrelated with anything stored, the rugged debris of the crosstalk. The lesson: *a designed landscape is still a landscape — you specify the minima you want, and the geometry supplies others for free.* There is even a first argument here for Week 3's turn to finite temperature: the spurious minima are typically *shallower* than the retrieval states, so a little thermal noise destabilizes them before the memories — the same temperature that brings back Z and the Boltzmann distribution also cleans the landscape.

A biological disclaimer is in order, since the model uses the word "neuron." Real synapses are not symmetric; real neurons are not binary; real updates are not asynchronous-random; and Hopfield (1984) himself showed the collective behavior survives at least the second idealization, in a graded-response (continuous) variant we merely name. The model's value is not fidelity to cortex — it is that it is *minimal, solvable, and generalizable*: the entire modern edifice of energy-based and diffusion models inherits its shape, which is exactly the sense in which the 2024 Nobel citation calls it foundational.

Trap

Convergence is not correctness. The Lyapunov theorem guarantees arrival at *some* minimum; whether it is the cued memory, its mirror image, or a mixture is a question about basin geometry, which the theorem does not touch. And do not invert Take-home 1: "stored patterns are minima" is true; "minima are stored patterns" is false.

3.8 Outlook

The next lecture takes the question this lecture left open — *how many patterns fit?* — and answers it quantitatively: the signal-to-noise argument made self-consistent, the error avalanche tamed or not, the strict-stability estimate $P_{\max} \approx N/(2\ln N)$ for perfect recall of every bit, and the celebrated retrieval boundary $\alpha_c \approx 0.138$ of Amit, Gutfreund, and Sompolinsky (1985) — with the caveat that the full derivation runs on replica machinery this course quotes rather than builds; Module 2 acquires its

modern mean-field descendants, and the free-energy saddle of Section 2.5 returns there in earnest. In the tutorial, you will load a network past α_c and watch memory die as a phase transition.

Then Week 3 brings back the partition function. Today's network was deterministic and stored its patterns by fiat. Do the two natural next things — heat it up, so that the update rule becomes stochastic and the stationary law becomes exactly Week 1's Boltzmann distribution over the designed energy; and ask it to *learn* J by maximum likelihood rather than by Hebb's fiat — and the training gradient acquires a term $\nabla_J \log Z$. That term is intractable, it is the negative phase of the Boltzmann machine, and it is the central obstruction of energy-based learning. The energy landscape was designed today; from Week 3 on, training it requires computing Z.

4 Storage capacity: how memory dies

Recommended reading

Core (~1 h). Hertz, Krogh, and Palmer (1991), Ch. 2 — the storage-capacity section is the cleanest elementary treatment in print, and this chapter follows its logic closely. Alongside it, Amit, Gutfreund, and Sompolinsky (1985) — four pages, the primary source for today's headline number. Read it for the *results* and the phase-diagram figure, not the method: the replica machinery inside is spin-glass theory proper — Module 2 equips you with its modern mean-field descendants — and learning to extract the conclusions of a paper whose methods you do not yet command is a physicist's skill worth practicing on purpose.
Optional background. Amit (1989), the capacity and phase-diagram chapters, for the full attractor-network statistical mechanics by one of the people who did it; Nishimori (2001), Ch. 7, for the same analysis in the notation Module 2 will adopt; MacKay (2003), Ch. 42, for the information-theoretic side of the question — how many *bits* does the network actually store?
Prerequisite reminder. Everything from Chapter 3 — the model, the Lyapunov theorem, the Hebb rule, the overlap m, and the signal-plus-crosstalk split — plus the central limit theorem and the Gaussian toolkit of Chapter 2; the Gaussian *tail* asymptotic we quote where it is first needed. The replica method is explicitly *not* assumed: where the full derivation needs it, we say so and post the debt to Module 2.

4.1 Is $0.15\,N$ the answer?

We start from where the last lecture left us, in three lines. We *design* the energy: Hebb's rule Equation 3.4 writes P patterns into the couplings in one shot. Descent is a theorem: for symmetric couplings and asynchronous updates, the dynamics rolls monotonically downhill and halts in a local minimum (Section 3.4). And at a stored pattern, every neuron's aligned local field splits as

$$h_i\,\xi_i^\mu \;=\; 1 \;+\; C_i,$$

a clean unit signal plus a crosstalk term C_i that we named and deliberately did not compute (Section 3.5). What remains is the computation of C_i — and of what the network does about it.

There is also a number on the table, and it came from your own reading. Hopfield (1982) reports, from simulations of networks of $N = 100$ neurons, that recall degrades severely once the number of stored patterns exceeds about $0.15\,N$. The obvious question is: *is* 0.15 *the answer?* It cannot be accepted as it stands, for two reasons that turn out to carry real physics. First, we do not yet know *which question* it

answers. "How many patterns fit?" splits, on inspection, into two demands with no obligation to agree: **strict stability** — every bit of every stored pattern sits aligned with its field, no exceptions, so the memories are exact fixed points — and **retrieval** — the pattern is recalled up to a small dressing of wrong bits, and cues in its basin still flow to it. The first is a worst-case demand, governed by the *largest* of NP random crosstalk pulls; the second is a typical-case demand, governed by the bulk of the crosstalk distribution. You have met this split before in other fields: worst-case versus average-case in the analysis of algorithms, minimum-distance versus Shannon capacity in coding theory (MacKay 2003). The two answers here differ not by a constant but by a factor of $\ln N$.

Second, we do not know *how* memory fails when it fails. The natural guess — more patterns, gradually more wrong bits, memory dimming like an old photograph — is wrong, and wrong in an instructive way. The truth is a cliff: retrieval survives, nearly intact, up to a sharp critical load, and then collapses discontinuously. Understanding why it is a cliff is the substance of the analysis below — the course's first genuine phase transition, arrived at not in a magnet but inside a learning machine.

Today we compute what interference costs, and find that the answer depends on what we demand — perfection is sublinear, tolerance is extensive — and that when Hebbian memory fails, it fails all at once.

The partition function stays out of sight: everything until the final section happens at $T = 0$. The difficulty is *interference*, and the tool that quantifies it is the central limit theorem. The lecture also does to Hopfield's model what Amit, Gutfreund, and Sompolinsky (1985) did — computes its phase diagram. The diagram has a temperature axis, and the final section explains why Week 3 walks up that axis on purpose.

Take-home 1

"How many patterns fit?" is two questions. Demanding every stored bit be exactly stable gives $P_{\max} \approx N/(2\ln N)$ — perfection costs a logarithm. Tolerating a percent of wrong bits gives extensive capacity $P_{\max} = \alpha_c N$ with $\alpha_c \approx 0.138$. Capacity is not a property of the network alone; it is a property of the network *plus the demand placed on it.*

4.2 The statistics of interference

The first task is the distribution of the crosstalk. Recall its definition — at stored pattern ξ^μ, the aligned field on site i carries, besides the unit signal,

$$C_i \;=\; \frac{1}{N}\sum_{\nu\neq\mu}\sum_{j\neq i}\xi_i^\mu\,\xi_i^\nu\,\xi_j^\nu\,\xi_j^\mu \;\equiv\; \frac{1}{N}\sum_{\nu\neq\mu}\sum_{j\neq i} t_{\nu j},$$

a sum of $(P-1)(N-1)$ terms $t_{\nu j} = \pm 1$, each a product of four independent, unbiased signs. The bookkeeping takes one careful paragraph, because "pseudo-random" needs justification. Each term has mean zero: for $\nu \neq \mu$ and $j \neq i$ the four factors involve at least one sign that appears exactly once, and its average kills the product. Better, the terms are *pairwise uncorrelated*: for two distinct index pairs $(\nu, j) \neq (\nu', j')$, the expectation $\langle t_{\nu j}\, t_{\nu' j'}\rangle$ always contains some pattern entry to an

odd power — if $\nu \neq \nu'$ the two patterns are independent and each factor averages to zero separately; if $\nu = \nu'$ but $j \neq j'$ the product contains $\xi_j^\mu \xi_{j'}^\mu$ times signs of pattern $\nu \neq \mu$, and again averages to zero. So the variance of the sum is *exactly* the sum of the variances — no approximation yet:

$$\mathrm{Var}(C_i) \;=\; \frac{(P-1)(N-1)}{N^2} \;\approx\; \frac{P}{N} \;=\; \alpha.$$

(Full independence does fail at higher orders — quadruples of terms can share signs — but these correlations affect the distribution only at relative order $1/N$.) The central limit theorem then does the rest: for large N at fixed load α,

$$C_i \;\sim\; \mathcal{N}(0,\,\alpha), \qquad \alpha = \frac{P}{N}. \tag{4.1}$$

crosstalk statistics

The physical reading: *the signal is* 1 *no matter how many patterns are stored; the noise grows as* $\sqrt{P/N}$ *— and interference depends on* P *and* N *only through their ratio.* The load α is the model's control parameter, and all quantities below are functions of it.

A stored bit comes out wrong on the first update sweep when the crosstalk overwhelms its signal, $1 + C_i < 0$ — a fluctuation of $1/\sqrt{\alpha}$ standard deviations. The Gaussian tail integral (the Gaussian integral of Section 2.4, at work once more) gives the *first-sweep bit-error probability* you already met empirically in Section 3.6:

$$P_{\mathrm{err}} \;=\; \Pr[\,C_i < -1\,] \;=\; \tfrac{1}{2}\,\mathrm{erfc}\Big(\frac{1}{\sqrt{2\alpha}}\Big). \tag{4.2}$$

first-sweep error rate

Figure 4.1 plots it, and the plot teaches two things at once. The first is the steepness of a Gaussian tail: between $\alpha = 0.01$ and $\alpha = 0.2$ — a factor of twenty in load — the error probability climbs by twenty-one orders of magnitude, from 10^{-23} to 10^{-2}. Steepness like this is why the capacity question has a sharp answer at all: whatever criterion we impose, the crossover from "satisfied absurdly well" to "violated badly" is compressed into a narrow window of α. The second lesson is a warning, drawn on the figure deliberately: the curve is perfectly smooth through $\alpha_c \approx 0.138$, the load where memory actually dies. Nothing in the first-sweep error rate marks the edge; that mismatch is the central puzzle of the analysis.

4.3 Perfection costs a logarithm

Now impose the strong demand: *strict stability.* Every stored pattern should be an exact fixed point of the dynamics — all N bits of all P patterns aligned with their fields, so that a memory, once retrieved, is letter-perfect. How many patterns does that permit?

Consider first a single stored pattern and ask that all N of its bits hold. Bit failures are (to leading order) independent rare events with probability P_{err} each, so the expected number of unstable bits is NP_{err}, and the pattern is exactly stable, with high probability, as long as

$$N\,P_{\mathrm{err}} \;\lesssim\; 1.$$

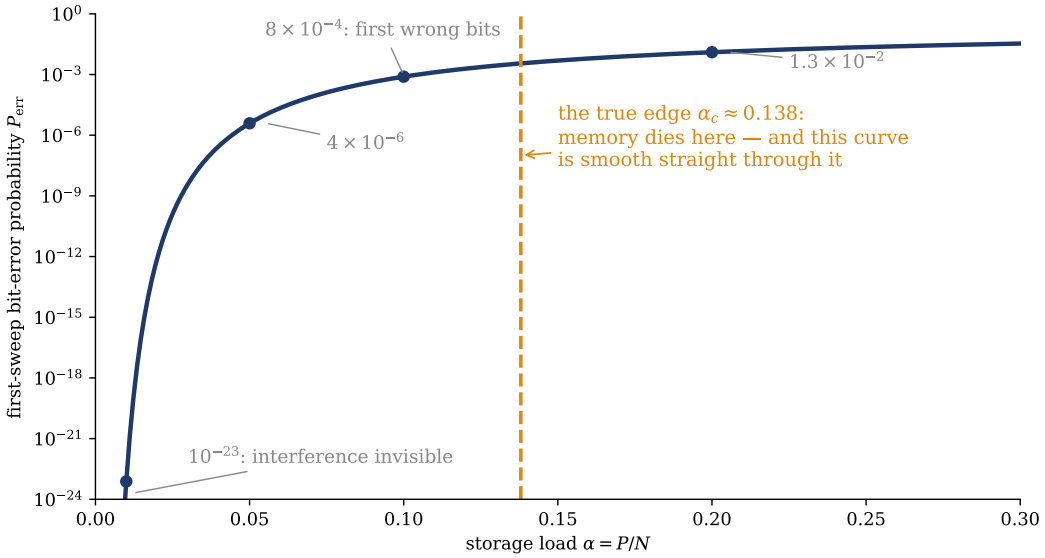

Figure 4.1: The first-sweep bit-error probability $P_{\text{err}} = \frac{1}{2}\operatorname{erfc}(1/\sqrt{2\alpha})$ of Equation 4.2, on a logarithmic scale. The Gaussian tail is remarkably steep — twenty-one orders of magnitude between $\alpha = 0.01$ and $\alpha = 0.2$ — which is why capacity questions have sharp answers. But note the dashed line: the true storage limit $\alpha_c \approx 0.138$, where retrieval collapses, leaves no trace whatsoever on this curve. First-sweep error rates cannot locate the transition; the transition is made of feedback.

Insert the Gaussian tail asymptotic — for $x \gg 1$, $\frac{1}{2}\operatorname{erfc}(x/\sqrt{2}) \approx e^{-x^2/2}/(x\sqrt{2\pi})$, here with $x = 1/\sqrt{\alpha}$ — and keep only the exponentially dominant balance:

$$N\sqrt{\frac{\alpha}{2\pi}}\, e^{-1/2\alpha} \;\approx\; 1 \qquad \Longrightarrow \qquad \frac{1}{2\alpha} \;\approx\; \ln N + \mathcal{O}(\ln\ln N),$$

where the prefactor's logarithm is subleading and swept into the correction. Solving for the load,

$$\boxed{\alpha_{\text{strict}} \approx \frac{1}{2\ln N}, \qquad P_{\max} \approx \frac{N}{2\ln N}} \tag{4.3}$$

strict-stability capacity

— *sublinear* in N. If instead we demand that *every* pattern be letter-perfect simultaneously, the union runs over all $NP \approx \alpha N^2$ bits, the balance becomes $\frac{1}{2\alpha} \approx \ln(NP) \approx 2\ln N$, and the capacity halves again to $P_{\max} \approx N/(4\ln N)$. Both statements — with the constants, the basin-of-attraction refinements, and full rigor in place of our union-bound physics — are the content of a celebrated information-theory paper (McEliece et al. 1987); our two-line estimate lands exactly on their scalings.

Demanding zero errors makes capacity sublinear: the price of perfection is a logarithm, and the logarithm counts how many trials you insist survive. The mechanism recurs across physics and statistics: the *maximum* of M independent Gaussian pulls grows like $\sqrt{2\ln M}$, so the *worst* bit in the network — not the typical one — sets the rule, and worst cases cost logarithms. At $N = 1000$: about 72 patterns (36 if every pattern must be perfect). Hopfield's 150 is nowhere in sight — so either his

simulations were wrong, or exact fixed points are the wrong demand. They were not wrong.

Notice what strict stability does *not* say. A pattern that fails it has, at $\alpha = 0.1$, roughly one misaligned bit per thousand: the dynamics flips that bit, the state comes to rest one or two spin flips away from the stored pattern, and any user of the memory would call the recall a success. Strict stability is a bookkeeping demand, not a statement about whether the memory works. The working question — does a fixed point *near* ξ^μ survive, with a basin around it? — is a question about the typical crosstalk and about what the first few errors do next. That question has more physics in it.

> Trap
>
> **A small error rate is not safety.** $P_{\text{err}} = 10^{-4}$ sounds negligible, but there are NP stored bits: rare events at a fixed rate become certainties at scale, which is why the strict criterion collapses from "essentially never fails" to "fails somewhere, always" over a narrow range of α. Extreme-value reasoning — the worst of many — obeys different arithmetic than typical-value reasoning. Know which question you are asking.

4.4 The collapse

So relax the demand and follow the dynamics. Start the network *at* a stored pattern with a few wrong bits — or far from it, at overlap m — and ask what one sweep of updates does. For a state with overlap m with pattern ξ^μ and no other special correlations, the same central-limit argument as before gives an aligned field $m + \mathcal{N}(0, \alpha)$ on every site, so the overlap after one parallel sweep is the probability of ending aligned minus the probability of ending misaligned:

$$m_1 \;=\; \operatorname{erf}\!\Big(\frac{m_0}{\sqrt{2\alpha}}\Big). \tag{4.4}$$

first-sweep retrieval map — legitimate once

This formula is valid *for the first sweep only* — a restriction that is about to matter. As a first-sweep statement it is already useful: it reproduces Equation 4.2 at $m_0 = 1$ (since $m_1 = 1 - 2P_{\text{err}}$), and at small α it becomes a step function of m_0, recovering the one-pattern result of Section 3.5 that any positive initial overlap retrieves in a single sweep.

The natural next step is to *iterate* the map, treating each sweep as a fresh draw of Gaussian noise. Figure 4.2 cobwebs the map. The retrieval fixed point $m^* = \operatorname{erf}(m^*/\sqrt{2\alpha})$ sits near 1 at small load, slides gracefully downward as α grows, and survives — check the slope at the origin — all the way to $\alpha = 2/\pi \approx 0.64$, where it vanishes *continuously*. The naive theory thus predicts that memory fades, with the overlap degrading smoothly and useful recall extending out to $\alpha \approx 0.64$.

Both the number and the *kind* are wrong. The truth is $\alpha_c \approx 0.138$ — five times smaller — and the failure is discontinuous. The calculation was clean and yet wrong by a factor of five: what did we throw away?

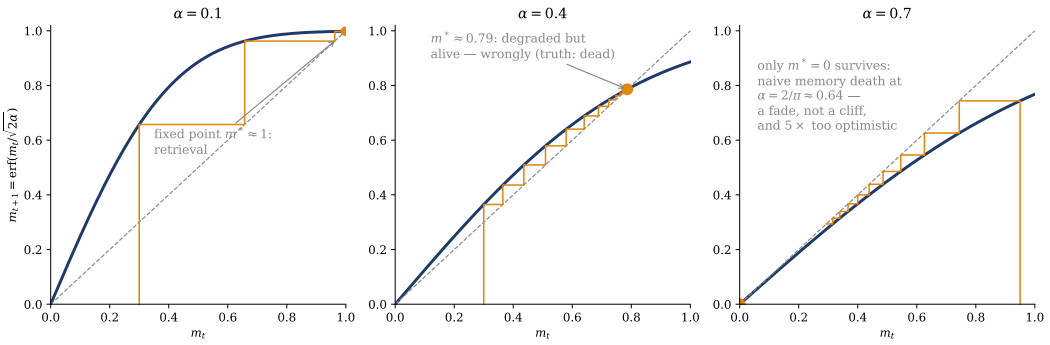

Figure 4.2: The naive theory and its instructive failure. The first-sweep retrieval map $m_{t+1} = \mathrm{erf}(m_t/\sqrt{2\alpha})$ of Equation 4.4, cobwebbed at three loads. Left, $\alpha = 0.1$: rapid convergence to a retrieval fixed point near $m^* = 1$. Center, $\alpha = 0.4$: the naive fixed point survives at $m^* \approx 0.79$ — although the true network is long dead here. Right, $\alpha = 0.7$: past $\alpha = 2/\pi \approx 0.64$ only $m^* = 0$ remains, and the naive theory predicts memory *fading* continuously to nothing. Both the predicted capacity and the predicted kind of failure are wrong, because iterating the map re-randomizes the crosstalk each sweep — precisely what the network does not do.

We threw away the *feedback*. Iterating Equation 4.4 assumes the crosstalk is re-randomized at every sweep — that the noise a spin feels now is statistically independent of the errors made earlier. It is not. A bit that flipped did so *because* the crosstalk pushed it; the flipped bit is therefore correlated with the noise, and through the couplings it pollutes the fields of every other site in a way that does not average out. Errors amplify the very noise that produced them. The correct self-consistent treatment — set up by Amit, Gutfreund, and Sompolinsky (1985) and carried out in full in Amit, Gutfreund, and Sompolinsky (1987) — replaces the bare noise variance α by an *amplified* one, and at zero temperature its equations read (stated here, derived with Module 2's tools):

$$m = \mathrm{erf}\Big(\frac{m}{\sqrt{2\alpha r}}\Big), \qquad \chi = \sqrt{\frac{2}{\pi \alpha r}}\, e^{-m^2/2\alpha r}, \qquad r = \frac{1}{(1-\chi)^2}, \tag{4.5}$$

AGS self-consistency at $T = 0$ (stated)

where χ measures the network's integrated response to its own errors — a susceptibility, distinct from the crosstalk C_i above — and $r \geq 1$ is the factor by which that response inflates the crosstalk. Solving the three equations numerically — an afternoon's exercise once you trust them, and Figure 4.3 is exactly that computation — produces the picture the naive map could not. As α grows, the retrieval solution (overlap near one) is approached from below by an unstable partner solution; the two branches bend toward each other, meet, and *annihilate* at

$$\boxed{\alpha_c \;\approx\; 0.138, \qquad m(\alpha_c) \approx 0.97.} \tag{4.6}$$

storage capacity (Amit–Gutfreund–Sompolinsky)

This is a *saddle-node bifurcation*: beyond α_c the retrieval state does not degrade — it *ceases to exist*, and the dynamics has nowhere to fall but the spin-glass phase at $m = 0$. The overlap jumps discontinuously from 0.97 to 0: memory death is a **first-order transition**. And the number 0.97 is itself the transition's signature: even at the edge, retrieval is still 98.4% correct per bit. Memory near capacity remains nearly perfect — and then, one pattern later, it is absent. (For the physics-native:

a discontinuous order parameter with no symmetry broken between the phases — the transition is about a metastable branch existing or not, spinodal-flavored rather than Ising-flavored. Module 2 gives this remark its proper home.)

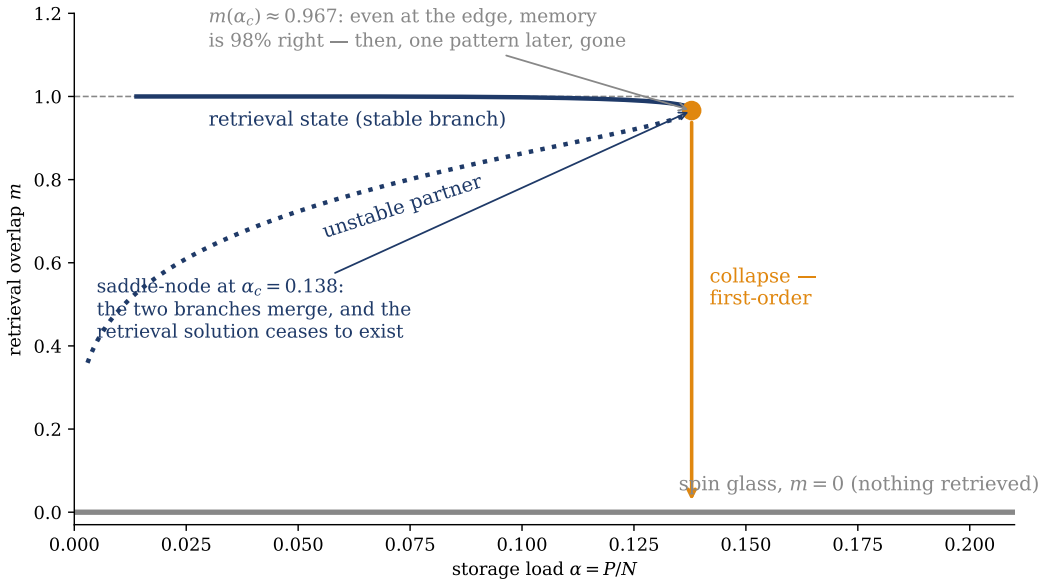

Figure 4.3: How memory actually dies: the retrieval overlap from the self-consistent equations Equation 4.5. The stable retrieval branch (solid) descends only slightly — from $m = 1$ to $m \approx 0.967$ — while its unstable partner (dotted) rises to meet it; at $\alpha_c \approx 0.138$ the two annihilate in a saddle-node bifurcation and no retrieval solution exists beyond. The overlap drops discontinuously to the spin-glass value $m = 0$: a first-order transition. Compare Figure 4.2 — feedback moved the edge from 0.64 to 0.138 *and* changed a fade into a collapse.

Two refinements are worth noting. First, dynamics and equilibrium do not agree perfectly here: tracking the *transient* — overlap and noise variance sweep by sweep, with the long-term correlations handled carefully — is the statistical neurodynamics of Amari and Maginu (1988), which yields a discontinuous transition at $\alpha \approx 0.16$; the residual gap to 0.138 is exactly the error correlations that only the full equilibrium treatment resolves. Second, finite size rounds the cliff: at $N \sim 10^3$ the collapse is a steep sigmoid spread over a window of a few times 10^{-2} in α, drifting toward 0.138 from above as N grows — which is why Hopfield (1982) saw trouble "near $0.15\,N$" at $N = 100$, and why a rounded curve in a simulation does not falsify a sharp theory.

Trap

First-sweep estimates never locate a transition — feedback does. Equation 4.4 is legitimate once and illegitimate iterated: each reuse silently assumes fresh noise, i.e. no correlation between the errors and the crosstalk that caused them. That assumption is not a small numerical sin — it changes the *order* of the transition and misses the capacity by 5×. When an iterated mean-field estimate fails this badly, the discarded correlations *are* the physics. Know which of your equations are laws and which are single-use.

Take-home 2

Retrieval errors feed back: flipped bits amplify the crosstalk that flipped them, inflating the noise variance from α to αr with $r = (1-\chi)^{-2}$. The self-consistent retrieval state disappears in a saddle-node at $\alpha_c \approx 0.138$ — a discontinuous, first-order collapse from $m \approx 0.97$ to 0, with retrieval still 98% correct per bit the instant before it vanishes.

4.5 Memory is a phase

The equations Equation 4.5 are the zero-temperature edge of something larger. Heat the network — we will see in Week 3 that the natural stochastic dynamics at temperature T exists and is well defined — and the same self-consistent analysis runs at every (α, T). The result is Figure 4.4: a genuine *phase diagram* for a machine-learning system, computed by Amit, Gutfreund, and Sompolinsky (1985, 1987) with the replica method of spin-glass theory, in the direct lineage of Sherrington and Kirkpatrick (1975). There are three phases. At high temperature, the **paramagnet** has noise overwhelming any structure, and no memory survives. Below the line $T_g = 1 + \sqrt{\alpha}$, a **spin-glass** phase has exponentially many minima, none of them correlated with anything stored — a rugged landscape of pure crosstalk. And in the region under the curve $T_M(\alpha)$, the **retrieval phase** has stored patterns as attractors, and the network is a working memory. The retrieval boundary runs from $T = 1$ at vanishing load down to $\alpha_c \approx 0.138$ at $T = 0$ — the capacity $\alpha_c \approx 0.138$ is the point where the memory phase pinches off against the temperature axis.

The diagram has two natural paths. The *horizontal* one — loading a cold network with ever more patterns — exits the retrieval phase at α_c; the tutorial follows it. The *vertical* one — heating a lightly loaded network — shows retrieval surviving to high temperature before melting near $T \approx 1$; Week 3 follows this path on purpose. The phase diagram also settles a point from the last lecture. Within the retrieval region, a little noise is *useful*: the spurious attractors — the mixture states of Chapter 3 — are shallower than the retrieval states and melt at lower temperature, so a warm network loses its spurious states while keeping its memories. The same temperature that restores the Boltzmann distribution also cleans the landscape.

Now we state the debt precisely. What the replica method buys — and what our tools could not — is the average of $\log Z$ over the *stored patterns themselves*: the couplings J_{ij} are random variables built from the ξ^μ, the free energy must be averaged over that *quenched disorder*, and for an extensive number of patterns ($P = \alpha N$) this average is a nontrivial computation. The technology that computes it — the replica trick, and order parameters beyond m (the Edwards–Anderson overlap q, and the noise amplification r you have already met) — is spin-glass theory proper. Module 2 acquires its modern descendants — mean-field theory, its systematic corrections, and the cavity ideas that became inference algorithms — and from that vantage the *logic* of these equations (feedback, reaction, self-consistency) will be yours to command; the replica computation itself remains, in this course, a named landmark rather than a technique. So Equation 4.5 and Figure 4.4 stand as *quoted* results, with the physics behind them developed in Module 2's own language.

One information-theoretic aside, because the number is instructive. At capacity, the network holds $P \cdot N = 0.138\, N^2$ pattern bits in N^2 real-valued couplings — about

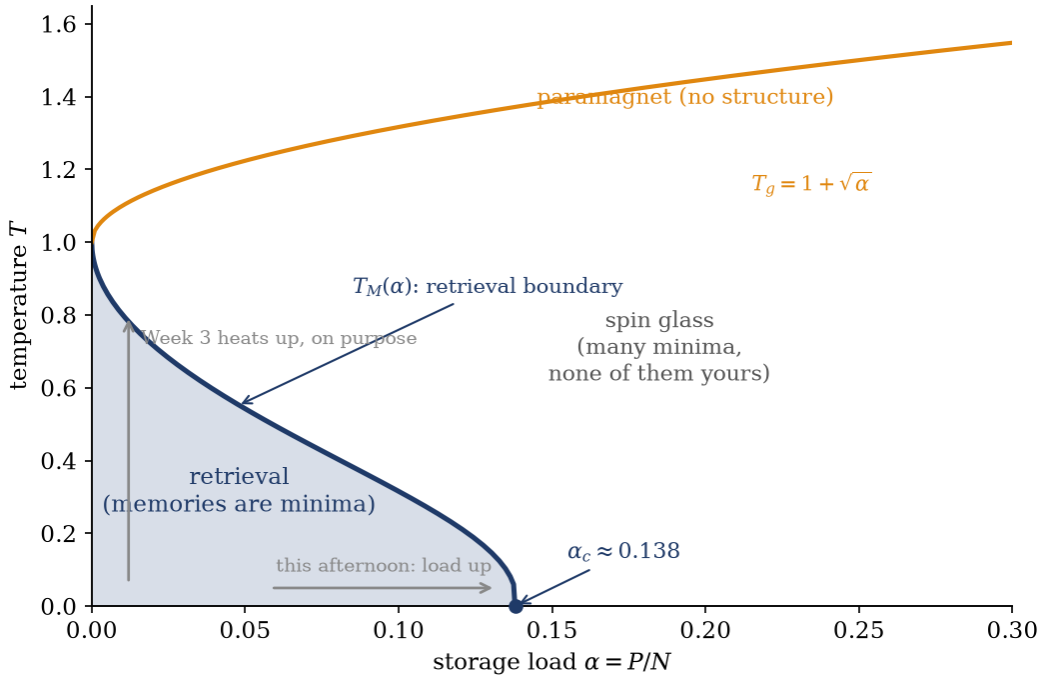

Figure 4.4: The phase diagram of the Hopfield model with Hebbian couplings, computed from the finite-temperature replica-symmetric theory of Amit, Gutfreund, and Sompolinsky (1985, 1987) (the boundary $T_M(\alpha)$ is solved numerically from the self-consistent equations; $T_g = 1 + \sqrt{\alpha}$). Below $T_M(\alpha)$, stored patterns are attractors and the network is a working memory; the retrieval phase ends at $\alpha_c \approx 0.138$ on the cold axis. Between T_M and T_g, only spin-glass states survive — many minima, none of them yours. The two gray arrows are the course's itinerary: the tutorial walks the horizontal path across α_c; Week 3 walks the vertical one, on purpose.

0.14 bits per synapse. That is *cheap*. The optimal capacity of a network permitted to choose its couplings freely — not restricted to Hebb's one-shot rule — is 2 bits per synapse, a bound computed by Elizabeth Gardner in another replica landmark (Gardner 1988) — one this course cites rather than derives. The Hebb rule trades a factor of fifteen in storage efficiency for the privilege of learning in a single pass, with no optimization at all.

> Take-home 3
>
> The Hopfield model has a phase diagram: retrieval, spin glass, and paramagnet in the (α, T) plane. Memory is a *phase of matter*; capacity is its boundary at $T = 0$. Within the phase, mild noise cleans the landscape — spurious states melt before memories do. The full computation runs on replica machinery (quenched average of $\log Z$) that this course quotes rather than builds; Module 2 acquires its modern mean-field descendants.

4.6 Example: the capacity ledger

One concrete network — $N = 1000$ neurons, the network of Section 3.6 — and every capacity claim of the lecture evaluated on it.

question asked	formula	$P_{\max}$	reading
every pattern letter-perfect	$N/(4 \ln N)$	≈ 36	perfection, all at once
a typical pattern letter-perfect	$N/(2 \ln N)$	≈ 72	perfection, one at a time
retrieval with $\sim$2% tolerated error	$\alpha_c N$, $\alpha_c \approx 0.138$	≈ 138	the AGS answer
Hopfield's simulation criterion	$\approx 0.15\, N$ (empirical, $N = 100$)	≈ 150	finite-size, tolerant
naive iterated map	$(2/\pi)\, N$	≈ 637	wrong value, wrong kind

Five numbers for a single network answer three genuinely different questions, with the wrong answer retained as a record of what the naive theory misses. The opening question is settled: Hopfield's 0.15 was the right physics, seen at $N = 100$ through finite-size rounding, *for the tolerant question*; sharpened, it is 0.138. The strict question was never the one his simulations asked, and its answer is smaller by a logarithm.

The second table makes the same point numerically. For each load: the first-sweep expectation — how many of the 1000 bits flip on the first sweep, starting *at* a stored pattern, from Equation 4.2 — against the true fate of the memory, from Figure 4.3:

$\alpha = P/N$	P	P_{err}	first-sweep flips (of 1000)	true fate
0.05	50	4×10^{-6}	~ 0	retrieval, $m \approx 1$
0.10	100	8×10^{-4}	~ 1	retrieval, m high
0.138	138	3.6×10^{-3}	~ 4	**the edge**: $m \approx 0.97$
0.15	150	4.9×10^{-3}	~ 5	**dead**: $m \approx 0$
0.20	200	1.3×10^{-2}	~ 13	dead

The $\alpha = 0.15$ row makes the point. The first sweep gets 99.5% of the bits right — and the memory is already gone. Between 138 and 150 patterns, nothing in the per-bit error rate changes appreciably; what changes is that the self-consistent retrieval state stops existing, and the almost-correct first sweep is the first step of an avalanche instead of the last step of a repair. The erfc column is smooth through the transition because the transition was never in it: it lives in the feedback.

> *Memory does not fade — it shatters: capacity is a phase boundary, not a budget.*

The tutorial loads a network at $N \sim 10^3$ through this exact table; every number required for predicting the results is already in the tables above. The hard problem builds on the warm-up stuck-point card. The week is computation-led, which means the prediction is committed on paper first and each group then runs the sweep itself — one laptop between three or four people, so bring one if you have one.

4.7 Outlook: the partition function returns

Week 2 closes with a substantial inventory. We built a machine out of an Ising model, proved it computes (the Lyapunov theorem), taught it memories in one shot (Hebb), computed what interference costs (today), and found that the machine has a phase diagram in which *memory is a phase and forgetting is a phase transition.* All of it at zero temperature, with the partition function absent — a stretch of the course where the energy landscape was used bare, with no probability on top.

The step to Week 3 is a single modification. Replace the deterministic update of Equation 3.1 by a stochastic one — flip spin k with probability

$$p(s_k \to -s_k) \;=\; \frac{1}{1 + e^{\beta\,\Delta E}},$$

Glauber dynamics (Week 3)

which needs exactly the $\Delta E = 2 s_k h_k$ we computed in the Lyapunov derivation, and which reduces to today's greedy rule as $\beta \to \infty$. Two statements, both proved in the previous lecture. First, this dynamics has a stationary distribution, and it is $p(s) \propto e^{-\beta E(s)}$: *Week 1's Boltzmann law returns, now over an energy we designed* — the two parts of the course connect. Second, the moment we ask the network to *learn* its couplings from data by maximum likelihood — rather than have Hebb write them by fiat — the gradient comes out as $\langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}$, and the second average requires sampling from a distribution normalized by Z. That machine is the Boltzmann machine, that gradient is the most transparent learning rule in this

course, and that Z is the obstruction of Week 1, now blocking the training of every energy-based model. Week 3 is where these threads converge.

5 The Boltzmann machine: learning meets the partition function

Recommended reading

Core (~1 h). Ackley, Hinton, and Sejnowski (1985) — the founding paper, and the source of this lecture's vocabulary: the stochastic network, the two-phase learning rule (their "positive" and "negative" phases), and the honest admission that the model expectation must be estimated by sampling. Read it for the learning rule and the phase language; the partition-function obstruction is written between its lines, and today we make it explicit. Alongside it, Mehta et al. (2019), §VII — the physicist's telling, in the notation this course uses; the moment-matching reading of the learning rule is cleanest there.
Optional background. MacKay (2003), Ch. 43, for the maximum-likelihood gradient done carefully from the inference side; Glauber (1963) for the single-spin-flip stochastic dynamics whose stationary law is Boltzmann — the $T > 0$ dynamics teased at the end of Chapter 4, sourced (skim for the flip rule and detailed balance). Hinton (2002) is deliberately *not* on this list: it is the tutorial's warm-up reading, and we will not spoil its workaround here.
Prerequisite reminder. Everything from Week 1 — the Boltzmann distribution and Z (Section 1.4), and the generating-function machinery of Chapter 2, where derivatives of $\log Z$ produce expectations — and everything from Week 2: the network's energy, the local field, the $\langle\cdot\rangle$ bookkeeping. New today: stochastic dynamics and its stationary law; and, the heart of the lecture, couplings that are *learned* from data rather than written by Hebb.

5.1 Week 2's promise, redeemed

Week 2 ended on a promissory note: replace the deterministic update Equation 3.1 by a coin flip — flip spin k with probability $1/(1+e^{\beta\Delta E})$ — and two things follow. First, the stationary distribution of this dynamics is exactly Week 1's Boltzmann law over the network's energy: the two halves of the course so far click together. Second, the moment we ask the network to *learn* its couplings from data by maximum likelihood — rather than have Hebb write them by fiat — the gradient acquires a term that requires Z. Both are proved below.

The history hook is the other half of the 2024 Nobel: Week 2 opened on the Hopfield half; today is the **Hinton half**. Ackley, Hinton, and Sejnowski (1985) took Hopfield's Ising network, made it stochastic, gave it a learning algorithm, and named the result after the distribution at its heart: the *Boltzmann machine*. The lineage traced in Chapter 1 thus closes its longest loop — Boltzmann's counting in the 1870s, Gibbs's partition function in 1902, Hopfield's landscape in 1982, and now, 1985, a machine that carries Boltzmann's name and learns a probability distribution from examples.

The citation honors this *as physics*: the machine is an Ising model, its dynamics is Glauber's, and its failure mode is a partition function.

The question is one step past Week 2's. That network remembered what *we* wrote into it — Hebb's rule Equation 3.4 is a storage prescription, not a learning algorithm. *Can a network learn what to remember — discover its own couplings from data alone?* Yes, by a short rule — a few lines of calculus. The price is the obstruction the course is organized around: Z, named in Chapter 1 as the training obstruction of Module 1, is here derived and pinned to a precise mathematical object.

Module 1's arc is **measure E (Week 1) → design E (Week 2) → learn E (Week 3)**, and this lecture completes it. A deeper symmetry runs through the argument: Week 1 *derived* the Boltzmann distribution from maximum entropy with the energy given (Section 1.4); here we run that derivation backwards — the data supply the constraints, and out fall both the learning rule and its obstruction. The course's opening derivation and its first learning algorithm turn out to be one theorem seen from two sides.

> Take-home 1
>
> A Boltzmann machine is Week 2's network heated up (Glauber dynamics ⇒ Boltzmann stationary law) and made *trainable* (couplings fit to data by maximum likelihood). Memorizing — Hebb, one shot, no objective — becomes learning: gradient ascent on the likelihood. The step from *design* to *learn* is the step that brings in Z.

5.2 Memorize versus learn

Two dichotomies organize what follows.

The first is **memorize versus learn**, drawn in Figure 5.1. Hebb's rule Equation 3.4 writes the couplings in a single pass over the patterns: no objective function, no gradient, no iteration — the minima of the landscape are *placed*, once, at prescribed locations. Maximum likelihood inverts the logic. We posit a *model distribution* $p(s) = e^{-\beta E(s)}/Z$ whose energy carries free parameters (J, θ), and we *choose* those parameters to maximize the probability the model assigns to the training data. The couplings are no longer written; they are *fit*, and the landscape is molded — iteratively, under an objective — until its low-energy regions sit under the data. Hebb, in this light, is a special, cheap, often-suboptimal solution to a problem maximum likelihood poses properly; we will meet Hebb again *inside* the exact learning rule, as one of its two terms.

The second dichotomy is the one the rest of the course keeps returning to: **positive phase versus negative phase**. Every gradient we write today is a difference of two brackets — an average over the *data* ("what the world shows you") and an average over the *model* ("what the network dreams up"). Hinton's informal names, the *wake* phase and the *sleep* phase, are a serviceable mnemonic (not to be confused with the later wake–sleep *algorithm* for Helmholtz machines, Dayan et al. (1995), a different construction); positive and negative phase, the terms of Ackley, Hinton, and Sejnowski (1985), are the technical vocabulary we keep. One of the two brackets is trivial to compute; the other is the wall — and telling them apart is the argument below.

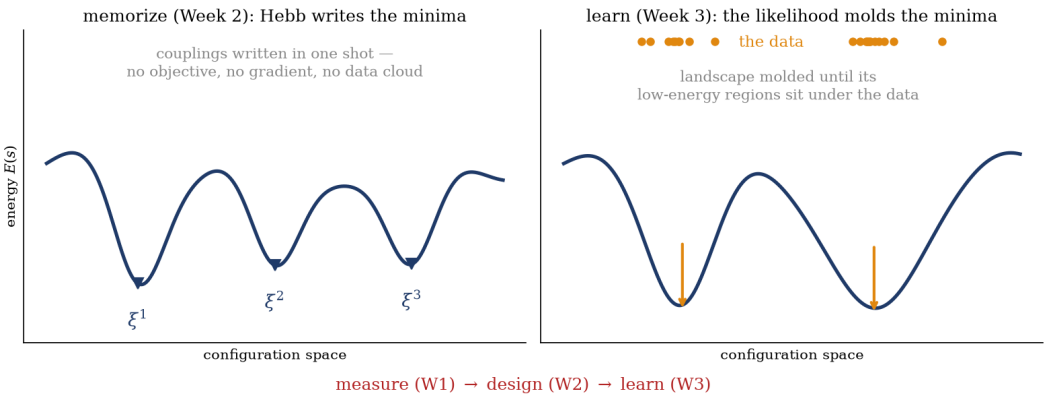

Figure 5.1: The module's last turn: from couplings written to couplings fit. Left: Hebb (Week 2) places the minima at prescribed patterns ξ^μ in one shot — no objective, no data cloud, no iteration. Right: maximum likelihood (this week) posits an energy with free parameters and molds the landscape under an objective until its basins sit beneath the data. The objective is the log-likelihood, and its gradient is the subject of this lecture — the two forces of Figure 5.2.

Figure 5.2 draws the central picture; every result below is a refinement of it — a single energy landscape over configuration space. At each **data point**, an arrow pulling the energy *down*: make what you saw more probable. At each **model sample** — each fantasy the network dreams on its own — an arrow pushing the energy *up*: make what you merely imagined less probable. Learning halts when the two sets of arrows balance, which happens precisely when the model's fantasies are statistically indistinguishable from the data. Training is *down on data, up on dreams.*

Three results make this precise: ① the model distribution, and why it is Boltzmann (Section 5.3); ② the exact form of the two forces — the maximum-likelihood gradient (Section 5.4); ③ why the "up on dreams" force is the expensive one (Section 5.5).

5.3 Heating the network up: Glauber dynamics

The dynamics comes first. Week 2's network ran at zero temperature — the update Equation 3.1 was the locally greedy choice, and the Lyapunov theorem of Section 3.4 rewarded it with deterministic descent. Now heat it up, following Glauber (1963). The energy is Week 2's, with one addition: the thresholds we set to zero there return as learnable *biases* (physics dialect: local fields),

$$E(s) \;=\; -\tfrac{1}{2}\sum_{i\neq j} J_{ij}\, s_i s_j \;-\; \sum_i \theta_i\, s_i, \qquad J_{ij} = J_{ji}, \quad J_{ii} = 0. \tag{5.1}$$

Boltzmann-machine energy

A notation flag before we move: θ_i is a *bias* — one learnable parameter of the energy — and when we later write ∇_θ for "the gradient with respect to all parameters," it ranges over the full collection (J, θ). As Chapter 1 warned, symbols will be overloaded; this is the first collision that matters.

The stochastic update. Pick a site k at random — asynchronously, as in Week 2 — and *propose* the flip $s_k \to -s_k$. The flip costs $\Delta E_k = 2\, s_k h_k$, where $h_k =$

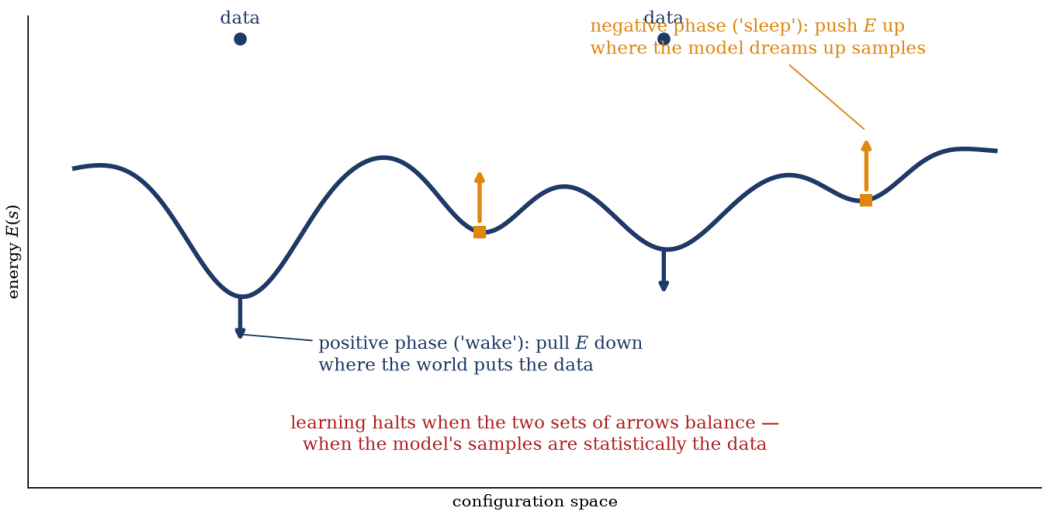

Figure 5.2: The central picture of the lecture, drawn before the mathematics. Training a Boltzmann machine exerts two forces on the energy landscape: the positive ("wake") phase pulls E down at the data, making the observed configurations more probable; the negative ("sleep") phase pushes E up at the model's own samples, making its fantasies less probable. The gradient we derive below is exactly this pair of forces, and learning stops when they balance — when sampling the model is statistically like sampling the data. The negative arrows are the expensive ones: finding where the model dreams requires Z.

$\sum_j J_{kj} s_j + \theta_k$ is the local field (the same ΔE that ran the Lyapunov argument). Accept the flip with probability

$$w(s_k \to -s_k) \;=\; \frac{1}{1 + e^{\beta \Delta E_k}} \tag{5.2}$$

Glauber flip rule

— downhill moves are accepted gladly ($w \to 1$ for $\Delta E \ll -T$), uphill moves reluctantly but *not never* ($w \to 0$ only as $\Delta E/T \to \infty$), and as $\beta \to \infty$ the rule collapses onto Week 2's greedy $\operatorname{sign}(h_k)$. Equivalently, the rule sets

$$p(s_k = +1 \mid s_{\setminus k}) \;=\; \frac{1}{1 + e^{-2\beta h_k}} \;=\; \sigma(2\beta h_k)$$

— the logistic function, which you last met as the two-state Boltzmann occupation in Section 1.6. The "activation function" of every neural network is, here, literally a conditional Boltzmann probability for a two-state degree of freedom in the field of its neighbors.

Detailed balance, in three lines. Where does this dynamics settle? Consider two configurations s and s' that differ at the single site k, and take the ratio of forward and backward flip probabilities. With $\Delta E = E(s') - E(s)$, the reverse move costs $-\Delta E$, so

$$\frac{w(s \to s')}{w(s' \to s)} \;=\; \frac{1 + e^{-\beta \Delta E}}{1 + e^{\beta \Delta E}} \;=\; e^{-\beta \Delta E}\, \frac{e^{\beta \Delta E} + 1}{1 + e^{\beta \Delta E}} \;=\; e^{-\beta [E(s') - E(s)]}.$$

The transition probabilities therefore satisfy *detailed balance*, $p(s)\,w(s \to s') = p(s')\,w(s' \to s)$, with respect to exactly one distribution:

$$p(s) \;=\; \frac{e^{-\beta E(s)}}{Z}, \qquad Z = \sum_{s} e^{-\beta E(s)} \tag{5.3}$$

the model distribution

— and a distribution satisfying detailed balance is stationary: the probability flowing out of any state along each single-flip edge is exactly balanced by the flow back in. In words: *Week 1's Boltzmann distribution has returned — but where Week 1 handed us the energy, here E carries free parameters we are about to fit, and Z is the normalizer we will spend the rest of the course working around.* Heating the network up is what turns a memory into a probability distribution over configurations: the machine no longer sits in one minimum; it *visits* configurations with Boltzmann weights, and can be asked what it thinks is likely.

Two caveats, one sentence each. Stationarity plus *irreducibility* (any state reachable from any state — true here, since every $w > 0$ at finite T) and aperiodicity guarantee that the chain *converges* to Equation 5.3 from any start; asynchronous random-site updating supplies both, and we take the ergodic bookkeeping as read. How *fast* it converges — the mixing time — is an entirely separate matter, and we flag now that it returns as the second obstruction in Section 5.5. Finally, the temperature: β multiplies parameters we are about to learn, so it can be absorbed into them — a machine at $\beta = 2$ with couplings J is the same model as a machine at $\beta = 1$ with couplings $2J$. **We set $\beta = 1$ from here on** and say so once.

Figure 5.3 shows the theorem at work on a machine small enough to check exhaustively — three spins, the same frustrated triangle we dissect by hand in Section 5.7: the empirical visit frequencies of the Glauber chain converge onto the eight exact Boltzmann probabilities, and a running average over the chain converges onto the exact model expectation $\langle s_1 s_2 \rangle_{\text{model}}$. The right panel matters: that wiggly line — an expectation estimated by running the dynamics — is precisely the object the learning rule of Section 5.4 demands, once per gradient step.

5.4 The engine: the maximum-likelihood gradient

To fit (J, θ) we need an objective and its gradient. The objective writes itself; but $\log p$ contains $-\log Z$, and Z depends on every parameter, so the gradient will contain $\partial \log Z / \partial (J, \theta)$ — and that term will not go away. We meet it by computing it.

Setup. The data are D configurations $\mathcal{D} = \{s^{(1)}, \dots, s^{(D)}\}$ — binarized images, spike patterns, spin snapshots — treated as independent draws. The (average) *log-likelihood* is

$$L(J, \theta) \;=\; \frac{1}{D} \sum_{d=1}^{D} \log p(s^{(d)}) \;=\; \big\langle \log p(s) \big\rangle_{\text{data}}, \qquad \log p(s) = -E(s) - \log Z,$$

with $\beta = 1$ absorbed as agreed. We climb L by gradient ascent, so we need its derivatives with respect to each coupling J_{ij} and each bias θ_i.

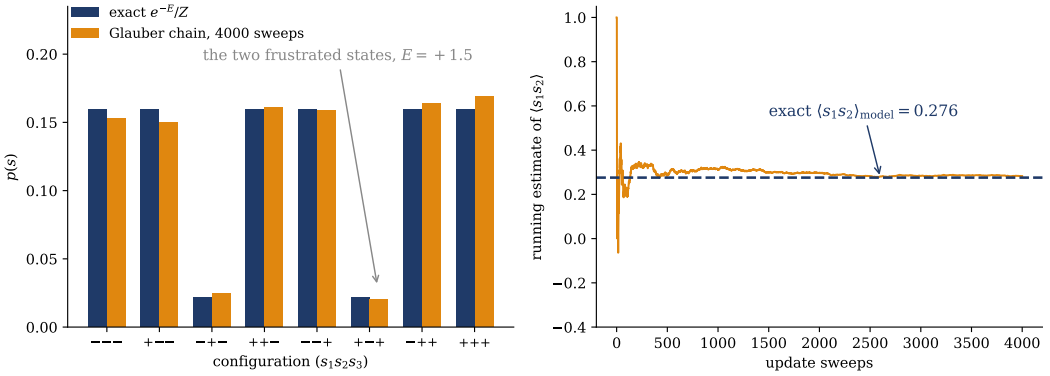

Figure 5.3: Glauber dynamics finding the Boltzmann distribution, on the $N = 3$ machine of Section 5.7 ($J_{12} = J_{23} = 0.5$, $J_{13} = -0.5$, $\theta = 0$; $Z = 10.34$). Left: exact Boltzmann probabilities $e^{-E(s)}/Z$ of the eight configurations (navy) against the visit frequencies of a single Glauber chain of 4000 sweeps (orange) — the six degenerate low-energy states at $E = -0.5$ and the two frustrated states at $E = +1.5$ are all reproduced. Right: the running chain estimate of $\langle s_1 s_2 \rangle$ converging onto the exact model expectation 0.276 (dashed). This wiggly line is the shape of things to come: the learning rule of Section 5.4 demands exactly such a model expectation, re-estimated at every gradient step.

The coupling gradient, term by term. Differentiating the two pieces of $\log p$ separately,

$$\frac{\partial L}{\partial J_{ij}} = \left\langle -\frac{\partial E}{\partial J_{ij}} \right\rangle_{\text{data}} - \frac{\partial \log Z}{\partial J_{ij}}.$$

The first term is bookkeeping. In the energy Equation 5.1, the symmetric pair (J_{ij}, J_{ji}) is one parameter that appears twice in the double sum, and the $\frac{1}{2}$ exists precisely to cancel that double count:

$$-\frac{\partial E}{\partial J_{ij}} = s_i s_j \qquad \Longrightarrow \qquad \left\langle -\frac{\partial E}{\partial J_{ij}} \right\rangle_{\text{data}} = \langle s_i s_j \rangle_{\text{data}}$$

— a plain correlation, read off the training set in a single pass. The second term is the move you already know from Chapter 2, where differentiating $\log Z$ with respect to β produced $\langle E \rangle$: a derivative of $\log Z$ is always an expectation. Here the derivative is with respect to a coupling instead of the temperature, and the same three steps give

$$\frac{\partial \log Z}{\partial J_{ij}} = \frac{1}{Z} \sum_s \frac{\partial}{\partial J_{ij}} e^{-E(s)} = \frac{1}{Z} \sum_s s_i s_j \, e^{-E(s)} = \langle s_i s_j \rangle_{\text{model}}$$

— the same correlation, evaluated not over the data but over *the model's own distribution* Equation 5.3. Assembling, and running the identical two-liner for the biases ($-\partial E/\partial \theta_i = s_i$, and $\partial \log Z/\partial \theta_i = \langle s_i \rangle_{\text{model}}$):

$$\boxed{\frac{\partial L}{\partial J_{ij}} = \langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}, \qquad \frac{\partial L}{\partial \theta_i} = \langle s_i \rangle_{\text{data}} - \langle s_i \rangle_{\text{model}}.} \tag{5.4}$$

Boltzmann-machine learning rule

This is the learning rule of Ackley, Hinton, and Sejnowski (1985), and we read it term by term. Gradient ascent, $\Delta J_{ij} = \eta\,(\langle s_i s_j\rangle_{\text{data}} - \langle s_i s_j\rangle_{\text{model}})$, says: *raise* the coupling between two units that co-fire in the data more often than the model expects; *lower* it where the model over-predicts co-firing. The first term is Hebb — "fire together, wire together," now averaged over data rather than written from patterns. The second is *anti*-Hebbian: unlearning applied to the network's own fantasies. Hebb's rule of Week 2 is thereby exposed as the positive phase running without its counterweight — which is exactly why it over-writes and interferes (Section 4.2), and why it tops out at $\alpha_c \approx 0.138$ while optimally chosen couplings reach Gardner's $\alpha = 2$ (Gardner (1988)). The two terms are the two arrow sets of Figure 5.2, no longer a cartoon: down on data, up on dreams, and learning stops when they cancel.

Differentiating Equation 5.4 once more gives the Hessian $\partial^2 L/\partial J_{ij}\,\partial J_{kl} = -(\langle s_i s_j s_k s_l\rangle_{\text{model}} - \langle s_i s_j\rangle_{\text{model}}\langle s_k s_l\rangle_{\text{model}})$ — minus a covariance matrix, which is negative semidefinite. The log-likelihood of a fully visible Boltzmann machine is therefore *concave* in (J, θ): there is one optimum and no bad local maxima. This point matters below: it says the problem we are about to declare intractable is intractable for one reason only, the cost of the negative phase, and not *also* because of a rugged objective. (Hidden units, in Section 5.6, break the concavity; the clean case is the fully visible one.)

5.5 The negative phase is the partition function

The *negative phase*, written out in full:

$$\boxed{\langle s_i s_j\rangle_{\text{model}} \;=\; \frac{1}{Z}\sum_{s\in\{-1,+1\}^N} s_i s_j\, e^{-E(s)} \;=\; \frac{\partial \log Z}{\partial J_{ij}}} \tag{5.5}$$

the training obstruction, derived

Both the sum and its normalizer run over all 2^N configurations of the network. This is precisely the object Chapter 1 named on day one — *"the Boltzmann machine cannot be trained because $\nabla_\theta \log Z$ is intractable"* — no longer a promise but a derived fact: the gradient of the log-likelihood *is*, in its second term, the parameter-gradient of $\log Z$.

Set the two phases side by side and the asymmetry is stark. The **positive phase** $\langle s_i s_j\rangle_{\text{data}}$ is a sum over D training points — linear in the dataset, one pass, trivial. The **negative phase** is a sum over 2^N model configurations, gated by Z — exponential in the system. And there is no clever reparameterization that removes it: the term exists *because the model is normalized*. Any p that integrates to one must pay $-\log Z$ in its log-likelihood, and any parameter that shapes the landscape moves Z. Learning forces the model expectation into the gradient; this asymmetry — cheap data averages, exponential model averages — recurs in every training method from here on.

> Trap
>
> **The negative phase is not a statistic of the data.** $\langle s_i s_j\rangle_{\text{model}}$ is an expectation over the model's *own* distribution — which shifts at every gradient step, because the parameters just moved. It must be re-estimated, from scratch

or by clever warm-starting, at *every* update. The student who computes it once and reuses it, or who quietly conflates it with a data correlation, has not made a small numerical error; they have deleted the "sleep" force entirely, and their model will happily assign ever-lower energy to everything. This is the most common misunderstanding of energy-based learning.

Maximum entropy, run backwards. At the optimum the gradient Equation 5.4 vanishes, so

$$\langle s_i s_j \rangle_{\text{model}} = \langle s_i s_j \rangle_{\text{data}} \quad \text{and} \quad \langle s_i \rangle_{\text{model}} = \langle s_i \rangle_{\text{data}} \quad \text{for all } i, j. \tag{5.6}$$

moment matching at the optimum

The trained machine reproduces the data's first and second moments *exactly* — and it does so with a distribution of the Boltzmann form $p \propto \exp\big(\sum_i \theta_i s_i + \tfrac{1}{2}\sum_{i\neq j} J_{ij} s_i s_j\big)$, which is exactly the functional form that maximum entropy produces when the constrained observables are $\{s_i\}$ and $\{s_i s_j\}$. The parallel with Section 1.4 is exact: there, we maximized entropy subject to a known mean energy, and β emerged as the Lagrange multiplier enforcing the constraint. Here, the constraints are the data's moments, and the multipliers enforcing them are precisely the biases and the couplings. *Training a fully visible Boltzmann machine is Jaynes's derivation run backwards* (Jaynes 1957): among all distributions matching the data's first two moments, gradient ascent on the likelihood finds the one of maximum entropy, and (θ, J) are its Lagrange multipliers. The course's opening derivation and its first learning algorithm are one theorem seen from two sides, and Module 1's loop (measure → design → learn) closes.

The duality, made explicit. The correspondence is exact, not analogical, and the derivation explains structurally why the learning rule has the form it has (Welling, Lu, and Holdijk 2026, ch. 1 on convex duality). Pose Jaynes's problem with the data's moments as the constraints: among all distributions $p(s)$, maximize the entropy $S[p] = -\sum_s p \log p$ subject to $\langle s_i \rangle_p = \langle s_i \rangle_{\text{data}}$, $\langle s_i s_j \rangle_p = \langle s_i s_j \rangle_{\text{data}}$, and normalization. Attach one multiplier per constraint — θ_i to each mean, J_{ij} to each correlation (the names are not an accident), λ to the normalization — and write the Lagrangian

$$\mathcal{L}[p;\, \theta, J, \lambda] = S[p] + \sum_i \theta_i \big(\langle s_i \rangle_p - \langle s_i \rangle_{\text{data}}\big) + \tfrac{1}{2}\sum_{i\neq j} J_{ij}\big(\langle s_i s_j \rangle_p - \langle s_i s_j \rangle_{\text{data}}\big) - \lambda\Big(\sum_s p(s) - 1\Big).$$

Stationarity in p is one functional derivative, exactly as in Section 1.4: $\delta\mathcal{L}/\delta p(s) = -\log p(s) - 1 + \sum_i \theta_i s_i + \tfrac{1}{2}\sum_{i\neq j} J_{ij} s_i s_j - \lambda = 0$, solved by $p(s) = e^{-E(s)}/Z$ with E the machine's energy Equation 5.1 — the model distribution Equation 5.3, its couplings and biases now revealed as the multipliers. What remains is the multipliers' own optimization. Substitute the stationary p back: $-\log p = E + \log Z$ turns the entropy into $\langle E \rangle_p + \log Z$, the moment terms assemble to $-\langle E \rangle_p$ minus the data sums, and the Lagrangian collapses to the *dual function*

$$\mathcal{L}_{\text{D}}(\theta, J) \;=\; \log Z \;-\; \sum_i \theta_i \langle s_i \rangle_{\text{data}} \;-\; \tfrac{1}{2}\sum_{i\neq j} J_{ij} \langle s_i s_j \rangle_{\text{data}} \;=\; -\,L(J, \theta),$$

the dual of maximum entropy is the negative log-likelihood

which is the negative of the log-likelihood of Section 5.4, term for term ($\langle -E \rangle_{\text{data}}$ is precisely the two moment sums). Read the dictionary in both directions. Maximum entropy is the primal problem; maximum likelihood is its dual, and since the dual of a maximization upper-bounds it, the dual problem *minimizes* $\mathcal{L}_{\text{D}}$ — that is, maximizes the likelihood. Gradient descent on the dual adjusts each multiplier until the constraint it polices is satisfied, $\partial \mathcal{L}_{\text{D}} / \partial \theta_i = \langle s_i \rangle_{\text{model}} - \langle s_i \rangle_{\text{data}}$, and that is the learning rule Equation 5.4, component by component. The moment matching Equation 5.6 thereby stops being a fortunate property of the optimum and becomes what it always was — the primal constraints, enforced by their multipliers. Even the concavity of Section 5.4 finds its structural home: a dual function is convex whatever the model (the covariance Hessian computed there is the general theorem's local face), and because the entropy is concave and the constraints affine in p, strong duality holds — the two problems share one optimum with no gap. Week 1's β, the single multiplier that enforced $\langle E \rangle = E$, was the one-parameter preview of this dictionary; the Boltzmann machine simply runs it with $N + \binom{N}{2}$ constraints at once.

First, the *KL reading*, in one line: since $\langle \log p \rangle_{\text{data}}$ differs from $-D_{\text{KL}}(p_{\text{data}} \| p_{\text{model}})$ only by the data's own entropy, which contains no parameters, maximizing the likelihood is *minimizing the Kullback–Leibler divergence from the data to the model*. This is the first appearance of the variational object Module 2 is built on, and it redeems a promise from Week 1 — the free-energy/KL identity deferred there is developed in full in Module 2.

Second, suppose someone gave you Z for free. You still could not evaluate the negative phase exactly (the 2^N sum remains), so the natural estimator is the one Ackley, Hinton, and Sejnowski (1985) already proposed: *run the Glauber chain of Section 5.3 to equilibrium and average* — the right panel of Figure 5.3, per gradient step. But the model whose landscape you are sampling is an Ising network with learned, heterogeneous couplings — exactly the terrain Week 2 showed can be rugged, with exponentially many metastable minima (Section 4.5). A chain started anywhere must *mix* across energy barriers to visit the distribution fairly, and on a rugged landscape mixing can itself take exponential time. The course now carries **two obstructions**: *normalization* — Z and its parameter-gradient, the through-line since Week 1 — and *sampling*: even evading the sum, you must draw fair samples from a distribution you cannot normalize, on a landscape built to have many basins. Module 2 attacks the first by approximation; Module 3 is devoted to the second; Module 4 dissolves both for generative modeling. Both appear together in Equation 5.5.

Take-home 2

The negative phase *is* $\partial \log Z / \partial J_{ij}$ — a sum over 2^N model configurations, the obstruction Week 1 promised, now derived. It cannot be reparameterized away (it is the price of normalization), must be re-estimated at every step, and even its sampling estimator runs into a second wall: mixing on a rugged landscape. At the optimum, model moments equal data moments — training a Boltzmann machine **is** maximum entropy with the data's moments as constraints, and (θ, J) are the Lagrange multipliers.

5.6 Hidden units: power at the price of a second expectation

One generalization must be stated before the next lecture, and we state it result-first, keeping the wall of Section 5.5 as the lecture's peak. A notation flag first, because a collision is unavoidable: from here, h denotes a *hidden unit*, not the local field h_k of Section 5.3 — the course reserves h_i with a site index under discussion of dynamics for the field, and plain h, h_j in the visible/hidden split for latent units. We flag it aloud in the room and once here.

The fully visible machine has a structural ceiling built into Equation 5.6: it matches first and second moments, *and nothing else* — it is second-order maximum entropy, blind to any higher-order structure in the data (three-way correlations, parity constraints, the composition rules of images). The fix is the physicist's oldest move: add degrees of freedom you do not observe, and trace them out. Split the units into **visible** units v — carrying the data — and **hidden** units h, with a joint energy $E(v, h)$ of the same pairwise form Equation 5.1 over the concatenated system, and let the model of the data be the *marginal*

$$p(v) \;=\; \frac{1}{Z} \sum_{h} e^{-E(v,h)}.$$

Marginalizing over h induces *effective interactions of all orders* among the visibles — integrate out a shared hidden neighbor and its visible partners inherit couplings beyond pairwise — so the machine is strictly more expressive at fixed visible count. (The modern reach of this expressivity — energy-based models with deep energies, and the identification of attention with a Hopfield energy — is Week 4's subject.)

The gradient keeps its two-phase shape — for a visible–hidden coupling W_{ij},

$$\frac{\partial L}{\partial W_{ij}} \;=\; \langle v_i h_j \rangle_{\text{clamped}} \;-\; \langle v_i h_j \rangle_{\text{model}},$$

with the derivation identical in structure to Section 5.4 — but read the first bracket carefully, because something just got worse. The hidden units are never observed, so the positive phase is no longer a lookup: it *clamps* the visibles to each data point and averages over the conditional $p(h \mid v)$ — an inference problem inside the learning loop. For a general graph (Figure 5.4, left), that conditional couples all the hidden units to each other, and is itself intractable. Hidden units thus add a *second* hard expectation on top of the Z wall — and, as flagged in Section 5.4, they break the concavity of the objective as well.

Two problems, two escapes — and they are exactly the next lecture's program, so we name them and stop. ① Make the *positive* phase exact by architecture: forbid within-layer couplings, so that given v the hidden units decouple and $p(h \mid v)$ factorizes — the *restricted Boltzmann machine* (Figure 5.4, right). ② Make the *negative* phase cheap by approximation: replace the equilibrated model average with a short Gibbs chain started *at the data* — *contrastive divergence* (Hinton 2002). Why a deliberately unconverged chain gives a usable gradient is a genuinely good question, and it is the next lecture's; the tutorial will make you crack it on paper.

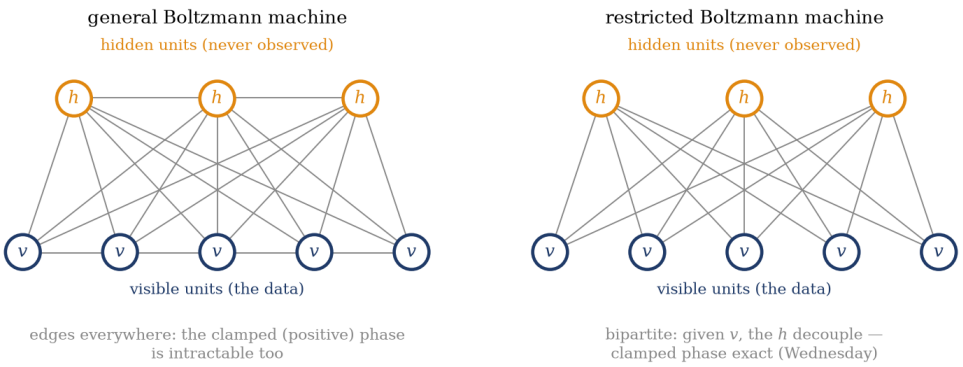

Figure 5.4: Hidden units, and the restriction that will save the positive phase. Left: a general Boltzmann machine — visible units (data) and hidden units (latent), undirected couplings everywhere, including within each layer; the clamped conditional $p(h \mid v)$ couples the hidden units to each other and is intractable. Right: the *restricted* Boltzmann machine — couplings only *between* the layers. Given the visibles, the hidden units decouple, the clamped (positive) phase factorizes exactly, and a whole layer can be sampled in parallel. The bipartite restriction and what it buys are the next lecture's subject.

> **Take-home 3**
>
> Hidden units buy expressive power — marginalizing latents induces effective interactions of all orders among the visibles — at the price of a *second* hard expectation: the clamped positive phase $\langle\cdot\rangle_{p(h|v)}$, intractable for a general graph (and concavity is lost). The bipartite restriction (RBM) makes the clamped phase exact; contrastive divergence makes the negative phase cheap; both are covered in the next lecture. Z itself remains standing, for Module 4.

5.7 Example: the Z ledger

One concrete machine, the negative phase computed exactly — then scaled until it breaks. Take $N = 3$ spins, couplings $J_{12} = J_{23} = 0.5$, $J_{13} = -0.5$, biases $\theta = 0$ — the machine already running in Figure 5.3. The product of couplings around the triangle is negative, so the loop is *frustrated* — no configuration can satisfy all three bonds at once (satisfying all three would cost $E = -1.5$; the true ground states make do with -0.5). The negative phase asks for $\langle s_1 s_2 \rangle_{\text{model}}$, and at $N = 3$ we simply enumerate all $2^3 = 8$ configurations:

$s_1\ s_2\ s_3$	$E(s)$	e^{-E}	$s_1 s_2\, e^{-E}$
$+\ +\ +$	-0.5	1.649	$+1.649$
$+\ +\ -$	-0.5	1.649	$+1.649$
$+\ -\ +$	$+1.5$	0.223	-0.223
$+\ -\ -$	-0.5	1.649	-1.649
$-\ +\ +$	-0.5	1.649	-1.649
$-\ +\ -$	$+1.5$	0.223	-0.223
$-\ -\ +$	-0.5	1.649	$+1.649$
$-\ -\ -$	-0.5	1.649	$+1.649$

$s_1\, s_2\, s_3$	$E(s)$	e^{-E}	$s_1 s_2\, e^{-E}$

Six degenerate low-energy states — the frustrated loop's signature, and Week 2's rugged-landscape story in miniature — and two high-energy ones. Summing the columns: $Z = 6\, e^{0.5} + 2\, e^{-1.5} = 10.34$, and

$$\langle s_1 s_2 \rangle_{\text{model}} \;=\; \frac{2.851}{10.34} \;=\; 0.276$$

— the number the Glauber chain of Figure 5.3 was hunting. The negative phase is a completely well-defined, completely mechanical sum: eight energies, eight exponentials, one division. If the data were to report, say, $\langle s_1 s_2 \rangle_{\text{data}} = 0.50$, the learning rule Equation 5.4 says $\partial L / \partial J_{12} = 0.50 - 0.276 = +0.224$: strengthen the coupling, re-enumerate, repeat. Nothing about the *rule* is hard.

Now scale it. The number of terms in Z — and in every negative-phase expectation, at *every gradient step* — is 2^N:

N	2^N terms in Z	feasibility
3	8	done by hand above
10	1,024	a laptop, instantly
20	$\sim 1.05 \times 10^6$	a laptop, seconds
30	$\sim 1.07 \times 10^9$	a memory-bound minute — the practical ceiling for brute force
100	$\sim 1.27 \times 10^{30}$	more terms than seconds since the Big Bang ($\sim 4 \times 10^{17}$)
500	$\sim 3.3 \times 10^{150}$	more terms than atoms in the observable universe ($\sim 10^{80}$)

The last two rows set the scale. A machine of 100 units — not a large model; a thumbnail image — asks, for its *exact* negative phase, for more terms than there have been seconds since the Big Bang. At 500 units the count dwarfs the atom inventory of the observable universe. And this is not a one-time cost: the trap of Section 5.5 says the sum is owed *per gradient step*, because the distribution under the bracket moves with the parameters. Figure 5.5 draws the scaling.

> *Learning is easy where you average over data and impossible where you average over the model — and that gap is the partition function.*

The warm-up card (due the evening before the next lecture, with the Hinton reading) puts you at this wall deliberately: a tiny machine with hidden units, Z computed by hand, then scaled up until it breaks — the ledger above, with the hidden-unit sum of Section 5.6 layered on top. The workaround arrives in the next lecture; for now the obstruction stands on its own.

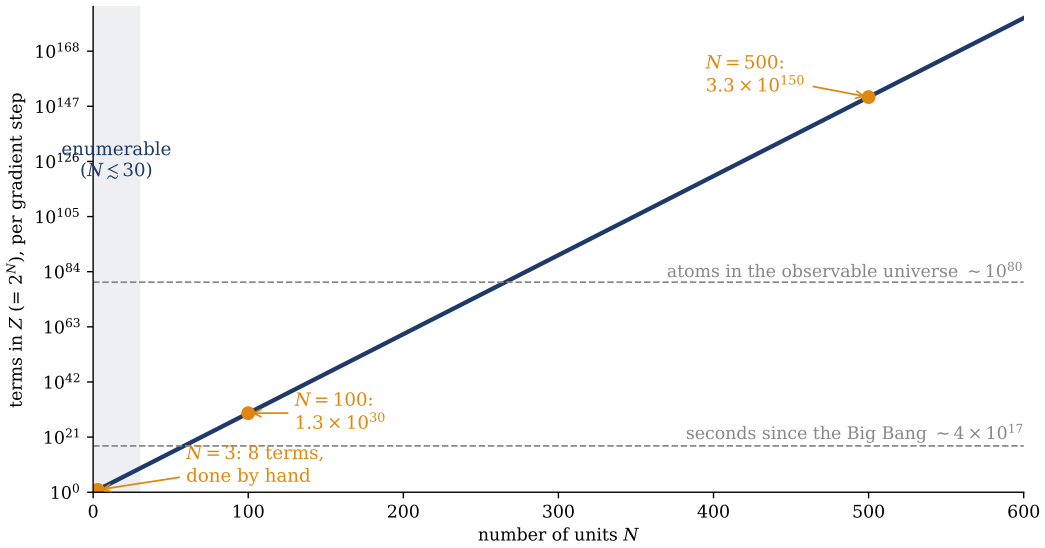

Figure 5.5: The Z wall: the number of terms in the partition function — and in every exact negative-phase expectation, at every gradient step — against the number of units N. Exhaustive enumeration is comfortable in the shaded region below $N \approx 30$ and hopeless beyond: by $N = 100$ the sum has more terms than seconds since the Big Bang ($\sim 4 \times 10^{17}$), by $N = 500$ more than atoms in the observable universe ($\sim 10^{80}$). The learning rule Equation 5.4 is one line; the negative phase inside it is this curve. The rest of the course is strategies for the region right of the shading.

5.8 Outlook: two escapes, and a far arrow

We heated Week 2's network and recovered Week 1's distribution — the course's two opening moves now joined in one machine. We asked that machine to learn and obtained a physically transparent learning rule: data correlations minus model correlations, wake minus sleep. And we derived, rather than asserted, why that rule cannot be run as written: its second term is $\nabla_\theta \log Z$, an exponentially large sum, owed anew at every step — with a second difficulty, mixing, standing behind the first. The practitioner's summary: *an energy-based model is easy to write down and intractable to normalize, and every training algorithm you will meet from here on is a strategy for the negative phase.*

The next lecture addresses the wall in two moves already named: the bipartite restriction that makes the clamped phase exact and lets whole layers sample in parallel, and contrastive divergence — a deliberately short, deliberately biased chain started at the data (Hinton 2002) — with the good question of *why* a truncated chain is allowed at its center. The tutorial works through the RBM gradient, identifying and evading the Z-term.

One forward pointer, now precise. The negative phase exists because we differentiate $\log Z$ *with respect to the parameters*: $\nabla_\theta \log Z = \langle \cdot \rangle_{\text{model}} \neq 0$. In Week 10, score matching will fit the very same unnormalized energies by working instead with the gradient *with respect to the configuration* — and there $\nabla_x \log Z = 0$, identically, because Z does not depend on where you evaluate the density (Hyvärinen 2005). The claim is precise: Z is not made zero — a *different derivative* of it is zero, and

shifting the derivative from θ to x without losing the ability to learn is the whole trick. The derivation comes in Week 10.

6 The restricted Boltzmann machine: one wall falls, one we walk around

Recommended reading

Core (~1 h). Hinton (2002) — the contrastive-divergence paper and the products-of-experts framing (an RBM is a product of experts, one per hidden unit). This was the warm-up reading, so it is already behind you; re-skim the CD section, which this lecture makes precise. Alongside it, Mehta et al. (2019), §VII — the physicist's account of the RBM, the factorized conditionals, and CD, in the course's notation; our derivation follows its logic.
Optional background. Smolensky (1986) — the original "harmonium," the RBM before it had the name, worth reading for the bipartite idea in its birthplace; Hinton (2012) — the engineering companion, CD-k in practice and the gap between clean theory and what actually trains; Hinton and Salakhutdinov (2006) — the stacked-RBM result in *Science* that reignited deep learning, the historical culmination of today's machinery; MacKay (2003), Ch. 43, for the maximum-likelihood gradient done carefully.
Prerequisite reminder. All of Chapter 5 — the model, the two-phase gradient Equation 5.4, hidden units and the two-expectation problem, and the 2^N ledger — plus the logistic-conditional fact from Section 5.3 and the free energy $F = -T \log Z$ from Week 1. New today: conditional independence from graph structure, block Gibbs sampling, a closed-form marginal free energy, and the CD-k approximation with a quantitative account of its bias.

6.1 Two walls, one already read

The previous lecture left the room facing two walls, and we restate them in three lines because today is organized around which of them falls. ① The learning rule: $\Delta\theta \propto \langle\cdot\rangle_{\text{data}} - \langle\cdot\rangle_{\text{model}}$, transparent and exact. ② With hidden units, the *positive* phase is no longer a lookup: it clamps the visibles and averages over $p(h \mid v)$ — intractable for a general graph. ③ The *negative* phase is $\nabla_\theta \log Z$, a sum over 2^N configurations — intractable regardless of architecture. One wall falls to a *theorem* and the other we *walk around* — the difference between exact and approximate.

You have, unusually, already read today's reading. The warm-up card had you compute Z for a toy machine with hidden units and watch the sum blow up in your hands, with Hinton (2002) alongside — so you have met both the wall and the workaround's name. (The natural stuck point is *why a short chain is allowed at all* — a question the tutorial is built on.) The two results follow: the architecture that makes half the problem exact, and the approximation that makes the other half cheap.

The history has three stages. Smolensky (1986) built the bipartite machine first, inside the connectionist collection that named the era, and called it the *harmonium* — the restricted Boltzmann machine before it had the name. Hinton (2002) gave it a fast training rule, contrastive divergence, as a by-product of a more general framework (products of experts: the RBM's hidden units multiply together like independent critics, one constraint each). And in 2006 the combination broke through: Hinton and Salakhutdinov (2006) stacked RBMs, trained them greedily layer by layer with CD, and used the stack to initialize a deep autoencoder that beat PCA soundly — on the pages of *Science*. That result, with its companion deep-belief-net paper (Hinton, Osindero, and Teh 2006), is widely credited with reigniting deep learning, and the architecture we derive today is where it started.

The thesis: *forbidding within-layer connections makes inference exact and sampling fast; computing Z stays impossible; so we approximate the one average we cannot take — and accept a known bias for a workable algorithm.* We are still in Module 1, still at the learned-energy hinge — but this is the module's first *escape technology*, and the nature of the escape matters.

> Take-home 1
>
> An RBM is a Boltzmann machine with the within-layer couplings removed. That single restriction makes $p(h \mid v)$ factorize — exact positive phase, fast layer-parallel sampling — while leaving Z exactly as intractable as in the previous lecture. Architecture defeats one wall exactly; the other admits only approximation.

6.2 Restrict versus approximate

The two moves are *independent ideas with opposite characters*, and keeping their characters apart is most of the understanding.

The first move is **architectural, and exact**. Remove the visible–visible and hidden–hidden couplings, and — as we prove in a moment — given one layer, the units of the other become *conditionally independent*: the clamped positive phase collapses to a formula, and a whole layer can be resampled in one parallel shot. This costs nothing but expressivity-per-layer (recovered, historically, by stacking), and it is a theorem about factorization, not an approximation.

The second move is **algorithmic, and biased**. No architecture computes Z — the negative phase still demands equilibrium samples from the model. Contrastive divergence declines to pay for equilibrium: it runs the sampler a *few steps* from the data and uses whatever it reaches. That is a workaround, and its bias is real; we will quantify it below. Note the independence of the two ideas: CD would apply to any Boltzmann machine; the bipartite structure is what makes CD's Gibbs steps cheap and the positive phase free. The RBM + CD combination is a fit, not a package deal.

Figure 6.1 draws the central picture — both halves of it. On the left, the bipartite graph: a visible layer v carrying the data, a hidden layer h, couplings W_{ij} only *between* the layers, and the forbidden within-layer edges crossed out — those crosses are the entire design. On the right, the consequence: the *block-Gibbs ladder* $v^{(0)} \to h^{(0)} \to v^{(1)} \to h^{(1)} \to \cdots$, each arrow resampling one full layer from its factorized

conditional. The next section treats the left half (why the layers factorize, and what that buys), and contrastive divergence is a statement about the right half (how far up the ladder we bother to climb).

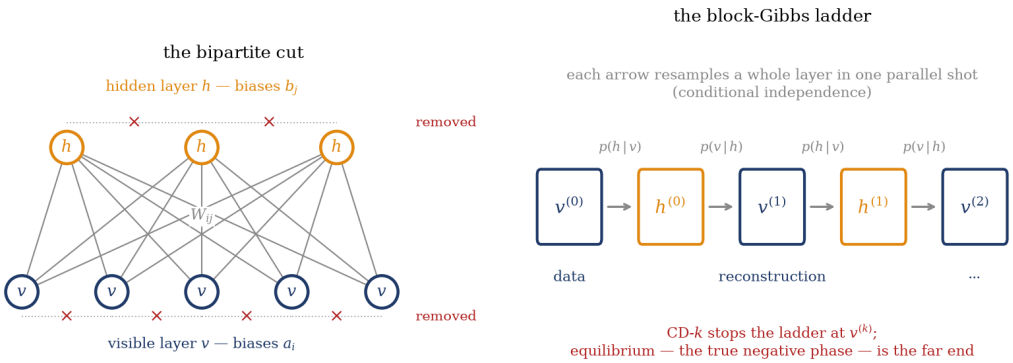

Figure 6.1: The central picture of the lecture. Left: the bipartite cut — visible units v (biases a_i), hidden units h (biases b_j), weights W_{ij} only between the layers; the crossed-out within-layer edges are the restriction that gives the RBM its name and all of its tractability. Right: the block-Gibbs ladder — because each layer is conditionally independent given the other, the alternating chain $v^{(0)} \to h^{(0)} \to v^{(1)} \to \cdots$ resamples a whole layer per arrow. The data sit at $v^{(0)}$; the one-step reconstruction is $v^{(1)}$; the true negative phase lives at the far, equilibrium end of the ladder — and CD-k stops climbing after k rungs.

6.3 The engine: bipartite structure and its consequences

The previous lecture's obstacle: the hidden-unit positive phase needed $p(h \mid v)$, intractable on a general graph. *What is the least structure we can remove to make that conditional trivial?* Answer: the within-layer edges — and here is why.

Setup, with a convention shift flagged. RBM units are binary $\{0, 1\}$ — active or silent — rather than the ± 1 spins of Weeks 2–3. This is the convention of the entire RBM literature and of the tutorial, so we adopt it; the affine map $s = 2v - 1$ converts one convention to the other, renormalizing couplings and shuffling constants into the biases (mix the two silently and it is precisely the biases that get corrupted). With visible biases a_i, hidden biases b_j, and a weight matrix W_{ij}, the energy is

$$E(v, h) \;=\; -\sum_i a_i v_i \;-\; \sum_j b_j h_j \;-\; \sum_{ij} v_i \, W_{ij} \, h_j \tag{6.1}$$

RBM energy

— no $v_i v_{i'}$ terms, no $h_j h_{j'}$ terms. The key observation is immediate: *for fixed v, this energy is linear in h* (and symmetrically, for fixed h, linear in v). Group the h-dependent terms accordingly,

$$E(v, h) \;=\; -\sum_i a_i v_i \;-\; \sum_j c_j(v) \, h_j, \qquad c_j(v) \;=\; b_j + \sum_i v_i W_{ij},$$

$c_j(v)$: the effective field on hidden unit j

where $c_j(v)$ collects everything hidden unit j feels — its bias plus the weighted activity of the visible layer.

The conditional factorizes. Compute $p(h \mid v)$ head-on. The factor $e^{\sum_i a_i v_i}$ is common to numerator and denominator and cancels; what remains splits *per hidden unit*, because both the exponential of a sum and the sum over $h \in \{0,1\}^{n_h}$ factorize:

$$p(h \mid v) \;=\; \frac{e^{\sum_j c_j h_j}}{\sum_{h'} e^{\sum_j c_j h'_j}} \;=\; \prod_j \underbrace{\frac{e^{c_j h_j}}{\sum_{h'_j \in \{0,1\}} e^{c_j h'_j}}}_{=\,1+e^{c_j}} \;=\; \prod_j p(h_j \mid v).$$

Reading off the $h_j = 1$ entry, and running the identical algebra with the roles of the layers exchanged:

$$\boxed{p(h_j = 1 \mid v) \;=\; \sigma\Big(b_j + \sum_i v_i W_{ij}\Big), \qquad p(v_i = 1 \mid h) \;=\; \sigma\Big(a_i + \sum_j W_{ij} h_j\Big).} \tag{6.2}$$

RBM conditionals — the master factorization

In words: given the other layer, each unit is an *independent logistic neuron* — the sigmoid of Section 5.3, a conditional Boltzmann probability for one two-state degree of freedom in the field of its neighbors, now sitting inside a learning machine. Conditional independence is the consequence of the bipartite cut, and it has three uses.

The positive phase is exact. Clamp v to a data point. Because the h_j are independent given v, the clamped expectation needs no sum over hidden configurations at all: $\langle v_i h_j \rangle_{h|v} = v_i\, \sigma(c_j(v))$, a number. Averaging over the training set,

$$\langle v_i h_j \rangle_{\text{data}} \;=\; \frac{1}{D} \sum_d v_i^{(d)}\, \sigma\big(c_j(v^{(d)})\big) \tag{6.3}$$

exact positive phase

— no sampling, no Z, cost linear in the dataset. The wall the previous lecture raised for the general graph is *gone*, by structure. (This is also, note, the general two-phase gradient of Section 5.6 specialized to the RBM: $\partial \log p(v)/\partial W_{ij} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$, with the first bracket now a formula.)

Block Gibbs sampling. The same independence makes the sampler fast: draw all of h from $p(h \mid v)$ in one parallel shot (n_h independent coin flips with probabilities $\sigma(c_j)$), then all of v from $p(v \mid h)$, and alternate — the ladder of Figure 6.1. Each half-update resamples a block of variables from its exact conditional, which leaves the joint $p(v,h) = e^{-E}/Z$ invariant (resampling h from $p(h \mid v)$ cannot change a distribution that already factorizes as $p(v)\,p(h \mid v)$ — the one-line stationarity argument; irreducibility and aperiodicity hold as in Section 5.3 since every conditional probability is strictly positive). This is Glauber's dynamics (Glauber 1963) upgraded from one spin per move to one *layer* per move — the sampler that every downstream method, starting with CD, will lean on.

Tracing out the hidden layer. The factorized sum that gave us the conditionals gives us one more thing, and it is a result of directly statistical-mechanical character.

Sum the joint weight over *all* hidden configurations — the same per-unit split as before:

$$\sum_h e^{-E(v,h)} \;=\; e^{\sum_i a_i v_i} \prod_j \sum_{h_j \in \{0,1\}} e^{c_j(v)\, h_j} \;=\; e^{\sum_i a_i v_i} \prod_j \left(1 + e^{c_j(v)}\right),$$

so the marginal distribution of the visibles is exactly Boltzmann in an *effective energy*, $p(v) = e^{-F(v)}/Z$, with

$$F(v) \;=\; -\sum_i a_i v_i \;-\; \sum_j \operatorname{softplus}\left(b_j + \sum_i v_i W_{ij}\right), \qquad \operatorname{softplus}(x) = \log(1 + e^x). \tag{6.4}$$

RBM free energy

Physically, $F(v) = -\log \sum_h e^{-E(v,h)}$ is a *free energy* — we integrated out the hidden ("fast") degrees of freedom to obtain an effective theory over the visible ("slow") ones, the same coarse-graining move statistical mechanics makes whenever it trades microscopic detail for an effective potential. Each hidden unit contributes one softplus term — Figure 6.2 draws the function and the landscape it builds — and the softplus is where the previous lecture's promise is kept: $c_j(v)$ is linear in v, but softplus is not, and expanding it generates products of visibles of *all orders*. The effective higher-order interactions that hidden units were bought for are now explicit, computable, and sitting in one line.

The contrast is immediate. $F(v)$ is a closed-form expression — evaluate it at any v in microseconds. The normalizer $Z = \sum_v e^{-F(v)}$ is a sum over 2^{n_v} visible configurations — as intractable as ever. *Everything about this model is computable except its normalizer.*

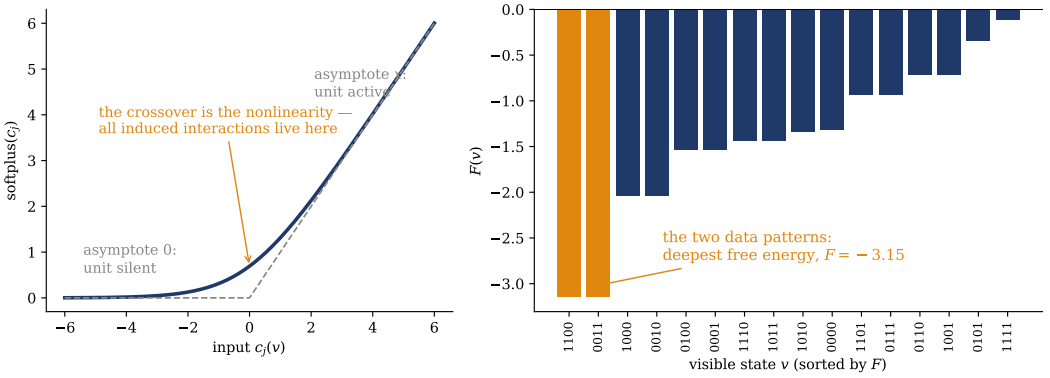

Figure 6.2: The coarse-graining, drawn. Left: $\operatorname{softplus}(x) = \log(1+e^x)$, the effective potential a single hidden unit contributes to $F(v)$ once traced out — flat (≈ 0) when its input is strongly negative, linear ($\approx x$) when strongly positive, with the crossover at $c_j = 0$ carrying all the nonlinearity (and hence all the induced higher-order interactions among the visibles). Right: the free energy Equation 6.4 of the small RBM of Section 6.6, evaluated in closed form at all $2^4 = 16$ visible states: the two data patterns (1100) and (0011) are the two deepest states at $F = -3.15$. Every bar is a microsecond's arithmetic; the sum $Z = \sum_v e^{-F(v)}$ over the bars is what remains intractable at scale.

Trap

Conditional independence is not marginal independence. $p(h \mid v)$ factorizes; $p(v)$, $p(v, h)$, and above all Z do **not** — the visibles are strongly coupled *through* the hidden units they share (that coupling is the model's entire expressive power), and Z is still a sum over exponentially many states. "The RBM's partition function factorizes" is the most tempting wrong sentence of the week. The conditionals factorize; the partition function does not.

6.4 Contrastive divergence: dodging the negative phase

Here the lecture changes character: we stop deriving exact results and start approximating. The RBM's structure fixed the positive phase; the negative phase $\langle v_i h_j \rangle_{\text{model}}$ is untouched — it wants an average over the *free-running* model, which means block Gibbs run to equilibrium, at every gradient step, on a landscape that (Week 2) may mix exponentially slowly. Equilibrium is the cost we now refuse to pay. The question Hinton (2002) asked: *what if we run the ladder just a little?*

The CD-k rule. Start the Gibbs chain **at a data point** — not at a random state — and climb k rungs of the ladder: $v^{(0)} = v_{\text{data}} \to h^{(0)} \to v^{(1)} \to \cdots \to v^{(k)}$. Then estimate the negative phase from where the chain got to:

$$\boxed{\Delta W_{ij} \;\propto\; \langle v_i\, h_j \rangle_{\text{data}} \;-\; \langle v_i\, h_j \rangle_{v^{(k)}}} \tag{6.5}$$

contrastive-divergence (CD-k) update

with both brackets evaluated by the exact formula Equation 6.3 (using the hidden *probabilities* $\sigma(c_j)$ rather than sampled h_j cuts variance at no cost in expectation — standard practice (Hinton 2012)). CD-1 — one up–down–up pass — is the workhorse: the negative sample is the model's *one-step reconstruction* of the data, and the update pushes energy down at the data and up at the reconstruction. Figure 6.3 places the move on the free-energy landscape, and against the previous lecture's global picture.

Why starting at the data is the clever part. There are two readings, one cheap and one deep. The cheap one: a chain started at the data begins in a high-probability region of a decent model, so k steps of honest Gibbs produce a sample that is "trying" to equilibrate locally — far better than k steps from a random state, which would still be essentially noise. The deep one: *if the model were perfect, the data would already be an equilibrium sample*, one Gibbs step would leave its distribution unchanged, the two brackets in Equation 6.5 would agree in expectation, and the update would vanish. The maximum-likelihood solution is therefore (approximately) a fixed point of CD — the rule is engineered to stop pushing exactly when there is nothing left to learn. When the model is wrong, one step drifts the data toward the model's *nearby* fantasies, and CD raises the energy precisely there: a negative phase sampled where it matters most, at the front line between data and model, rather than uniformly over the model's whole support.

The price, stated precisely. CD is *biased*, and the bias is not a rounding error — it is structural, and three progressively sharper statements pin it down. First, the CD-k update is **not the gradient of the log-likelihood**: the true negative phase averages over the model everywhere; CD's averages only over what a k-step

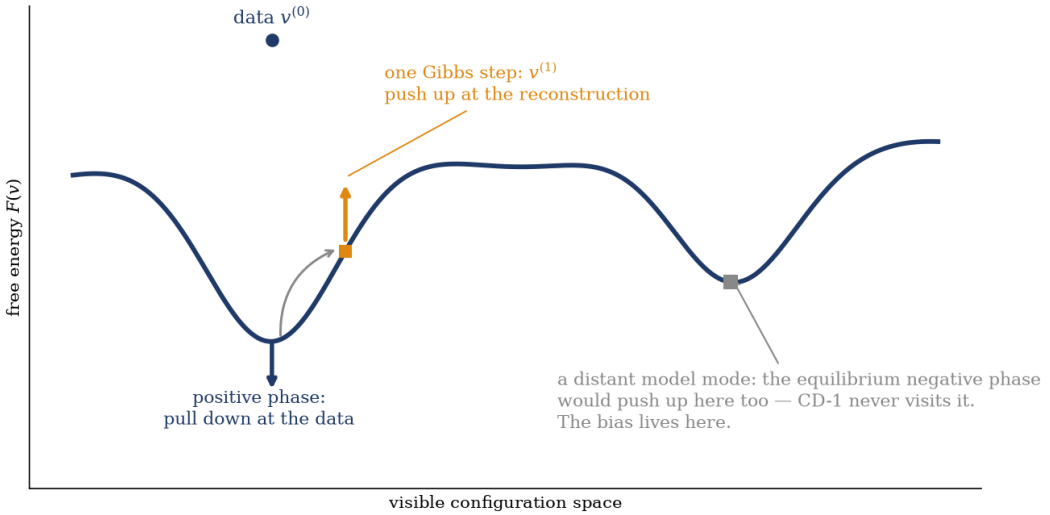

Figure 6.3: What CD-1 actually does, drawn on the visible free-energy landscape $F(v)$ — compare Figure 5.2, where the negative phase pushed up at *all* the model's modes. CD starts at the data, takes one Gibbs step, and pushes up at the resulting reconstruction: a *local* negative phase, applied where the model's probability mass sits *near the data*. If the model is right, the step doesn't move ($v^{(1)} \approx v^{(0)}$) and the two arrows cancel. The price is drawn in gray: a distant spurious mode that equilibrium sampling would find and suppress is never visited by a short chain — CD' 's bias lives exactly there.

chain reaches (the gray mode of Figure 6.3 is real, and a landscape's distant spurious modes are exactly what short chains never police). Hinton's own motivation — that CD-k approximately descends the *contrastive* objective $D_{\mathrm{KL}}(p_0 \| p_\infty) - D_{\mathrm{KL}}(p_k \| p_\infty)$, the amount by which k Gibbs steps move the data distribution toward the model — is illuminating but heuristic, and he said so (Hinton 2002). Second, it is not the gradient of that objective either, nor of *any* objective: Sutskever and Tieleman (2010) showed the CD update is not the gradient of any function at all — there is no hidden loss being optimized. Third, the effect is measurable: the fixed points of CD learning differ from the maximum-likelihood solutions, an effect analyzed and quantified on exactly-solvable small models by Carreira-Perpiñán and Hinton (2005) — whose diagnosis is also the reassurance: the bias is usually small, shrinks as k grows (as $k \to \infty$ the chain equilibrates and CD-k *is* maximum likelihood), and rarely matters for learning good *features*, though it does distort the learned *distribution*.

The standard modern mitigation costs one sentence: **persistent CD** (Tieleman 2008) keeps a set of "fantasy" chains running *across* parameter updates — never resetting to the data — so the negative-phase samples track the slowly moving model far better than a freshly truncated chain, at essentially no extra cost. (The chains stay approximately equilibrated because each parameter update moves the model only slightly — an adiabatic argument a physicist will recognize.)

Trap

CD never computes Z — and never estimates it. It sidesteps the negative-phase *expectation* by sampling; the normalizer itself is untouched, unknown, and unknowable at scale. Corollary: you cannot evaluate the likelihood of a trained RBM either — training monitors *proxies* (reconstruction error, pseudo-likelihood), and **a low reconstruction error is not a high likelihood**: a model can reconstruct data beautifully while assigning most of its probability mass to garbage it never shows you. Mistaking the proxy for the objective is the standard failure mode of energy-based practice. (How one *does* estimate Z when it truly matters — annealed importance sampling — is Week 11's fluctuation-theorem machinery; noted, not developed.)

Take-home 2

Contrastive divergence estimates the negative phase from a short Gibbs chain started at the data: $\Delta W_{ij} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{v^{(k)}}$. It trades an unbiased-but-intractable gradient for a biased-but-cheap one; the bias is real (CD is not maximum likelihood — not even the gradient of any objective — and its fixed points differ), shrinks with k, and is mitigated by persistent chains. CD **dodges** Z; it never computes it.

6.5 From workaround to revival

What the workaround bought warrants two paragraphs of history, because they explain why a superseded model still owns a week of the course.

Through the 1990s and early 2000s, deep networks could not be trained: backpropagation through many layers lost its gradient, and every serious attempt stalled. The RBM + CD combination broke the deadlock from an unexpected side. Hinton and Salakhutdinov (2006) trained a *stack* of RBMs greedily — each layer learned, by CD, to model the hidden activities of the layer below — then unrolled the stack into a deep autoencoder and fine-tuned; the pretrained network compressed images and documents better than PCA, and, decisively, *trained at all*. The companion paper (Hinton, Osindero, and Teh 2006) built generative *deep belief nets* the same way. For roughly five years, greedy layer-wise RBM pretraining was how one trained anything deep — until better activations, initializations, optimizers, and GPUs (circa 2012) made the scaffolding unnecessary, and the field moved on.

The current status, then: as production models, RBMs and CD are *historical* — you will not deploy one, and this course will not pretend otherwise. We spend a week on them because they are the cleanest habitat of ideas the course cannot do without: energy-based learning, the two-phase gradient, the Z obstruction, sampling as the price of normalization, and contrastive estimation — the move of pitting data against the model's own samples, which echoes through noise-contrastive estimation, GAN discriminators, and the negative sampling of modern representation learning. The RBM is to energy-based learning what the Ising model is to phase transitions: nobody's production system, everybody's understanding.

Normative practice, for the record and the tutorial (the flat ranking a practitioner actually uses): for *feature learning*, CD-1 is usually enough — its bias barely touches

the filters. For *generation*, use CD-k with k in the tens, or better, persistent CD. Sample the hidden states when they are driven by the data — the stochastic binary bottleneck is what stops each hidden unit from passing more than a bit, and all-probabilities chains collapse into deterministic mean-field fixed points — but use probabilities, not samples, for the visible reconstructions and for the hidden activities that enter the final gradient statistics. Monitor reconstruction error and pseudo-likelihood *knowing they are proxies*, and watch for the signature failure where reconstruction keeps improving while the model's samples degrade. All of it is systematized in Hinton (2012) — the rare paper that is accurately titled a *practical guide*. (Script asides, named for completeness: real-valued data uses Gaussian-visible RBMs — binary units are a choice, not a law; and the closed-form $F(v)$ makes a trained RBM usable as a *classifier* by comparing free energies of label-conditioned configurations.)

6.6 Example: watching the bias

One concrete RBM, small enough to enumerate, so that the CD bias — invisible in any real application, because seeing it requires the very sum CD exists to avoid — is *visible*. Take $n_v = 4$ visible and $n_h = 3$ hidden units: $2^{4+3} = 128$ joint states, comfortably inside the enumerable region of the previous lecture's ledger (Section 5.7). The machine is hand-built to be legible: hidden unit 1 is tuned to the pattern "left pair on" ($W_{\cdot 1} = (2, 2, -2, -2)$), hidden unit 2 to "right pair on" ($W_{\cdot 2} = (-2, -2, 2, 2)$), hidden unit 3 weakly coupled ($W_{\cdot 3} = (0.5, -0.5, 0.5, -0.5)$); biases $a_i = -0.3$, $b = (-1, -1, 0)$. The dataset is six points in two clusters — four copies of $(1, 1, 0, 0)$, two of $(0, 0, 1, 1)$. This is a machine already well adapted to its data: brute force gives $Z = 98.9$, and the two data patterns are the two lowest-free-energy states of Figure 6.2, each carrying $p(v) = 0.236$ of the model's mass — though note the model splits its mass *symmetrically* while the data arrive $4 : 2$, so there is still something to learn, and the exact gradient will point at it.

For the single weight W_{11} (first visible unit, first hidden unit), the three ways of producing the two brackets of Equation 6.5:

quantity	how computed	cost	value
positive phase $\langle v_1 h_1 \rangle_{\text{data}}$	formula Equation 6.3	exact, $\mathcal{O}(D\, n_h)$	0.635
negative phase $\langle v_1 h_1 \rangle_{\text{model}}$	brute force, 128 states	exact, $\mathcal{O}(2^{n_v+n_h})$	0.352
CD-1 estimate $\langle v_1 h_1 \rangle_{v^{(1)}}$	one Gibbs step from the data	cheap, $\mathcal{O}(D\, n_v n_h)$	0.490

(For the CD row we tabulate the *expected* CD-1 estimate — the data distribution propagated exactly through one block-Gibbs step, so the number is deterministic; a sampled run scatters around it.) Read the columns across. The positive phase *is* the truth — exact by Equation 6.3, at trivial cost; e.g. clamping $v = (1, 1, 0, 0)$ gives $\sigma(c_1) = \sigma(3) = 0.95$, hidden unit 1 all but certainly on. The middle column is the truth CD cannot afford: at 128 states we can buy it; at 128 units we never will. And the CD-1 column *approximates the middle column while leaning toward the first*: one Gibbs step has not carried the data far into the model, and the

reconstruction statistics still remember where they started. The resulting gradients: exact, $0.635 - 0.352 = +0.283$; CD-1, $0.635 - 0.490 = +0.145$. The CD-1 gradient has the right sign at half the size: it learns in the correct direction and undershoots, which is very much its general character.

Climb the ladder and watch the bias drain — Figure 6.4 plots the expected CD-k estimate against k, sliding from the data value at $k = 0$ (the estimate before any Gibbs step *is* the positive phase) to the exact negative phase as $k \to \infty$:

k	expected CD-k estimate	bias
1	0.490	+0.138
2	0.446	+0.094
5	0.394	+0.042
10	0.364	+0.012
20	0.353	+0.001

The decay is controlled by the chain's *mixing*: this machine's couplings ($|W| = 2$) carve two well-separated modes, and a chain started in one takes many rungs to visit the other in fair proportion — the previous lecture's second obstruction, sampling hardness, quietly setting the price of CD's shortcut. Strengthen the couplings and the bias at fixed k grows; this is the toy version of why CD struggles precisely on the strongly structured models one most wants.

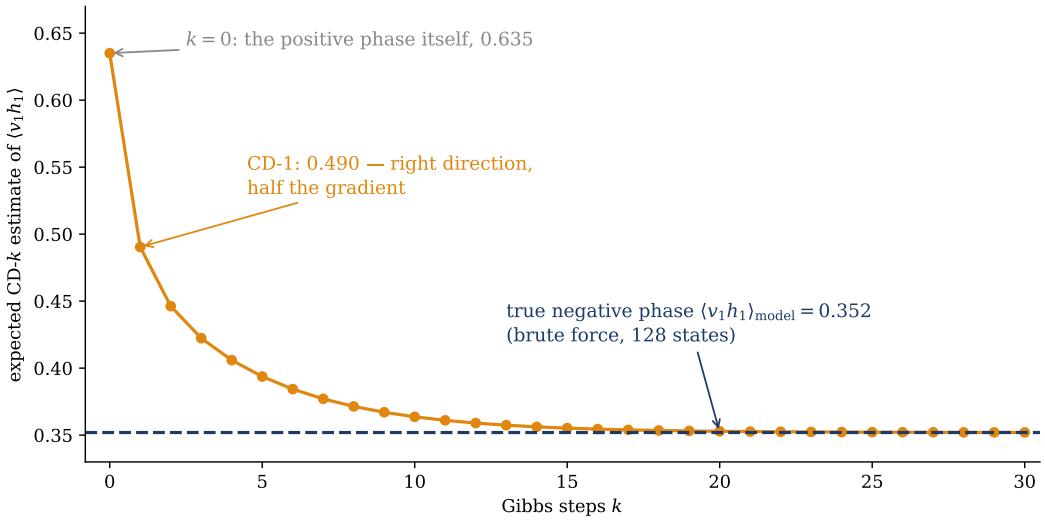

Figure 6.4: The CD-k bias, exactly, on the enumerable RBM of this section ($n_v = 4$, $n_h = 3$; $Z = 98.9$). Orange: the expected CD-k estimate of the negative phase $\langle v_1 h_1 \rangle$, computed by propagating the data distribution through k exact block-Gibbs steps (no sampling noise). At $k = 0$ the estimate is the positive phase itself (0.635); as $k \to \infty$ the chain equilibrates onto the true negative phase (0.352, navy dashed). CD-1 stops one rung up the ladder at 0.490 — right direction, roughly half the true gradient — and the bias decays with k at a rate set by how fast the chain mixes between the machine's two modes. On this toy we can see the whole curve; on any real model, only the k we pay for.

The tutorial (16–18, Ph12 106) runs on exactly this machinery, scaled up: a real RBM, CD-1 against CD-10, learned filters and sampled fantasies side by side. Your

job in the room is to *predict* which is which, and why, before anything renders — the prediction is committed on paper, and only then does each group open a machine and run it. Every formula required is already in this chapter. The hard problem builds on the warm-up's stuck point.

The conditionals factorize; the partition function does not — that gap is exactly what we approximate.

Take-home 3

On a brute-forceable RBM the three columns are all visible: the positive phase is exact by formula, the negative phase is exact by enumeration, and CD-1 approximates the latter with a bias that leans toward the data — right sign, wrong magnitude, decaying with k at the chain's mixing rate. The bias is the measurable price of dodging Z — visible here because the model is tiny, invisible and unavoidable when it is not.

6.7 Outlook: where the obstruction stands

The previous lecture *derived* the learning rule and the obstruction. Today the positive-phase wall **fell** — to a theorem, the bipartite factorization, at no cost in exactness — and the negative-phase wall was **walked around** — by contrastive divergence, at the cost of a real, quantified bias. The partition function itself is exactly as intractable as it was in the previous lecture: we have become good at *avoiding* Z, not at computing it. That asymmetry reflects the state of the field, and every module until Week 10 is a better avoidance — mean-field and variational bounds in Module 2, smarter sampling in Module 3.

Next week the energy-based view scales to the present: modern Hopfield networks carry an energy whose capacity is exponential rather than $0.138\,N$, and whose retrieval update turns out to be *exactly* the attention mechanism of the transformer. Week 2's machine never went away; Week 4 makes the connection precise. And **Week 10** keeps the promise the previous lecture made precise: score matching trades $\nabla_\theta \log Z$, which is intractable, for $\nabla_x \log Z$, which is zero — not another approximation but a resolution (Hyvärinen 2005). CD postpones the negative phase; score matching dissolves it.

7 Dense associative memory: the wall was never a law

Recommended reading

Core (~1 h). Krotov and Hopfield (2016) — the founding paper of modern Hopfield networks, and Hopfield's own return to his 1982 model: the energy $E = -\sum_\mu F(\xi^\mu \cdot s)$, the interaction function F as a capacity dial, and the polynomial capacity $\sim N^{n-1}$. Read for the energy and the capacity claim; the engine below makes the mechanism precise. Alongside it, Demircigil et al. (2017) — the exponential-interaction case and the proof that capacity can be *exponential in* N; read the main theorems and the signal-to-noise heart of the proof, which is Week 2's argument with a sharper filter.
Optional background. Ramsauer et al. (2021) — the continuous-state modern Hopfield network; this is *the next lecture's* paper, so skim the energy and the update rule only and stop there, so that the tutorial still has its discovery to make; Krotov and Hopfield (2021) for the unifying "general Hopfield network" framing; and Chapter 4 of this script, whose signal-to-crosstalk machinery is re-used wholesale today.
Prerequisite reminder. All of Week 2 — the Hopfield energy, the overlap m, the load $\alpha = P/N$, the signal-plus-crosstalk split of Section 3.5, the CLT-Gaussian crosstalk of variance α, and $\alpha_c \approx 0.138$ — plus the Gaussian-tail estimates of Chapter 2. New today: the interaction function F as a capacity dial, the amplification of signal over crosstalk by F', and the polynomial and exponential capacity laws.

7.1 The wall was never a law

Recall where Week 2 left associative memory. A network of N neurons and N^2 Hebbian couplings stores $\alpha_c N \approx 0.138\,N$ patterns; past that load, retrieval does not degrade but *ceases to exist* (Section 4.4, Equation 4.6). For decades that number was read as a verdict on the whole idea: associative memory is fundamentally inefficient — a hundred and thirty-eight memories from a million synapses — and the energy-based approach to computation, however beautiful, was capped. The capacity question looked closed, with a disappointing answer.

The verdict was a misreading. The $0.138\,N$ wall is a property of the particular energy function Hopfield happened to write down, and of nothing else. In 2016, Krotov and Hopfield — Hopfield himself, returning to his own model thirty-four years on — showed that sharpening the energy's interaction function lifts the capacity to N^{n-1} for any degree n (Krotov and Hopfield 2016); a year later, Demircigil et al. (2017) proved that an exponential interaction stores a number of patterns *exponential in*

N, with basins of attraction that remain macroscopic. What set the wall was the *shape* of the energy function.

The question, then: *what set the wall, and how do you move it?* Capacity is determined by how sharply each stored pattern's contribution to the energy is peaked around that pattern. There is a function — the *interaction function* F — whose steepness controls this: degree two gives linear capacity, polynomial degree n gives N^{n-1}, and an exponential gives 2^{cN}.

We are still in Module 1, and we have *returned to the cold*: today's network is Week 2's retrieval machine — deterministic, $T = 0$, couplings written rather than learned, no probabilities and no Z (the partition function, central in Week 3, plays no role this week; Section 7.6 says precisely why). In the next lecture the scaled-up retrieval rule will turn out to sit at the center of the architecture that now dominates machine learning. One contrast worth noting now: Week 3's exponential 2^N counted the states in Z and made training hopeless; this week's exponential counts the patterns you can *store* — the same combinatorial explosion, entering with the opposite sign. The ledger below (Section 7.6) makes the comparison quantitative.

> Take-home 1
>
> The $0.138\,N$ capacity is a property of the *quadratic* Hopfield energy, not of associative memory. Generalize the energy to $E = -\sum_\mu F(\xi^\mu \cdot s)$ and capacity becomes a dial on the sharpness of F: quadratic gives $\sim N$, degree n gives $\sim N^{n-1}$, exponential gives $\sim 2^{cN}$. The wall moves — all the way up.

7.2 The model: an energy with a dial

Two pictures organize the lecture, and Figure 7.1 draws them before any formula. In the **classical** landscape (quadratic energy), each stored pattern contributes a *broad, shallow* basin. Broad basins overlap; overlapping basins interfere; and Week 2 quantified the interference — a CLT-Gaussian crosstalk of variance α that drowns the unit signal at α_c (Equation 4.1). In the **dense** landscape (sharp interaction), each pattern contributes a *deep, narrow* well that dominates near itself and dies off elsewhere. Narrow wells do not overlap, so vastly more of them fit before they interfere — and the count will turn out to be exponential. In short: *capacity is basin geometry*, and the interaction function sets the geometry.

The dense-memory energy. Following Krotov and Hopfield (2016), keep everything from Week 2 — N binary neurons $s_i = \pm 1$, P random stored patterns ξ^μ, the overlap as the natural variable — and change one thing: how the overlap enters the energy. Define

$$\boxed{E(s) \;=\; -\sum_{\mu=1}^{P} F(\xi^\mu \cdot s)} \tag{7.1}$$

dense-memory energy

where $\xi^\mu \cdot s = \sum_i \xi_i^\mu s_i$ and F is a smooth, rapidly growing *interaction function* (Krotov and Hopfield also say *separation function* — it separates the matching pattern from the rest, which is exactly the job description). A notation flag, stated once:

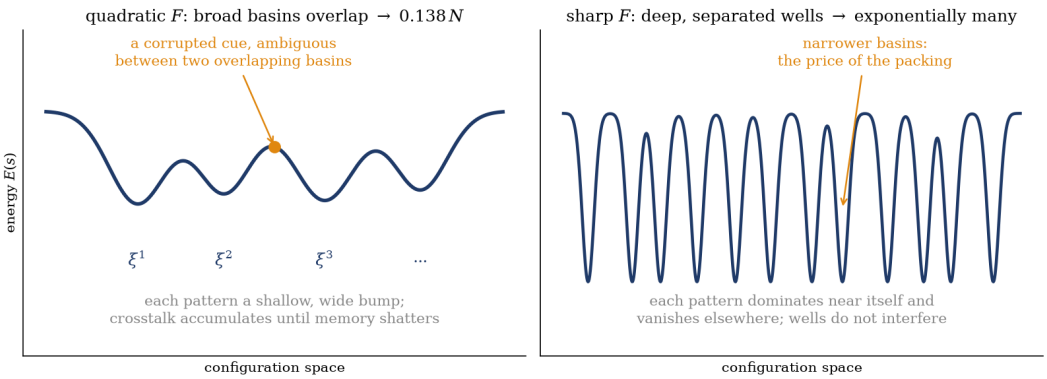

Figure 7.1: The central picture of the lecture: capacity is basin geometry. Left: the quadratic energy of Week 2 — every stored pattern a broad, shallow bump; basins overlap, crosstalk accumulates, and at $\alpha_c \approx 0.138$ retrieval shatters. A corrupted cue sits ambiguously between neighbors. Right: a sharp interaction function — each pattern a deep, narrow, well-separated well that dominates near itself and vanishes elsewhere; exponentially many fit before they interfere. The price is visible on sight: narrower wells mean smaller basins of attraction, the tradeoff quantified below.

this F is not the free energy $F = -T \log Z$ of Weeks 1 and 3 — an unfortunate but standard collision, confined to this week, and the free energy makes no appearance today to collide with.

The classical model is the quadratic special case, and the script does the one line of algebra: with $F(x) = x^2/(2N)$,

$$E = -\frac{1}{2N}\sum_{\mu}\left(\xi^{\mu}\cdot s\right)^2 = -\frac{1}{2}\sum_{i,j}\underbrace{\left(\frac{1}{N}\sum_{\mu}\xi_i^{\mu}\xi_j^{\mu}\right)}_{=J_{ij}\ \text{(Hebb, @eq-hebb)}} s_i s_j = -\frac{1}{2}\sum_{i\neq j} J_{ij}\, s_i s_j - \frac{P}{2},$$

Week 2's Hopfield energy Equation 3.2 up to a constant (the diagonal term, since $s_i^2 = 1$). *Everything Week 2 proved and computed is the $F \propto x^2$ row of a bigger table.*

The update rule. The dynamics generalizes Week 2's greedy descent in the only sensible way: flip neuron i to whichever sign gives the lower energy,

$$s_i \longleftarrow \operatorname{sign}\left[\sum_{\mu}\Big(F(\xi^{\mu}\cdot s|_{s_i=+1}) - F(\xi^{\mu}\cdot s|_{s_i=-1})\Big)\right], \tag{7.2}$$

dense-memory update

asynchronously, one neuron at a time. Two sanity checks, both one line. First, for the quadratic case the bracket collapses (each term is a difference of squares) to $\propto \sum_{\mu}\xi_i^{\mu}\sum_{j\neq i}\xi_j^{\mu}s_j$ — exactly the local field of Equation 3.1: the classical rule falls out unchanged (Demircigil et al. 2017). Second, the Lyapunov argument of Section 3.4 survives verbatim for *any* F: the rule flips only when the flip lowers E, configuration space is finite, so the dynamics descends and halts in a local minimum. The landscape picture is a theorem here too.

For smooth F the bracket linearizes — the two arguments differ by $\pm 2\xi_i^\mu$ — and the update reads $s_i \leftarrow \operatorname{sign}(h_i)$ with the *effective field*

$$h_i \;=\; \sum_\mu F'(\xi^\mu \cdot s)\,\xi_i^\mu \tag{7.3}$$

effective field: an F'-weighted vote

— the central object of the analysis. The field is a *vote of the stored patterns*, exactly as in Week 2 — but now each pattern votes with weight F'(its current overlap). A steep F' means the pattern the state currently resembles dominates the vote, and every other pattern barely registers. The interaction function is a *filter* on the vote, and capacity is decided by how sharply that filter discriminates.

7.3 The engine: signal versus crosstalk, revisited

The derivation repeats Week 2's signal-versus-crosstalk analysis — the same apparatus, the same CLT, the same Gaussian tails — with F' now amplifying the signal. In Week 2 the aligned field was $1 + C_i$ with crosstalk $C_i \sim \mathcal{N}(0, \alpha)$, and signal barely exceeded crosstalk — the margin vanished at α_c. A steeper F' widens that margin.

Step 1 — where the two sides live. Put the network *at* a stored pattern, $s = \xi^\nu$, and read the effective field Equation 7.3. The matching term has $\xi^\nu \cdot s = N$: the signal enters through $F'(N)$. Every other overlap $\xi^\mu \cdot s$ ($\mu \neq \nu$) is a sum of N independent fair signs — Week 2's CLT once more — so it is Gaussian of mean zero and width $\sqrt{N}$: the crosstalk enters through $F'(\mathcal{O}(\sqrt{N}))$. The two sides of the contest sit at *different arguments of the same function*,

$$h_i \;=\; \underbrace{F'(N)\,\xi_i^\nu}_{\text{signal}} \;+\; \underbrace{\sum_{\mu\neq\nu} F'(\xi^\mu \cdot s)\,\xi_i^\mu}_{\text{crosstalk, arguments } \sim\sqrt{N}} \quad ,$$

and the signal-to-crosstalk ratio is governed by the *amplification factor* $F'(N)/F'(\sqrt{N})$. Figure 7.2 draws it for several choices: for the quadratic, $F' = 2x$ is a straight line and the factor is a modest $\sqrt{N}$ — the marginal separation of Week 2. For $F = x^n$ the factor is $N^{(n-1)/2}$; for $F = e^x$ it is $e^{N-\sqrt{N}}$, already $\sim 10^{39}$ at $N = 100$.

Step 2 — the bookkeeping. The scaling argument becomes a capacity number exactly the way it did in Week 2: compute the mean and variance of the stability margin, and ask when the Gaussian tail leaks past zero. Following Krotov and Hopfield (2016), take $F(x) = x^n$ and ask whether bit i of the stored pattern ξ^ν holds — that is, compare the energy of $s = \xi^\nu$ against the energy with bit i flipped. The matching term supplies the *energy gap* (the signal):

$$\langle \Delta E \rangle \;=\; N^n - (N-2)^n \;\approx\; 2n\,N^{n-1},$$

while each of the $P-1$ crosstalk terms shifts by $\approx \mp 2n\,(\xi^\mu \cdot s)^{n-1}$ with $\xi^\mu \cdot s \sim \mathcal{N}(0, N)$. Their fluctuation is set by a Gaussian moment — $\langle z^{2(n-1)} \rangle = (2n-3)!!\,N^{n-1}$ for $z \sim \mathcal{N}(0, N)$, which is where the double factorial that decorates every formula in this field comes from — so the crosstalk variance is

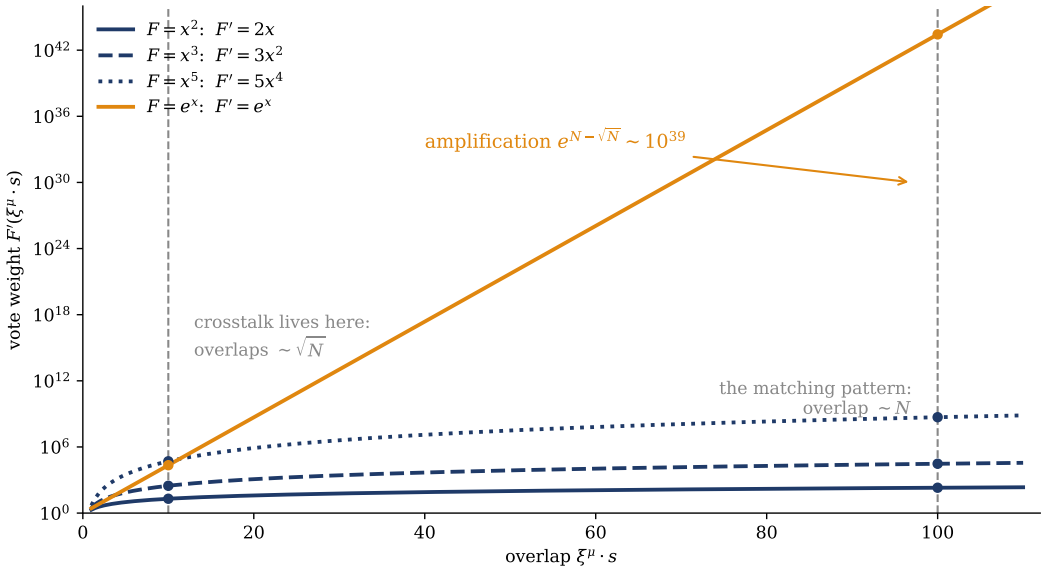

Figure 7.2: The interaction function as a filter, drawn for a network of $N = 100$ neurons. The derivative $F'(x)$ weights each stored pattern's vote in the effective field Equation 7.3; the crosstalk patterns sit at overlaps $\sim \sqrt{N} = 10$ (left dashed line) while the matching pattern sits at overlap $N = 100$ (right dashed line). The signal-to-crosstalk amplification is the vertical gap between the two operating points: a factor $\sqrt{N} = 10$ for the quadratic (the close contest of Week 2), $N = 100$ for the cubic, 10^4 for degree five — and thirty-nine orders of magnitude for the exponential. The capacity laws of Equation 7.4 are this plot, read off.

$$\Sigma^2 \;=\; 4n^2\,(2n-3)!!\,(P-1)\,N^{n-1}.$$

The bit destabilizes when the crosstalk fluctuation beats the gap, which happens with the Gaussian tail probability

$$P_{\text{err}} \;=\; \tfrac{1}{2}\,\text{erfc}(x/\sqrt{2}), \qquad x \;=\; \frac{\langle \Delta E\rangle}{\Sigma} \;=\; \sqrt{\frac{N^{n-1}}{(2n-3)!!\;P}}.$$

As a consistency check: $n = 2$ gives $(2n-3)!! = 1$ and $x = \sqrt{N/P} = 1/\sqrt{\alpha}$ — exactly the first-sweep error rate Equation 4.2, constant and all. The machinery is identical; only the exponent has changed.

Step 3 — read off the capacity laws. Fix a tolerated per-bit error (retrieval with a small dressing of wrong bits, Week 2's tolerant question): then $x = \mathcal{O}(1)$, i.e. $P \sim N^{n-1}$. Demand instead that *every* bit of every pattern hold — the strict question — and the union bound over NP bits costs the usual extreme-value logarithm (Section 4.3, the identical move): $x^2 \approx 2\ln(NP)$, i.e. $P \sim N^{n-1}/(2(2n-3)!!\ln N)$ (Krotov and Hopfield 2016). And for the exponential interaction $F(x) = e^x$, the amplification $F'(N)/F'(\sqrt{N}) = e^{N-\sqrt{N}}$ suppresses the crosstalk to the point that the capacity is exponential — the result Demircigil et al. (2017) made a theorem: storing $P = e^{\alpha N}$ patterns, every one is a fixed point of the dynamics with probability $\to 1$ for any $\alpha < \ln 2/2$, i.e. up to

$$P_{\max} \sim \begin{cases} \alpha_c\, N & F = x^2 \quad \text{(Week 2's wall — recovered, not repealed)} \\ \alpha_n\, N^{\,n-1} & F = x^n \quad \text{[@krotov2016]} \\ e^{\alpha N},\ \alpha < \frac{\ln 2}{2} \;\Rightarrow \sim 2^{N/2} & F = e^x \quad \text{[@demircigil2017]} \end{cases} \tag{7.4}$$

the capacity laws

One caveat, inherited from Week 2. These estimates are *first-sweep, no-feedback* statements — precisely the kind that Section 4.4 showed cannot locate the classical transition, where the errors' feedback moved the answer from $2/\pi$ to 0.138 and changed its character. For $n = 2$, then, read the first row as the scaling with the correct constant *imported* from the self-consistent theory. For $n > 2$ the situation genuinely improves: the crosstalk suppression is now polynomial or exponential rather than marginal, the union-bound argument can be made rigorous — Demircigil et al. (2017) prove both the polynomial law (with $c_n > 2(2n-3)!!$) and the exponential one as theorems — and the first-sweep logic is, for once, the whole story.

The linear derivative $F' = 2x$ is what made the classical capacity linear. Any F whose derivative grows faster lets the matching pattern dominate the crosstalk: polynomial sharpness gives polynomial capacity, exponential sharpness gives exponential capacity.

Figure 7.3 confirms the scaling numerically: a network of $N = 500$ neurons loaded with $P = 1000$ patterns — load $\alpha = 2$, fourteen times past the classical wall — cued with corrupted memories under the classical update and under the cubic dense update Equation 7.2. The quadratic network, far past its capacity, drifts into the spin-glass phase from any starting point. The cubic network repairs a 35%-corrupted cue to $m = 1.000$ in about one sweep.

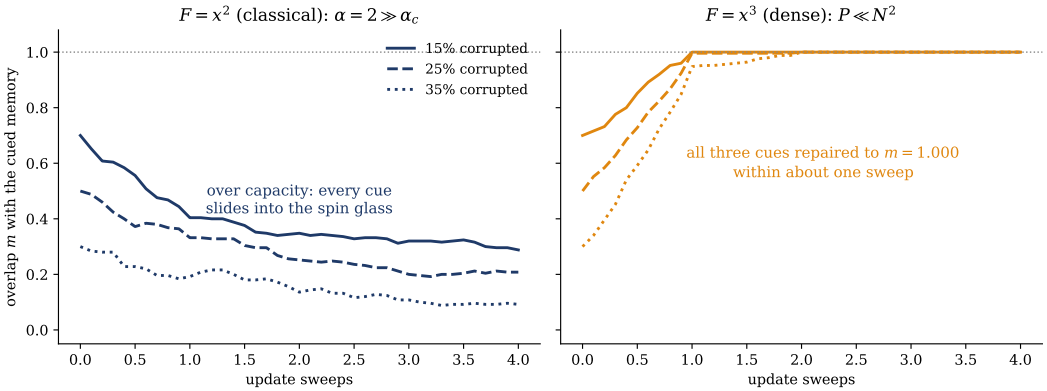

Figure 7.3: The capacity laws, run rather than stated: $N = 500$ neurons, $P = 1000$ stored patterns — load $\alpha = 2$, fourteen times the classical wall $\alpha_c \approx 0.138$ — cued with a stored pattern corrupted in 15%, 25%, and 35% of its bits, under asynchronous energy-descent updates Equation 7.2. Left: the quadratic interaction (classical Hopfield, $\sim N$ capacity): far over capacity, every cue slides away from the memory into spin-glass rubble. Right: the cubic interaction ($\sim N^2$ capacity, comfortably sufficient for $P = 10^3 \ll N^2/(2{\cdot}3!!\ln N) \approx 7{\times}10^3$): every cue is repaired to $m = 1.000$ within about a sweep. Same neurons, same patterns, same dynamics — only the interaction function differs.

7.4 The price: capacity against robustness

Exponential capacity raises the obvious question: *if a sharper F simply stores more, why would anyone ever use anything but $F = e^x$?* The answer is visible in Figure 7.1: the deep, narrow well that lets a pattern ignore its neighbors is also a well that a corrupted cue can *miss*. Sharpening F shrinks the basins of attraction, and a memory with no basin is a lookup table that demands the exact key.

For the exponential model the tradeoff admits an exact formula. The theorem of Demircigil et al. (2017) is quantitative on exactly this point: storing $P = e^{\alpha N}$ patterns, a cue corrupted in a fraction ϱ of its bits is repaired (in one step of the dynamics, with probability $\to 1$) provided

$$\alpha \;<\; \frac{I(1-2\varrho)}{2}, \qquad I(x) = \tfrac{1}{2}\big[(1+x)\log(1+x) + (1-x)\log(1-x)\big] \tag{7.5}$$

the capacity–robustness trade curve

— a strictly decreasing function of ϱ. Figure 7.4 plots the curve, and it makes the tradeoff quantitative. At $\varrho \to 0$ (stability only, no error-correction demanded) the storage exponent reaches its maximum $I(1)/2 = \ln 2/2 \approx 0.347$: the full $2^{N/2}$. Demand that the memory still repair 25% corruption and the exponent drops to $I(1/2)/2 \approx 0.065$ — at $N = 1000$ still a comfortable $\sim 10^{28}$ patterns, which is the content of the paper's title: *huge* capacity **with** macroscopic basins. But push the storage toward its ceiling and the correctable fraction is squeezed to zero: at maximum packing, the wells are pinpricks. Capacity and robustness are two ends of one rope.

The same rope runs along the polynomial family, in a form Krotov and Hopfield turned from bug into insight. At small n, each stored pattern's well is broad, and the network's fixed points blend nearby training examples — retrieval lands on

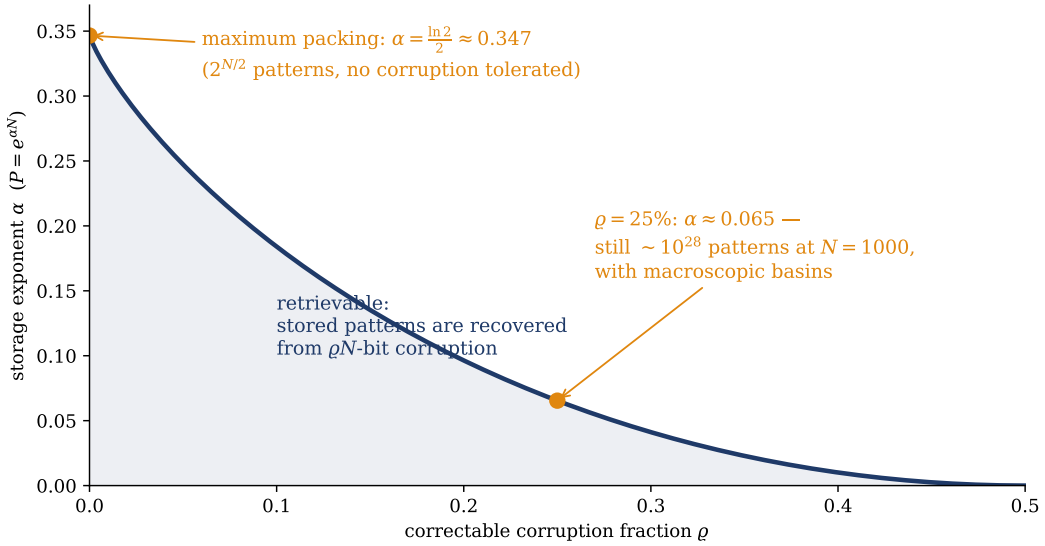

Figure 7.4: The capacity–robustness tradeoff for the exponential interaction, exactly, from Equation 7.5: the largest storage exponent α (capacity $P = e^{\alpha N}$) compatible with repairing a cue corrupted in a fraction ϱ of its bits. At $\varrho = 0$ the exponent reaches $\ln 2/2 \approx 0.347$ — the full $2^{N/2}$, with no error-correction to spare. At $\varrho = 0.25$ the exponent is ≈ 0.065: still $\sim 10^{28}$ patterns at $N = 1000$, with genuinely macroscopic basins. As $\varrho \to 1/2$ (a cue barely better than chance) the storable exponent vanishes. More memories or wider basins — the dial buys one with the other.

something like a class average. At large n, wells are narrow and private — retrieval lands on the individual example. Krotov and Hopfield call the two ends *feature-matching* and *prototype* regimes, and demonstrate on real data (MNIST digits) that the network's character — what kind of representation it learns, how it generalizes, how it errs — changes qualitatively along the dial (Krotov and Hopfield 2016). The interaction function does more than set capacity: it selects *what kind of memory you have*: a few robust, heavily-averaged attractors, or a vast archive of brittle individual ones. Spurious mixture states, Week 2's uninvited guests, populate the low-n end and thin out with sharpness — and the averaging they perform, useless for exact recall, is precisely what starts to look like *generalization* at the feature end. The next lecture develops this connection.

Trap

Exponential capacity is not a free lunch. Sharpening F multiplies the wells and narrows every one of them: the correctable corruption shrinks as the storage exponent grows, with Equation 7.5 the exact exchange rate. A network holding $2^{N/2}$ patterns at maximum packing forgives essentially no corruption — capacity and robustness must always be quoted together.

Take-home 2

The dial has a price: sharper F means deeper, *narrower* basins — more memories, weaker error-correction, with the exact trade curve $\alpha < I(1-2\varrho)/2$ for the exponential case. Along the polynomial family the same rope shows as

Krotov–Hopfield's feature-vs-prototype regimes: broad wells average examples (generalize), narrow wells archive them (memorize). Choose F for the task.

7.5 Toward the next lecture: continuous states, one-step retrieval

Consider once more the exponential filter: $F' = e^x$ weights each stored pattern by the exponential of its overlap, so the effective field is dominated — by the factor $\sim 10^{39}$ of Figure 7.2 at $N = 100$ — by the *single best-matching pattern*. The exponential interaction is a smooth implementation of "return the nearest stored memory," and the sum over patterns in Equation 7.3 is, in disguise, a *soft maximum* over overlaps. Varying the steepness with an inverse-temperature-like parameter interpolates between exact one-winner retrieval and the averaging of the prototype regime.

The next lecture takes this energy and makes two moves. The states become *continuous* vectors rather than ± 1 spins — the energy built on the *log-sum-exp* of the overlaps, the smooth soft-maximum function — and retrieval, which today takes a sweep of single-spin flips, collapses to essentially **one step**: a single update carries a query to (a soft average of) the matching stored pattern, with a sharpness parameter β tuning exact retrieval against averaging (Ramsauer et al. 2021). That update rule is the subject of the next lecture — and it is one you have seen before, under another name.

All of this week's sums run over the P *stored patterns* — and even $2^{N/2}$ of them is a sum you index, not the 2^N sum over *configurations* that constitutes Z. Today's normalizations live in Week 1's tractable regime (a softmax over a finite menu, like the classifier of Section 1.6); nothing here needs, or computes, a partition function. Z is absent this week because retrieval never requires it; it returns when an energy must represent a normalized *distribution* — Week 3's problem, and Module 2's.

7.6 Example: the capacity ledger, and a familiar number

One network — $N = 1000$ neurons, the size Week 2's ledger used — and every capacity law of Equation 7.4 evaluated on it. The printed numbers are the pure scaling laws; the constants of Krotov and Hopfield (2016) (α_n, the double factorial, the strict-stability logarithm) trim one to two orders of magnitude off the polynomial rows, which at this table's scale is a rounding error.

interaction F	capacity law	$P_{\max}$ at $N = 1000$	basins
x^2 (classical)	$\alpha_c N$	≈ 138	wide; robust recall
x^3	$\sim N^2$	$\sim 10^6$	narrower
x^5	$\sim N^4$	$\sim 10^{12}$	narrower still
e^x	$\sim 2^{N/2}$	$\sim 3 \times 10^{150}$	macroscopic *if* α backs off the ceiling (Equation 7.5)

The last row's number appeared before. In Week 3's ledger (Section 5.7), a Boltzmann machine of 500 units had a partition function of $2^{500} \approx 3 \times 10^{150}$ terms — more than the atoms of the observable universe, the number that made training hopeless. Today, a memory of 1000 neurons *stores* 2^{500} patterns — numerically the same. The coincidence is structural: the exponential explosion of configuration space enters the *denominator* when you must normalize (Week 3's problem) and the *capacity* when you get to store into it. Same combinatorics, opposite role. Figure 7.5 draws the four bars, with Week 3's comparators kept deliberately in frame.

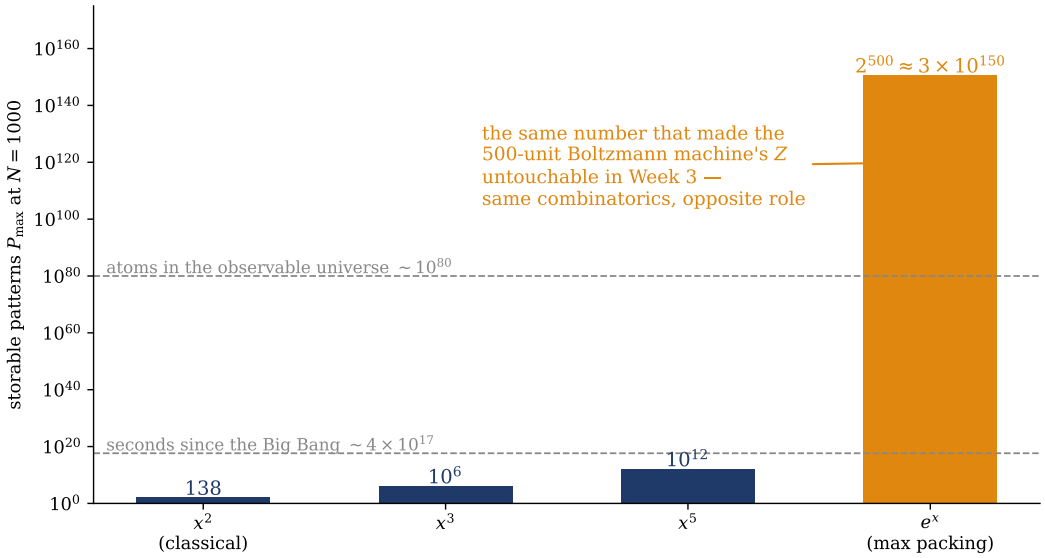

Figure 7.5: The capacity ledger at $N = 1000$, on the log scale it demands. Four interaction functions, four capacity laws: 138 patterns (quadratic — Week 2's wall), $\sim 10^6$ (cubic), $\sim 10^{12}$ (degree five), $\sim 3 \times 10^{150}$ (exponential, at maximum packing $2^{N/2}$). The gray comparators are kept from Week 3's ledger on purpose: the exponential memory stores more patterns than there are atoms in the observable universe — and its capacity, 2^{500}, is numerically *identical* to the count of terms that made the 500-unit Boltzmann machine's partition function untouchable. Same combinatorics, opposite role.

The fourth column qualifies the third. Beside every capacity, its basin: the quadratic network at low load repaired a 49%-corrupted cue (Section 3.6); the exponential network at maximum packing repairs essentially nothing, and buying back a 25% basin costs the exponent a factor of five. The two columns move in opposite directions and should always be read together.

Practical pointers for the week: the warm-up card (due the evening before the next lecture) works from this lecture and the two capacity papers — you will run the F' filter argument yourself on a case the lecture did not do, and meet the tradeoff quantitatively. In the next lecture we make the states continuous and the retrieval collapses to a single step — and the rule that does it is one you will recognize the instant you see it written as a matrix product. The tutorial is derivation-led: the hard problem is cracked at the board. The session is device-free.

> *The* $0.138\,N$ *wall was a fact about a quadratic, not about memory — sharpen the energy and it moves without limit.*

Take-home 3

At $N = 1000$: the quadratic memory holds 138 patterns, the cubic a million, the exponential $2^{500} \approx 3 \times 10^{150}$ — numerically the same count of states that made Week 3's partition function hopeless. The exponential explosion enters the denominator when you must normalize (Week 3) and the capacity when you store into it (Week 4) — same combinatorics, opposite role.

7.7 Outlook: Module 1 closes

Module 1 is now complete. Week 2 **designed** an energy: memory as attractors, one shot of Hebb, capacity capped at $0.138\,N$, with Z absent because nothing was normalized. Week 3 **learned** an energy: the same network heated up became a probability model, and the instant its couplings were fit by likelihood, $\nabla_\theta \log Z$ rose as the training obstruction — a sum over 2^N states. Week 4 **scaled** the energy: back in the cold, a sharper interaction function moved the capacity wall from 138 to beyond the atoms of the universe, and Z dropped out again because retrieval only ever normalizes over the stored patterns. In the next lecture the scaled-up retrieval rule turns out to be the transformer's own retrieval mechanism — and the 2024 Nobel's two halves meet in a single equation.

Module 2 begins when we ask an energy model to represent a *distribution* rather than retrieve a *pattern* — there the normalizer is 2^N again, non-negotiably, and mean-field and variational methods are the first principled machinery for taming it by *bounding* it rather than dodging it as Week 3 did. **Week 10** remains the standing promise: $\nabla_x \log Z = 0$ dissolves the negative phase outright, the resolution that every intervening module's approximation postpones. The two derivatives — with respect to parameters and with respect to inputs — remain distinct.

8 The modern Hopfield network: attention is memory

Recommended reading

Core (~1 h). Ramsauer et al. (2021) — the continuous modern Hopfield network: the log-sum-exp energy, the one-step softmax update, exponential storage, and the attention equivalence; read the energy, the update theorem, and the attention identification — this lecture makes them precise. Alongside it, Vaswani et al. (2017), §3.2 — scaled dot-product attention, the equation we collide with Ramsauer's update. The transformer itself is assumed largely familiar from the companion *Machine Learning and Physics* course; re-read the attention section only.
Optional background. Krotov and Hopfield (2021) — the general "large associative memory" framework (visible/hidden neurons, general Lagrangians) containing both the previous lecture's discrete network and today's continuous one; Yuille and Rangarajan (2003) — the concave–convex procedure behind the update's descent guarantee; Bahdanau, Cho, and Bengio (2015) — where attention entered machine learning, as an alignment mechanism for translation, three years before anyone knew it was a Hopfield update.
Prerequisite reminder. All of Chapter 7 — the dense-memory energy, the F'-filter, exponential capacity, and the metastable averaging of the tradeoff section — plus the softmax-is-Boltzmann corollary of Section 1.6 and scaled dot-product attention from the companion course. New today: the log-sum-exp energy, the one-step softmax update with its descent guarantee, and the identification with attention.

8.1 Two titles, one claim

The previous lecture ended on a deliberately unresolved gesture: make the states continuous, and the retrieval rule collapses to a single step — *a rule you will recognize.* This lecture delivers the identification.

In 2017, Vaswani and coauthors announced to machine learning that *Attention Is All You Need* (Vaswani et al. 2017) — and the transformer architecture built on that paper came to dominate the entire field: language, vision, protein structure, code. In 2021, Ramsauer and coauthors — Hochreiter's group in Linz — replied with a paper called *Hopfield Networks is All You Need* (Ramsauer et al. 2021). **The two titles are one claim.** The attention mechanism at the center of modern AI *is* the retrieval update of a continuous Hopfield network — Week 2's machine, scaled by the previous lecture's dial and made smooth, written in an engineer's notation. (Attention entered machine learning as a learned alignment for machine translation (Bahdanau, Cho, and Bengio 2015); nobody involved was thinking about

spin systems. The identification came after the architecture had taken over, making it a discovery about structure, not a design choice.)

An attention layer is one step of energy descent in an associative memory — a differentiable, content-addressable lookup, the direct continuous descendant of Hopfield (1982). Far from being a historical curiosity capped at $0.138\,N$, the energy-based view of computation — scaled (the previous lecture) and made continuous (today) — is the mechanism at the heart of the architecture that now dominates machine learning. Today's normalizer is a softmax denominator over a *finite* set of stored patterns, Week 1's tractable case; Z remains absent. Module 2, which requires normalized distributions, reintroduces it.

> Take-home 1
>
> *Attention Is All You Need* and *Hopfield Networks is All You Need* are the same claim. A transformer attention layer is one step of retrieval in a continuous modern Hopfield network — content-addressable memory, written in ML notation.

8.2 From hard max to soft max

Two objects that look unrelated — a physicist's *associative memory* (an energy over states, minimized to retrieve a stored pattern from a cue) and an ML engineer's *attention layer* (a query, a stack of keys and values, a softmax-weighted readout) — will each be built from its own side and then set on the same line. Both share a single picture: a query vector reaching into a cloud of stored patterns, a weight on each pattern proportional to how well it matches, and an output assembled as the weighted sum. The sections below construct the picture from each side and identify them.

A single substitution separates the previous lecture's network from the continuous version. The previous lecture's exponential interaction made the effective field a filter, $F' = e^{\text{overlap}}$, which in the sharp limit *selects the best-matching pattern* — a hard max over overlaps. Two things are wrong with a hard max if you want to build it into a neural network: it is not differentiable (no gradients can flow through it), and the previous lecture's retrieval still ran by discrete spin flips, many of them. The fix is one of applied mathematics' oldest moves — replace the max by its *smooth* version:

$$\text{lse}(\beta, z) \;=\; \frac{1}{\beta}\log\sum_{\mu=1}^{M} e^{\beta z_\mu}, \qquad \nabla_z\,\text{lse}(\beta, z) \;=\; \text{softmax}(\beta z),$$

log-sum-exp: the smooth max; softmax: the smooth argmax

the *log-sum-exp*, whose gradient is the *softmax*. As $\beta \to \infty$, lse $\to$ max and softmax $\to$ the one-hot argmax: the previous lecture's hard selection recovered. At finite β, both are smooth, differentiable, and therefore *trainable*. Figure 8.1 draws the pair. And note two familiar objects: for two options, lse *is* the softplus of Section 6.3 — the RBM's traced-out hidden unit was a two-state log-sum-exp all along — and the softmax is Week 1's Boltzmann distribution over a finite menu (Section 1.6), which is precisely why Z poses no problem today.

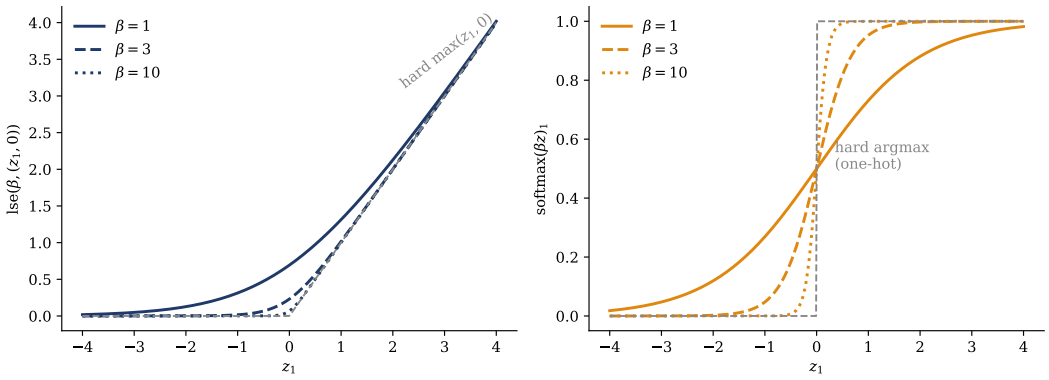

Figure 8.1: The smooth max and the smooth argmax, drawn for two options $z = (z_1, 0)$. Left: $\mathrm{lse}(\beta, z) = \beta^{-1}\log(e^{\beta z_1}+1)$ approaches the hard $\max(z_1, 0)$ (dashed) as β grows — and the $\beta = 1$ curve is exactly the softplus that appeared when Week 3 traced out an RBM's hidden unit. Right: the corresponding weight $\mathrm{softmax}(\beta z)_1 = \sigma(\beta z_1)$ sharpens from a gentle preference to the hard one-hot argmax (dashed step). The continuous modern Hopfield network is the previous lecture's hard pattern-selection with these smooth versions substituted — differentiable, one-step, trainable — and β sets how close to hard.

Three steps remain: ① the energy whose minimization performs this soft selection; ② its one-step update; ③ the identification of that update with attention.

8.3 The engine: the log-sum-exp energy and its one-step update

Setup. Stack the M stored patterns as the columns of a matrix, $X = [\,x_1, \dots, x_M\,]$, each $x_\mu \in \mathbb{R}^d$ — continuous vectors now, not spins. The state is a continuous *query* $\xi \in \mathbb{R}^d$, and $\beta > 0$ is an inverse temperature. (A notation flag, since we adopt Ramsauer's: through Weeks 2–4 the symbol ξ^μ was a *stored pattern* and the count was P; from here ξ is the *query state*, the stored patterns are the x_μ, and their number is M — the relabeling brings us into line with the attention literature we are about to meet.) Following Ramsauer et al. (2021), take the smooth max of the overlaps as the attracting part of the energy, and add a quadratic to keep it bounded below:

$$E(\xi) \;=\; -\,\mathrm{lse}\,(\beta,\, X^T\xi) \;+\; \tfrac{1}{2}\,\xi^T\xi \;+\; \mathrm{const} \tag{8.1}$$

continuous modern Hopfield energy

(The constant, $\beta^{-1}\log M + \frac{1}{2}M_{\max}^2$ with $M_{\max}$ the largest pattern norm, makes $E \geq 0$; it carries no ξ-dependence and will not survive a gradient, so we drop it from every display that follows.) Before differentiating, consider the landscape: $-\,\mathrm{lse}$ digs a well wherever ξ has a large overlap with some stored pattern — one well per pattern, deeper and narrower as β sharpens — and the quadratic term is the bowl that keeps a query from running off to infinity along a pattern direction. This is the previous lecture's picture (Figure 7.1, right panel), drawn in a continuous space.

The gradient, in full. The computation is the chain rule. Component i:

$$\frac{\partial}{\partial \xi_i} \operatorname{lse}(\beta, X^T\xi) \;=\; \frac{1}{\beta}\frac{\sum_\mu \beta\, x_{\mu i}\, e^{\beta\, x_\mu \cdot \xi}}{\sum_\nu e^{\beta\, x_\nu \cdot \xi}} \;=\; \sum_\mu x_{\mu i} \underbrace{\frac{e^{\beta\, x_\mu \cdot \xi}}{\sum_\nu e^{\beta\, x_\nu \cdot \xi}}}_{=\,\operatorname{softmax}(\beta X^T \xi)_\mu}\,,$$

or, assembled over components,

$$\nabla_\xi \operatorname{lse}(\beta, X^T\xi) \;=\; X\,\operatorname{softmax}(\beta X^T\xi), \qquad \Longrightarrow \qquad \nabla_\xi E \;=\; \xi - X\,\operatorname{softmax}(\beta X^T\xi).$$

The stationary condition is the update. Set $\nabla_\xi E = 0$:

$$\boxed{\xi^{\text{new}} \;=\; X\,\operatorname{softmax}(\beta\, X^T\xi)} \tag{8.2}$$

one-step retrieval update

In words: *the retrieved state is a softmax-weighted average of the stored patterns, each weighted by how well it matches the current query.* This is the previous lecture's F'-filter made smooth — score every memory against the cue, exponentiate, normalize, and blend — with softmax playing the role of "select the matching pattern," softly.

Three properties make this equation a usable neural-network layer.

First, iterating the update descends the energy. The energy Equation 8.1 splits by construction into a *concave* piece ($-\operatorname{lse}$; the log-sum-exp is convex, so its negative is concave) plus a *convex* piece ($\frac{1}{2}\|\xi\|^2$). For exactly such splittings, the *concave–convex procedure* of Yuille and Rangarajan (2003) supplies a descent scheme: at the current point, replace the concave piece by its tangent plane — which, by concavity, lies *above* it everywhere — and minimize the resulting convex upper bound exactly. Minimizing $\frac{1}{2}\|\xi\|^2$ minus the linearization $(X\operatorname{softmax}(\beta X^T\xi^t))\cdot\xi$ is a one-line quadratic problem, and its solution is precisely Equation 8.2. Each iterate therefore minimizes a surface that touches E at the current point and majorizes it elsewhere, so E can only decrease: the Lyapunov property of Section 3.4, in its third incarnation — discrete flips (W2), Glauber's stochastic flips (W3), and now a smooth map.

Second, one step is essentially enough. Ramsauer et al. (2021) quantify what "well-separated" buys: define a pattern's *separation* Δ_μ as the gap between its self-overlap and its largest rival overlap, $\Delta_\mu = x_\mu \cdot x_\mu - \max_{\nu\neq\mu} x_\mu \cdot x_\nu$. For Δ_μ sufficiently large (relative to $1/\beta$), the update is a contraction near x_μ, a fixed point exists exponentially close to the pattern, and — the key consequence for practice — *a single application* of Equation 8.2 starting anywhere in the basin lands exponentially close to that fixed point. Retrieval is not Week 2's long relaxation of single-spin flips; it is one differentiable map, evaluated once — which is exactly the property that lets it sit as a layer inside a network trained end-to-end by backpropagation.

Third, the storage matches the previous lecture's. The number of random patterns (say, placed on the sphere) that remain well-separated in the above sense — and hence individually retrievable — grows *exponentially with the dimension* d (Ramsauer et al. 2021): the continuous network inherits the exponential capacity of the $F = e^x$ interaction it descends from, with d playing the role the previous lecture's N played. (For the unified view of why the inheritance is exact — discrete and continuous networks as two choices of "Lagrangian" in one general framework, with the update rules and energies generated mechanically from that choice — see Krotov

and Hopfield (2021), the general treatment behind both of this week's lectures; a script-level pointer, not an examinable one.)

The inverse temperature β controls the behavior. Large β: the softmax approaches a one-hot vector, and the update returns the single nearest pattern — exact retrieval, the previous lecture's sharp limit. Small β: the softmax spreads, and the update returns a broad weighted average of several patterns — the previous lecture's metastable mixtures, now produced deliberately. One parameter sweeps from lookup to blending; in the next section it acquires a famous name.

8.4 This update is attention

The update, read as an engineer would, is a matrix, a softmax, and a matrix — the structure of transformer attention.

Recall, from the companion course, the equation at the center of the transformer (Vaswani et al. 2017). Given a stack of *queries* Q (one per row), *keys* K, and *values* V (one per row), scaled dot-product attention computes

$$\text{Attention}(Q, K, V) \;=\; \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)\, V.$$

Take our update Equation 8.2, transpose it into the same row-vector convention (a query is a row $q = \xi^T$; the stored patterns are the rows of X^T), and it reads $q^{\text{new}} = \text{softmax}\,(\beta\, q\,(X^T)^T)\, X^T$. Set the two equations on one line:

$$q^{\text{new}} = \text{softmax}\,(\beta\, qK^T)\, V \qquad \longleftrightarrow \qquad \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)\, V,$$

and read off the dictionary:

$$Q \leftrightarrow \text{the query state } \xi, \qquad K \leftrightarrow \text{the stored patterns (keys)}, \qquad V \leftrightarrow \text{the stored patterns (values)}, \qquad \beta \leftrightarrow \tag{8.3}$$

attention = one step of Hopfield retrieval

One transformer attention layer is one step of modern Hopfield retrieval. The queries are cues; the keys are the stored patterns the cue is matched against; the values are the stored patterns that get blended into the output; and $1/\sqrt{d_k}$ is the inverse temperature of the lookup. Figure 8.2 draws the identification: one picture, computed once, labeled twice.

For the record — the tutorial develops it at the board, and the script mirrors it here — the *dressed* dictionary, as transformers actually implement it. The correspondence above is the skeleton ($K = V = X$, no learned parameters); a real attention layer inserts three linear maps: $Q = \Xi\, W_Q$, $K = Y\, W_K$, $V = Y\, W_V$, where Ξ holds the querying tokens and Y the attended-to tokens. Read as memory, the projections are natural rather than decorative: W_K chooses *in which learned subspace the matching is scored* (what counts as similar), W_V chooses *what is returned* when a pattern matches (the payload need not be the address), and W_Q maps the current token into the matching space — a content-addressable memory whose addressing scheme

and stored content are themselves learned. *Self-attention* is the case $Y = \Xi$: the sequence is its own memory, every token a query into all the others. *Cross-attention* stores one sequence and queries it with another — an encoder's output retrieved by a decoder, which is exactly the alignment problem attention was invented for (Bahdanau, Cho, and Bengio 2015). *Multi-head* attention runs h such retrievals in parallel, each with its own projections — h associative memories reading the same tokens through different learned lenses, their outputs concatenated — and *stacking layers* is not iterating one memory to convergence but taking single retrieval steps through a *sequence* of different learned memories, each one step deep.

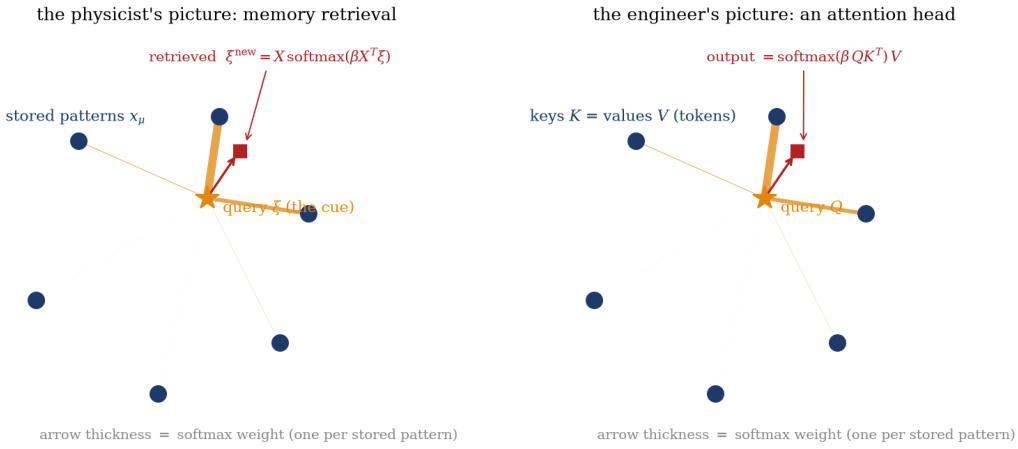

Figure 8.2: One calculation, two labelings. A query reaches into a cloud of six stored patterns; each pattern receives a softmax weight (arrow thickness) according to its overlap with the query; the output is the weighted average, pulled toward the best-matching pattern. Left, the physicist's labels: cue, stored patterns, retrieval $\xi^{\text{new}} = X\,\text{softmax}(\beta X^T \xi)$. Right, the engineer's labels: query, keys and values, attention output $\text{softmax}(\beta\, QK^T)V$. The geometry, the weights, and the output are identical — the relabeling is the theorem.

The role of $\sqrt{d_k}$. The scaling in Vaswani's denominator is introduced, in the original paper, as a numerical fix: dot products of d_k-dimensional vectors with $\mathcal{O}(1)$ components grow like $\sqrt{d_k}$, and without the division the softmax saturates — one weight near 1, the rest near 0, gradients vanishing through the flat tails. Read through today's dictionary, that fix is a temperature: $\beta = 1/\sqrt{d_k}$ holds the retrieval at moderate sharpness as the dimension grows, deliberately keeping attention **out** of the hard-argmax regime. This is a statistical-mechanics temperature inside a normalization constant from Vaswani et al. (2017), made explicit four years later by the Hopfield reading (Ramsauer et al. 2021) — and note the direction of the effect, because it is easy to read backwards: larger d_k means *smaller* β, hence a **softer**, more-averaging lookup. The $\sqrt{d_k}$ prevents over-sharpening; it does not sharpen.

Metastable states as a feature. The previous lecture's mixture states — retrievals that blend several nearby patterns — were discussed as a cost of the exponential interaction. For an attention head they are the desired output: a query (a token in context) rarely wants a single stored pattern; it wants a weighted blend of everything relevant — several previous tokens, in proportion to their relevance. That is precisely a metastable average, produced on purpose at moderate β. The defect of an exact memory is the mechanism of a context-mixer, and the transformer's

own temperature choice parks it deliberately in the mixing regime. Figure 8.3 runs the three regimes on one query.

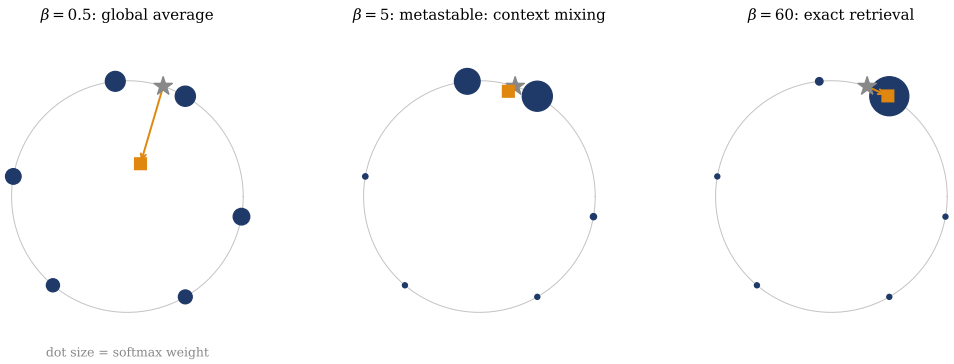

Figure 8.3: The three retrieval regimes of the one-step update Equation 8.2, computed exactly for a query (star) amid six stored patterns (dots, sized by their softmax weight; the orange square is the retrieved ξ^{new}). Two of the patterns are deliberately close, and the query sits near both. Left, $\beta = 0.5$: weights nearly uniform — the update returns a global average, useless as memory. Center, $\beta = 5$: the weight concentrates on the two matching patterns ($w \approx 0.57$ and 0.42) — a *metastable* average, blending exactly the relevant items: this is what an attention head does to its context, and where $\beta = 1/\sqrt{d_k}$ deliberately operates. Right, $\beta = 60$: one-hot weights — exact single-pattern retrieval, the previous lecture's sharp limit.

Two caveats, to state the equivalence precisely. First, **one layer is one step**: a transformer does not iterate Equation 8.2 to a fixed point — it takes a single retrieval step per attention layer and stacks layers, each with its own learned memory. "Attention minimizes an energy" overclaims; *attention takes one descent step on an energy* is the theorem. Second, the softmax denominator here — the sum over the M stored patterns or tokens — is a partition function over a *finite menu*, Week 1's tractable case (Section 1.6), not the 2^N configuration sum of Week 3. Z is genuinely absent from every transformer forward pass; do not import Week 3's obstruction into a lookup that does not have it.

Trap

Two readings that are wrong in opposite directions. ① $\beta = 1/\sqrt{d_k}$ means larger key dimension gives a *softer*, more-averaging attention — the scaling exists to prevent softmax saturation, not to sharpen the lookup. ② One attention layer performs *one* retrieval step, not a relaxation to a fixed point; and in a trained transformer $K \neq V$ in general (separate learned projections of the same tokens). The equivalence is exact about structure and about one step — claim neither more nor less.

Take-home 2

Transformer attention $\text{softmax}(QK^T/\sqrt{d_k})\,V$ *is* the one-step modern Hopfield update $X\,\text{softmax}(\beta X^T\xi)$: query = cue, keys = values = stored patterns (learned projections in general), $\beta = 1/\sqrt{d_k}$. The $\sqrt{d_k}$ is a retrieval tempera-

ture that keeps attention in the soft regime; metastable averaging is context-mixing. Attention is content-addressable memory.

8.5 Example: one calculation, two names

One small memory, the update computed by hand, and the same numbers read twice. Store $M = 4$ patterns in $d = 3$: the unit vectors $x_1 = (1, 0, 0)$, $x_2 = (0, 1, 0)$, $x_3 = (0, 0, 1)$, and the blend $x_4 = (0, 1, 1)/\sqrt{2}$. The query is a corrupted version of the first memory, leaning toward the second: $\xi = 0.8\, x_1 + 0.2\, x_2$, with overlaps $X^T\xi = (0.8,\ 0.2,\ 0,\ 0.14)$. Compute $w = \mathrm{softmax}(\beta X^T\xi)$ and $\xi^{\mathrm{new}} = Xw$ at three temperatures:

regime	β	weights w	ξ^{new}	reading
soft	$1/\sqrt{3} \approx 0.58$	(0.33, 0.23, 0.21, 0.23)	(0.33, 0.39, 0.37)	global average — everything blended
moderate	3	(0.72, 0.12, 0.07, 0.09)	(0.72, 0.19, 0.14)	metastable: x_1 dominant, context mixed in
sharp	20	(1.00, 0.00, 0.00, 0.00)	(1.00, 0.00, 0.00) $= x_1$	exact retrieval — the previous lecture's hard limit

The three rows illustrate the full range: at $\beta = 20$ the memory returns its first pattern *exactly* (the weight on x_1 is 0.99999); at $\beta = 3$ it returns "mostly x_1, with the relevant neighbors mixed in"; at $\beta = 0.58$ it returns an undifferentiated average of everything. Now read the first row's temperature again: $1/\sqrt{3}$ is precisely the transformer's own $\beta = 1/\sqrt{d_k}$ for $d_k = 3$. On this toy the engineer's default temperature sits deep in the averaging regime — by design, and (in a real transformer) with learned projections rescaling the logits to taste around it.

The second reading makes the identification explicit. Set $Q = \xi$ as a row, $K = V =$ the patterns as rows, $d_k = 3$; then $\mathrm{softmax}(QK^T/\sqrt{d_k})\, V$ is — symbol for symbol — the first row of the table. *The physicist retrieved a memory at soft temperature; the engineer computed an attention head; the arithmetic was identical.* The calculation is one; only the labels differ.

The tutorial (16–18, Ph12 106) is derivation-led: both equations, from both fields, are set side by side — deriving attention from the Hopfield side and Hopfield from the attention side, then building out the full dictionary: separate W_Q, W_K, W_V projections, self- versus cross-attention, and multi-head attention as parallel associative memories.

A transformer attention layer is one step of Hopfield retrieval — attention is associative memory.

8.6 Module 1 closes: the landscape, all the way up

Four weeks ago we drew an energy landscape over configuration space. Week 2 **designed** the landscape: Hopfield's memory, minima placed at data by Hebb's rule, retrieval as descent — a machine that computes, capped at $0.138\,N$, with nothing normalized and no Z in sight. Week 3 **learned** the landscape: heat the network and it becomes a probability model; fit its couplings by maximum likelihood and the gradient sprouts $\nabla_\theta \log Z$ — a sum over 2^N states, dodged but not defeated by contrastive divergence. The previous lecture **scaled** the landscape: the interaction function turned out to be a capacity dial, and the wall at 138 patterns moved past the atom count of the universe. And today the landscape **deployed**: made continuous and smooth, its one-step retrieval *is* the attention layer — the mechanism running inside essentially every large model in existence. Figure 8.4 draws the four panels in a row, with the role of Z marked week by week.

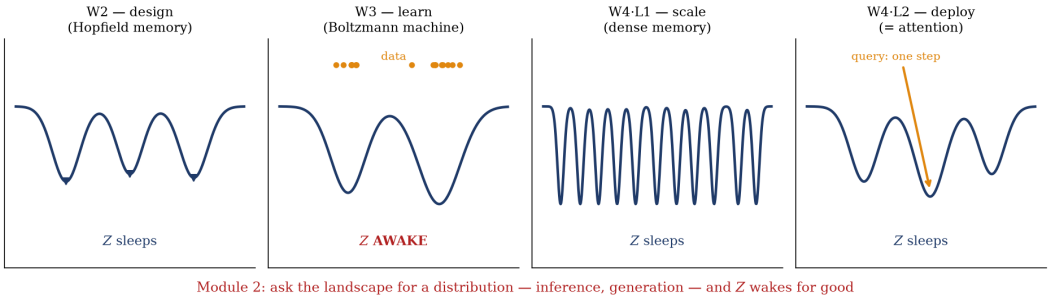

Figure 8.4: Module 1 in four panels — one landscape, four verbs. Week 2 *designs* it (memory basins, nothing normalized); Week 3 *learns* it (basins molded under data, and the partition function enters as the training obstruction); Week 4 *scales* it (dense memory, exponentially many wells) and *deploys* it (a query descending in one step: attention). The pattern is the module's lesson: Z is absent whenever the landscape only *retrieves*, and returns whenever the landscape must carry a normalized *distribution* — which is Module 2's opening condition.

Deep learning's central mechanism is itself energy-based. Hopfield's 1982 landscape and Vaswani's 2017 attention are one picture, thirty-five years apart, and the 2024 Nobel's two halves — the memory and the learning machine — meet in Equation 8.3.

Z was absent in Week 2 ($T = 0$, nothing normalized), central in Week 3 (learning a distribution), and absent again all through Week 4 (finite normalizers — a softmax over patterns, never a sum over configurations). The pattern is exact: **whenever the landscape only retrieves, Z plays no role; whenever the landscape must carry a normalized distribution, Z returns.** Module 2 asks for distributions everywhere — inference over hidden variables, marginals of rugged systems, generative modeling proper — so from next week Z is present throughout, and the response changes character, from Week 3's dodge to the first principled *approximation of the distribution itself*. Week 5 builds mean-field theory and the variational free

energy — the Gibbs–Bogoliubov bound that turns "compute $\log Z$" into "optimize over tractable families"; Week 6 sharpens it (TAP corrections, belief propagation); Week 7 carries it to latent-variable models and the variational autoencoder. And behind all of it stands, unchanged, the far promise this module made precise: what Module 2 approximates and Module 3 samples, Week 10 dissolves — $\nabla_x \log Z = 0$, the derivative that is zero because it moves the configuration and not the parameters.

Take-home 3 — the Module 1 capstone

Module 1 in one line: the energy landscape — *designed* (W2), *learned* (W3), *scaled and deployed* (W4) — is the substrate of modern generative AI, and transformer attention is its retrieval step. The recurring obstruction is Z, which appears exactly when the landscape must carry a normalized distribution; taming it — by approximation (M2), by sampling (M3), by elimination (M4) — is the rest of the course.

Part II

Module 2 — Variational and mean-field methods

9 Mean-field theory: the best tractable lie

Recommended reading

Core (~1 h). Opper and Saad (2001), the editors' introduction — the physics ML mean-field bridge in one book, and the variational-free-energy framing this lecture follows. Alongside it, Wainwright and Jordan (2008), §5 (with §3 for the exponential-family setup) — the machine-learning telling: variational inference as free-energy minimization over a tractable family; read for the ELBO/mean-field correspondence we make exact below. (For a shorter statistician's account, Blei, Kucukelbir, and McAuliffe (2017).)
Optional background. Mehta et al. (2019), the variational-methods sections, in the course's notation; MacKay (2003), Ch. 33, for the KL bound done cleanly from the inference side; Sethna (2021), the mean-field Ising sections, for the physics lineage — Weiss's molecular field and the self-consistent equation; Jordan et al. (1999), the founding machine-learning reference for naive mean field.
Prerequisite reminder. The Boltzmann distribution and $F = -T \log Z$ (Week 1); the Ising energy and the $\langle\cdot\rangle$ bookkeeping (Weeks 2–3); $D_{\mathrm{KL}}(q\|p) \geq 0$, flagged in Week 1 and used in earnest today; the two-state tanh/sigmoid average (Section 1.6, Section 5.3). New content: the variational bound as a controlled approximation, its identity to the ELBO, and the mean-field self-consistent equations.

9.1 The normalizer returns

Module 1 closed with a pattern (Figure 8.4): the partition function plays no role whenever an energy landscape only *retrieves*, and becomes unavoidable whenever it must carry a normalized *distribution*. Module 2 begins where the partition function is unavoidable. Inference over hidden variables, marginals of an interacting model, the likelihood of data under an energy-based model — every one of these asks for $p(x) = e^{-\beta E(x)}/Z$ as a distribution, and therefore for $Z = \sum_x e^{-\beta E(x)}$ over 2^N configurations, and therefore for the free energy $F = -T \log Z$ that Week 1 told us is the object worth wanting and Week 3's ledger (Section 5.7) told us we cannot have. Week 3 dodged the *gradient* $\nabla_\theta \log Z$ by sampling a little and accepting a bias. Now we face $\log Z$ itself — and a different strategy is required.

We will not compute p. We will pick a tractable distribution q and make it the best possible stand-in — and we will do it with a *bound*, so that we always know the sign of our error. Approximations are common; approximations whose error has a *known sign* are rare, because they can be optimized (push the bound as far as it goes and you can only get closer to the truth) and trusted (the

truth is never on the forbidden side). An approximation you can bound is worth more than an exact answer you cannot compute.

The method has a double lineage, and the two lines turn out to be one. On the physics side, Weiss (1907) replaced each spin's fluctuating neighbors by an average "molecular field" and obtained the self-consistent equation that *predicts ferromagnetism* — mean-field theory, the workhorse approximation of a century of statistical physics, later dignified into a variational *bound* by Bogoliubov, Peierls, and Feynman. On the inference side, machine learning rediscovered the same construction in the 1990s as *variational inference* (Jordan et al. 1999; Wainwright and Jordan 2008): approximate an intractable posterior by a tractable family, optimize a lower bound on the evidence. Two communities and two vocabularies arrived at one method — and today the physicist's free-energy minimization and the machine-learner's ELBO maximization are the same optimization of the same functional.

One notational flag before we start. Since Week 3 we have absorbed $\beta = 1$ into the couplings; from today **temperature is a live control parameter again**, written explicitly. Mean-field theory's central application in physics is to locate *phase transitions* — and a transition lives at a critical temperature, which requires an explicit β.

> Take-home 1
>
> When Z is intractable, do not compute p — approximate it by a tractable q, chosen to minimize a **bound** on the free energy. The bound's sign is known ($F \leq F_q$, always), so the approximation is controlled: optimizing it can only approach the truth, never cross it. This is Module 2's entire strategy; mean field is its simplest instance.

9.2 Two roads, and one picture

Week 3 identified two obstructions — normalization, and the sampling hardness that enforces it — and they now index the course's two escape roads. Road one **approximates the distribution**: replace p by a tractable q and optimize — deterministic, fast, and equipped with a bound, at the price of a bias fixed by the chosen family. That is this module. Road two **samples the distribution**: draw configurations from p without ever normalizing it — asymptotically exact, at the price of stochasticity and of mixing time on rugged landscapes. That is Module 3. Today we commit to road one, and within it a second split organizes the lecture: *any* trial q yields the variational bound (the next section); the *factorized* q yields mean-field theory (the section after).

Figure 9.1 draws the physical idea. A spin in an interacting system feels the field of its neighbors — a *fluctuating* field, different in every configuration, correlated with the spin itself. Mean field replaces that fluctuating environment by its average: every neighbor s_j is frozen to its mean m_j, and the spin responds to a single, deterministic *effective field* $h_i^{\text{eff}} = \sum_j J_{ij} m_j + h_i$. The substitution would be exact if the neighbors did not fluctuate; its error *is* the fluctuation. Everything the approximation gets wrong — and the next lecture will show it getting a phase transition's location and character wrong — traces back to this one sentence.

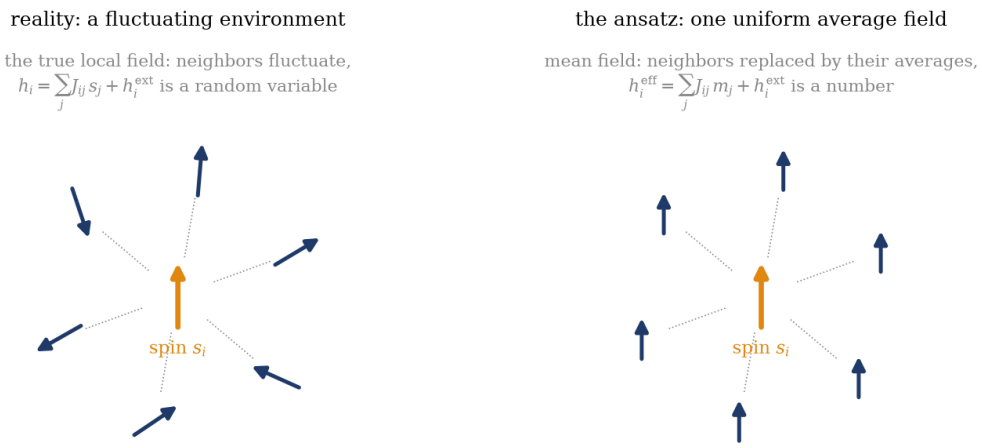

Figure 9.1: The central picture of Module 2's opening, a century old. Left: reality — a spin's local field is built from neighbors that fluctuate, so the field is a random variable, correlated with the spin it acts on. Right: the mean-field ansatz — every neighbor is replaced by its average magnetization, and the spin sits in a single deterministic effective field $h_i^{\text{eff}} = \sum_j J_{ij} m_j + h_i$. The substitution is exact if the neighbors do not fluctuate; the error is precisely the fluctuation — which is why mean field improves with connectivity (many neighbors average sharply) and fails where fluctuations dominate.

The formal version of the same move is a *projection*, and Figure 9.2 draws its geometry. The intractable p is a point in the space of distributions; the tractable candidates $\{q\}$ — say, all factorized distributions — form a surface that does not contain it. Variational inference finds the point on the surface closest to p, with distance measured by the Kullback–Leibler divergence — in a specific argument order whose consequences we will flag with a triangle. The next section supplies what the picture cannot: that the projection distance is exactly a free-energy difference, computable up to a constant *without knowing* Z.

9.3 The engine, part I: the Gibbs–Bogoliubov bound

We want a functional of q that ① is computable whenever q is tractable, ② upper-bounds the intractable F, and ③ touches it exactly at $q = p$. Week 1 supplied every ingredient.

Define the *variational free energy* of an arbitrary trial distribution $q(x)$ over the configurations:

$$F_q \;\equiv\; \langle E \rangle_q \;-\; T\,S[q] \;=\; \sum_x q(x)\,E(x) \;+\; T \sum_x q(x) \log q(x).$$

This is Week 1's free energy $F = \langle E \rangle - TS$ with the Boltzmann distribution replaced by q in both terms — an energy cost plus an entropy price, evaluated under the wrong distribution. Note what is *absent*: no Z. Every term is an average under q, computable exactly for any tractable family.

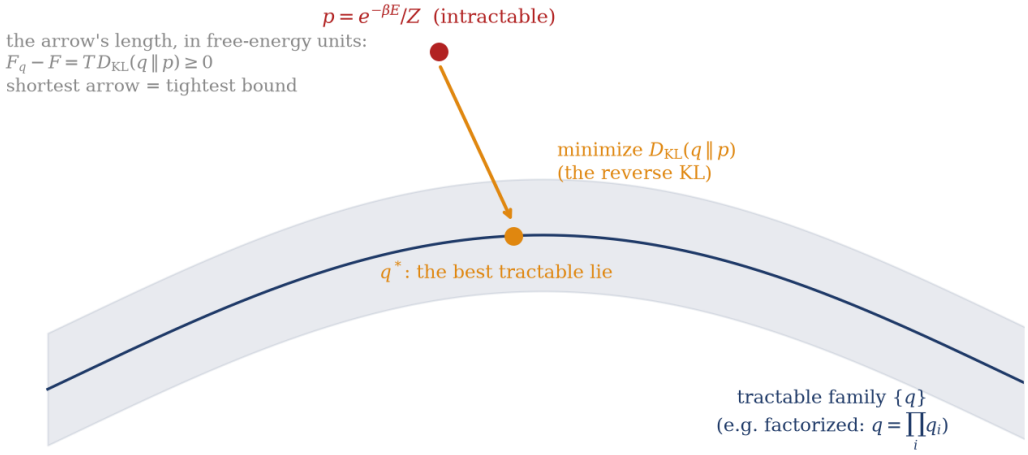

Figure 9.2: The variational principle as geometry: project the intractable p onto a tractable family. The projection minimizes the *reverse* KL divergence $D_{\mathrm{KL}}(q\|p)$ — the only order we can average under — and the arrow's length, measured in free-energy units, is the gap $F_q - F = T\, D_{\mathrm{KL}}(q\|p) \geq 0$. Minimizing the variational free energy over the family therefore does two jobs at once: it finds the best tractable stand-in for p, and it tightens a one-sided bound on the true free energy.

Compute the gap to the true $F = -T \log Z$. Insert $\log Z$ under the q-average (it is a constant), and reassemble:

$$F_q - F \;=\; \langle E + T\log q \rangle_q + T\log Z \;=\; T\Big\langle \log q + \beta E + \log Z \Big\rangle_q \;=\; T\Big\langle \log \frac{q(x)}{e^{-\beta E(x)}/Z} \Big\rangle_q \;=\; T\Big\langle \log \frac{q}{p} \Big\rangle_q .$$

Recognize the object on the right — it is the Kullback–Leibler divergence, and it is non-negative:

$$\boxed{F_q - F \;=\; T\, D_{\mathrm{KL}}(q \,\|\, p) \;\geq\; 0, \qquad \text{so} \qquad F \;\leq\; F_q, \quad \text{with equality iff } q = p.} \tag{9.1}$$

Gibbs–Bogoliubov variational bound

The identity admits three readings. As a *bound*: the variational free energy never dips below the truth — equivalently $\log Z \geq -F_q/T$, a **lower** bound on the log-partition function, the first quantitative grip the course has had on the normalizer. As an *objective*: minimizing F_q over a tractable family simultaneously tightens the bound on F **and** finds the member of the family closest to p in KL — one optimization serving both goals, which is why the method took hold in both fields. As a *diagnostic*: the gap is not vague "approximation error" but a named information-theoretic quantity, and when we can compute it (as in the example below) it tells us exactly how much distribution we have thrown away. (The classic physics statement of the same inequality — $F \leq F_0 + \langle H - H_0 \rangle_0$ for a trial Hamiltonian H_0, the *Peierls–Bogoliubov* form — is the special case $q = e^{-\beta H_0}/Z_0$, traditionally proved

from the convexity of the exponential, $\langle e^X \rangle \geq e^{\langle X \rangle}$; the KL route above is the same mathematics with the inference reading built in, which is why we lead with it.)

Trap

It is the reverse KL, and that is not a detail. The gap is $D_{\mathrm{KL}}(q\|p)$ — averaged under q — not $D_{\mathrm{KL}}(p\|q)$. The order is forced on us: we can only compute averages under the tractable distribution. But the two divergences penalize different sins. The reverse KL explodes where q puts mass and p has none, so the optimal q is *mode-seeking*: it commits to one region p likes and pays little for ignoring the others, and it systematically *underestimates variance*. The forward KL would instead spread q to cover everything (Figure 9.3). Every characteristic failure of the variational family of methods — mean field's overconfident magnetizations below, the VAE's posterior collapse in Week 7 — is this same asymmetry in another form. Know which divergence you are minimizing.

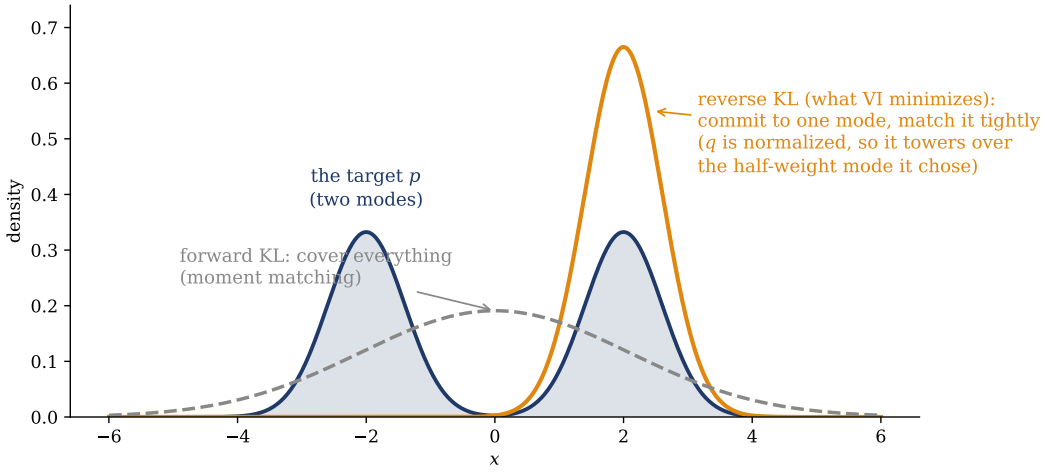

Figure 9.3: The asymmetry of the two KL divergences, on the simplest instructive case: fit a single Gaussian q to a bimodal p (equal-weight modes at ± 2, width 0.6). Minimizing the *forward* KL $D_{\mathrm{KL}}(p\|q)$ (gray dashed) matches moments: the optimum straddles both modes, centered at zero with $\sigma \approx 2.1$ — mass-covering, and placing most of its probability where p has almost none. Minimizing the *reverse* KL $D_{\mathrm{KL}}(q\|p)$ (orange) — the divergence variational inference is forced to use — commits to a single mode and matches it tightly: mode-seeking, and blind to the mode it abandoned. Mean field's overconfidence, and Week 7's posterior collapse, are this picture at scale.

9.4 The engine, part II: the bound is the ELBO

The identification requires no new mathematics — only a redefinition of the energy.

Consider the machine-learner's standard predicament: a *latent-variable model* $p(x, z)$ — observed data x, hidden variables z (cluster labels, latent codes, an RBM's hidden layer) — whose fit to data is measured by the *evidence* $p(x) = \sum_z p(x, z)$, and whose

inferential heart is the *posterior* $p(z \mid x) = p(x, z)/p(x)$. Both are intractable for the same reason Z was: the sum over z.

Fix the observation x and *manufacture an energy* over the hidden variables — not a physical Hamiltonian, just a definition:

$$E(z) \;\equiv\; -\log p(x, z), \qquad T = 1.$$

Then the Boltzmann distribution of this energy is $e^{-E(z)}/Z = p(x, z)/\sum_{z'} p(x, z')$ — *the posterior* — and its partition function is

$$Z \;=\; \sum_z e^{-E(z)} \;=\; \sum_z p(x, z) \;=\; p(x)$$

— **the evidence is a partition function.** Every object in the variational construction now translates mechanically. The variational free energy of a trial $q(z)$ becomes

$$F_q \;=\; \langle E \rangle_q + \langle \log q \rangle_q \;=\; -\Big[\big\langle \log p(x, z) \big\rangle_q - \big\langle \log q(z) \big\rangle_q \Big] \;=\; -\,\mathrm{ELBO}(q),$$

minus the machine-learner's *evidence lower bound*, term for term. And the master identity Equation 9.1, transcribed, is the ELBO inequality with its exact gap:

$$F_q = -\,\mathrm{ELBO}(q), \qquad F = -\log p(x), \qquad \log p(x) - \mathrm{ELBO}(q) = D_{\mathrm{KL}}\big(q(z) \,\big\|\, p(z \mid x)\big). \tag{9.2}$$

variational free energy
negative ELBO

So $F \le F_q$ *is* $\log p(x) \ge$ ELBO: **maximizing the evidence lower bound is minimizing a variational free energy** — the same functional under a substitution of symbols, not merely an analogy. The physicist bounding $\log Z$ from below and the machine-learner bounding the log-evidence from below are running one optimization; Weiss's molecular field and the inference engine inside every VAE are one method met at two ends of a century. Here is the dictionary in full — the dictionary for the next three weeks:

Table 9.1: The physics inference dictionary: one functional, two vocabularies.

statistical physics	variational inference
configuration x	latent variable z (data x fixed)
energy $E(x)$	$-\log p(x, z)$
Boltzmann distribution $p = e^{-\beta E}/Z$	posterior $p(z \mid x)$
partition function Z	evidence $p(x)$
free energy $F = -T \log Z$	negative log-evidence $-\log p(x)$
trial distribution $q(x)$	variational posterior $q(z)$
variational free energy F_q	$-\,\mathrm{ELBO}(q)$
gap $T\, D_{\mathrm{KL}}(q \Vert p)$	$\log p(x) - \mathrm{ELBO}$
minimize the free energy	variational inference

9.5 The engine, part III: mean field, the factorized ansatz

The bound holds for *any* q; it becomes a method the moment we commit to a family. Take the simplest tractable one: **complete independence**,

$$q(s) \;=\; \prod_{i=1}^{N} q_i(s_i), \qquad q_i(s_i) = \frac{1 + m_i s_i}{2},$$

for Ising variables $s_i = \pm 1$, parameterized by the per-spin magnetizations $m_i = \langle s_i \rangle_q \in [-1, 1]$ — these are the variational parameters we will optimize. (A hygiene note: this m_i is a *variational* per-site magnetization, a knob we turn; Week 2's overlap m was an order parameter of the true dynamics. Same letter, same physical flavor, different formal role.)

Evaluate the two halves of F_q for the Ising energy $E = -\frac{1}{2}\sum_{i \neq j} J_{ij} s_i s_j - \sum_i h_i s_i$. The energy term is where the approximation acts: under a factorized q, expectations of products split,

$$\langle s_i s_j \rangle_q = m_i m_j \quad (i \neq j) \qquad \Longrightarrow \qquad \langle E \rangle_q \;=\; -\tfrac{1}{2} \sum_{i \neq j} J_{ij}\, m_i m_j \;-\; \sum_i h_i m_i .$$

Read the first equality as the approximation *stated as an equation*: the factorization sets every connected correlation $\langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$ to zero — identically, not merely approximately. That is the entire price of tractability, localized in one line. The entropy term, by contrast, is exact for the family and splits by independence into a sum of binary entropies,

$$S[q] \;=\; \sum_i s(m_i), \qquad s(m) = -\frac{1+m}{2} \log \frac{1+m}{2} - \frac{1-m}{2} \log \frac{1-m}{2}.$$

Now minimize $F_q = \langle E \rangle_q - T \sum_i s(m_i)$ over each m_i. The energy contributes $\partial \langle E \rangle_q / \partial m_i = -(\sum_j J_{ij} m_j + h_i) \equiv -h_i^{\text{eff}}$ (the $\frac{1}{2}$ canceling the ij/ji double count, the same bookkeeping as Week 3's gradient); the entropy contributes $\partial s(m_i)/\partial m_i = -\operatorname{artanh}(m_i)$ (differentiate the binary entropy and the logs collapse). Stationarity, $-h_i^{\text{eff}} + T \operatorname{artanh}(m_i) = 0$, then inverts to

$$\boxed{\; m_i \;=\; \tanh\Big(\beta\Big(\sum_j J_{ij}\, m_j + h_i\Big)\Big) \;} \tag{9.3}$$

— the **mean-field equations**, and the cartoon of Figure 9.1 now derived rather than drawn. Each spin responds — with Week 1's exact two-state Boltzmann average, the tanh — to the *average* field of its neighbors rather than their fluctuating states. And the equations are *self-consistent*: the magnetizations determine the effective fields that determine the magnetizations, a loop closed on itself. That loop is where all the physics of the next lecture lives — a closed loop can sustain a solution with no external cause, which is what spontaneous magnetization *is*.

mean-field
self-consistent equations

The equation admits two readings. *Physics reading:* this is Weiss's molecular-field equation of 1907 (Weiss 1907) — but where Weiss *posited* the average field, we

have *derived* it as the optimum of a variational bound, so we know exactly what was discarded (the correlations) and in which direction the answer errs ($F_q \geq F$, always). *Inference reading:* iterate the equations — sweep the sites, update each $m_i \leftarrow \tanh(\beta h_i^{\text{eff}})$ from the current neighbors — and you are running **naive mean-field variational inference**, coordinate ascent on the ELBO, each update provably lowering F_q (the monotonicity is one line from the bound: each m_i-update is an exact minimization of F_q in that coordinate with the others frozen; the general-family version of the update, $q_i \propto \exp\langle \log p \rangle_{q_{\setminus i}}$, is the CAVI algorithm of the ML textbooks (Blei, Kucukelbir, and McAuliffe 2017)). The 1907 self-consistency and the 2017 inference algorithm are the same fixed-point iteration.

When should the lie be a good one? The cartoon answers: mean field is the exact theory of a system whose spins do not fluctuate about their means, so its quality is set by the *sharpness of the average field*. A spin with z neighbors feels an effective field that is a sum of z terms; its relative fluctuation scales as $1/\sqrt{z}$. High connectivity — high dimension, dense graphs, the fully-connected models of Weeks 2–4, and (not coincidentally) the wide dense layers of modern machine learning — is mean field's home turf. Low dimension and the vicinity of a critical point, where fluctuations are the whole story, are where it fails — instructively, as the next lecture will show. And one more standing warning: the self-consistent equations may have *several* solutions (that multiplicity is exactly how the phase transition will appear); the variational principle adjudicates — take the solution with the lowest F_q.

> Take-home 2
>
> The factorized ansatz $q = \prod_i q_i$ turns the bound into mean-field theory: F_q is minimized at $m_i = \tanh\left(\beta(\sum_j J_{ij} m_j + h_i)\right)$ — each spin in the *average* field of its neighbors. The price is exact and structural: all connected correlations are set to zero. Iterating the equations is naive mean-field variational inference, monotonically descending F_q; the approximation is best at high connectivity ($1/\sqrt{z}$ field fluctuations) and worst near criticality.

9.6 Example: the bound in action

One small model where the truth is computable, so that for once the bound, the gap, and the discarded correlations are all *visible*. Take $N = 6$ spins, fully connected with ferromagnetic couplings $J_{ij} = J_0/N$ ($J_0 = 1$ — the model the next lecture pushes to its $N \to \infty$ limit), plus a small symmetry-breaking field $h = 0.05$. With $2^6 = 64$ states, exact enumeration gives Z, F, and every moment; solving Equation 9.3 by iteration gives the mean-field m_i and F_q. The comparison, across temperature:

T	exact F	mean-field F_q	gap $= T\,D_{\mathrm{KL}}(q\|p)$	exact $\langle s_i \rangle$	mean-field m_i
3.0	−12.561	−12.480	0.081	0.023	0.023
1.0	−4.514	−4.201	0.313	0.131	0.262
0.7	−3.519	−3.214	0.305	0.263	0.742
0.3	−2.844	−2.805	0.039	0.754	0.994

Read the table by columns, because each column checks one claim of the lecture. The **gap column** is never negative — the bound Equation 9.1, made numerical

(and computed two independent ways in the code below: as $F_q - F$, and directly as $T\sum_s q\log(q/p)$ over all 64 states; they agree to machine precision, which is the identity Equation 9.1 verified rather than trusted). The gap is small at high temperature, where spins are nearly independent and the factorized family nearly contains the truth; small again at very low temperature, where the system freezes into an essentially factorized ordered state; and largest in between — peaking at 0.35 near $T \approx 0.85$, a tenth of the free energy — precisely where correlated fluctuations are strongest. The **magnetization columns** show the reverse-KL signature we promised: at $T = 0.7$ the exact magnetization is a modest 0.26, but mean field reports 0.74 — the factorized q, unable to spend probability on correlated fluctuations, buys its free energy by *overcommitting to order*, the mode-seeking overconfidence of Figure 9.3 realized in spins. Figure 9.4 draws both columns as curves, and marks the temperature $T_c^{\mathrm{MF}} = (N-1)J_0/N \approx 0.83$ where the next lecture will have much to say: mean field manufactures a sharp onset of order there, while the exact six-spin system — which, being finite, has no phase transition at all — crosses over smoothly. The gap peaks in exactly that neighborhood. *Where the approximation manufactures a sharp transition is where it is least trustworthy* — a diagnosis the next lecture turns into theory, and the tutorial into a projector experiment.

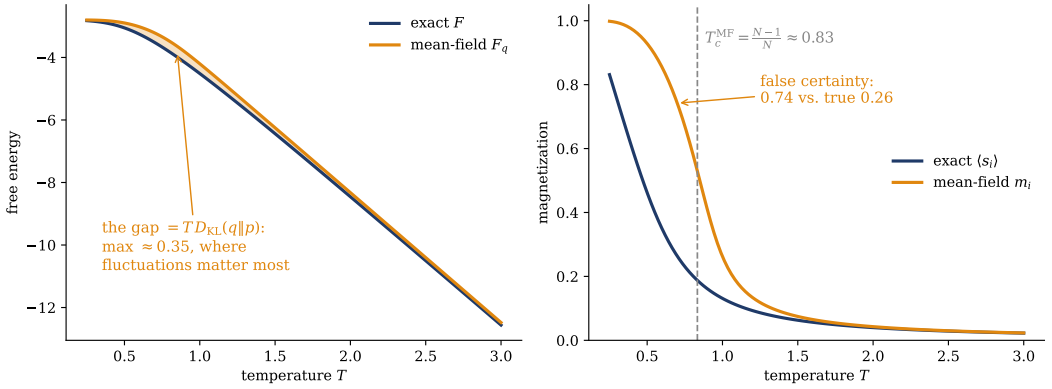

Figure 9.4: The bound in action: exact versus mean-field thermodynamics of a fully-connected $N = 6$ ferromagnet ($J_{ij} = 1/N$, $h = 0.05$), exact values from enumeration of all 64 states. Left: the true free energy F (navy) and the variational F_q (orange) — F_q never dips below F, and the shaded gap, equal to $T\,D_{\mathrm{KL}}(q\|p)$ by Equation 9.1 (verified in the code by direct enumeration of the KL), peaks at ≈ 0.35 near the mean-field critical temperature and closes at both ends. Right: the magnetizations. Mean field (orange) switches on sharply near $T_c^{\mathrm{MF}} = (N-1)/N \approx 0.83$ (dashed) — a phase transition it has manufactured — while the exact finite system (navy) crosses over smoothly; at $T = 0.7$ the approximation reports $m = 0.74$ against a true 0.26. Overconfident order is the reverse-KL signature: the factorized family cannot afford fluctuations, so it buys free energy with false certainty.

> *The mean-field free energy is never below the truth — that is the guarantee: we are always wrong in one known direction, by exactly a KL divergence.*

The next lecture takes Equation 9.3 to the ferromagnet proper — uniform couplings, $h = 0$, $N \to \infty$ — where a nonzero magnetization appears spontaneously below a critical temperature: **a phase transition, falling out of a variational approximation**, with an order parameter and a critical exponent. The tutorial then puts

mean field, Monte Carlo sampling, and exact enumeration side by side on one small lattice, testing where the three agree and where the bound opens a gap.

Take-home 3

On a brute-forceable model the guarantee is visible: $F_q \geq F$ at every temperature, by exactly $T\, D_{\mathrm{KL}}(q\|p)$, with the gap largest where correlated fluctuations are strongest — near the transition mean field itself manufactures — and the mean-field magnetizations overconfident there (reverse-KL mode-seeking, in spins). Exact at weak coupling; wrong, in one known direction, elsewhere.

9.7 Outlook: a phase transition from an approximation

Confronted with an intractable Z, we traded the true distribution for a tractable approximation — chosen by minimizing a one-sided bound whose gap is a KL divergence — and found that the physicist's version of this trade (Gibbs–Bogoliubov, Weiss's molecular field) and the machine-learner's (the ELBO, variational inference) are one functional under two names. The dictionary of Table 9.1 applies throughout Module 2.

The next lecture applies the method to the uniform ferromagnet: the self-consistency $m = \tanh(\beta z J m)$ *breaks symmetry* — below $T_c = zJ$ the disordered solution destabilizes and the system magnetizes spontaneously, with the square-root onset $m \sim (T_c - T)^{1/2}$ — a genuine collective phase transition, predicted by an approximation that ignores every correlation. But the same approximation overestimates T_c (it ignores the fluctuations that fight order), predicts a transition even in one dimension where none exists, and gets the critical exponents wrong — because at criticality fluctuations are not a correction but the dominant effect. Restoring the discarded correlations systematically is Week 6's agenda: the TAP correction (Thouless, Anderson & Palmer 1977, who corrected the very Sherrington–Kirkpatrick model Week 2 leaned on) and belief propagation, which is exact on trees. The machine-learning arrow runs parallel: naive mean-field VI is the workhorse of scalable Bayesian inference, and its mode-seeking bias becomes Week 7's posterior collapse inside the variational autoencoder.

10 The mean-field ferromagnet: order from an approximation

Recommended reading

Core (~1 h). Mézard and Montanari (2009), Ch. 2 — the Curie–Weiss model and the mean-field transition, in the notation Week 6 will extend to message passing; this was the warm-up reading, and today's engine follows its logic. Alongside it, Sethna (2021), the phase-transitions and order-parameters sections — the Landau picture, symmetry breaking, and critical exponents, cleanly.
Optional background. Mehta et al. (2019), the Ising/mean-field sections, for the course's notation and the ML framing; Goldenfeld (1992), Ch. 5, for mean-field exponents, the Ginzburg criterion, and *why* mean field fails below four dimensions — the renormalization-group story this course only points at; Onsager (1944) — the exact two-dimensional solution, named today as the decisive evidence against mean-field exponents, not read.
Prerequisite reminder. All of Chapter 9 — the variational bound and its ELBO reading, the mean-field equations Equation 9.3, and the reverse-KL overconfidence — plus the Ising model of Week 2 and the free-energy saddle of Section 2.5, which today stops being an exercise and becomes a theorem about mean field. New content: solving the self-consistency, the phase transition and the Landau free energy, spontaneous symmetry breaking, mean-field critical exponents, and an audit of where mean field can be trusted.

10.1 The ferromagnet

The previous lecture built the variational machinery; today we apply it to the ferromagnet. The central equation: minimize the variational free energy over factorized distributions and each spin settles into the average field of its neighbors, $m_i = \tanh\left(\beta(\sum_j J_{ij} m_j + h_i)\right)$ — an equation that is simultaneously Weiss's molecular field and coordinate-ascent variational inference. (The equation always admits $m = 0$ — the question is when it also admits nonzero solutions.)

Applied to the simplest interacting system — a ferromagnet, every spin nudging its neighbors toward alignment — **a phase transition falls out**. Below a sharp critical temperature, a second solution appears, the symmetric one destabilizes, and the system magnetizes *spontaneously*: collective order, predicted by an approximation that treats every spin as independent. Historically, this is where the subject's language was established: **Curie (1895)** measured the ferromagnetic transition and the susceptibility law that bears his name; Weiss (1907) explained both with the molecular field — the previous lecture's equation, applied and solved; and Landau (1937) abstracted the phenomenon into the universal vocabulary — *order parameter*,

symmetry breaking, a free energy expanded in powers of the order — that physics now applies to every transition it meets, including, this course keeps finding, the ones inside learning machines (Week 2's memory collapse already spoke this language; today we learn it properly).

The same equation, applied to a one-dimensional chain, predicts a transition at $T_c = 2J$ — **and there is no transition in one dimension at all.** The true 1D Ising model orders only at $T = 0$; the mean-field prediction is qualitatively wrong — a clean calculation that produces order where none exists. The lecture therefore has two halves: the transition itself, and the audit of when to trust the approximation that predicts it.

> Take-home 1
>
> Applied to the ferromagnet, the mean-field equation produces a phase transition: below T_c the system spontaneously magnetizes — collective order from an approximation. But the same approximation *misplaces* the order it predicts: wrong T_c, spurious transitions in low dimension. The lecture presents the transition and then audits its validity.

10.2 The transition

Setup. Make the system homogeneous so one number carries everything: every site couples ferromagnetically to its neighbors, and by symmetry every site has the same magnetization, $m_i = m$. Write $J_0 \equiv \sum_j J_{ij}$ for the *total coupling a site feels*: for a lattice of coordination z with bonds J, $J_0 = zJ$; for the fully-connected normalization $J_{ij} = J/N$ of the previous lecture's example, $J_0 = J(N-1)/N \to J$. (A symbol caution: the previous lecture used J_0 for the coupling *amplitude* in $J_{ij} = J_0/N$; here it is the summed field $\sum_j J_{ij}$ — the two agree as $N \to \infty$ but differ by $(N-1)/N$ at finite N.) At zero external field, the previous lecture's equations collapse to a single scalar self-consistency:

$$m \;=\; \tanh(\beta J_0\, m).$$

We now face that question head-on: $m = 0$ solves this at every temperature. When does anything *else*?

The graphical solution. Plot both sides (Figure 10.1): the diagonal $y = m$ and the curve $y = \tanh(\beta J_0 m)$. Both pass through the origin, so everything hinges on which is steeper there — the tanh leaves the origin with slope βJ_0. If $\beta J_0 \le 1$, the tanh starts *below* the diagonal and stays below: the origin is the only intersection, and the system is disordered. If $\beta J_0 > 1$, the tanh starts *steeper* than the diagonal, rises above it, and — being bounded — must bend back and recross it: two new intersections $\pm m^*$ appear, symmetric about the origin. The threshold is exact and immediate:

$$\beta_c\, J_0 = 1, \qquad \text{i.e.} \qquad T_c = J_0 \tag{10.1}$$

mean-field critical temperature

Physically: *order sets in when the thermal energy drops below the total coupling a spin feels — when alignment beats agitation.* For coordination z this reads $T_c = zJ$:

more neighbors, higher transition temperature — a first quiet hint of the validity story to come, since "more neighbors" is also what makes the mean field a better approximation.

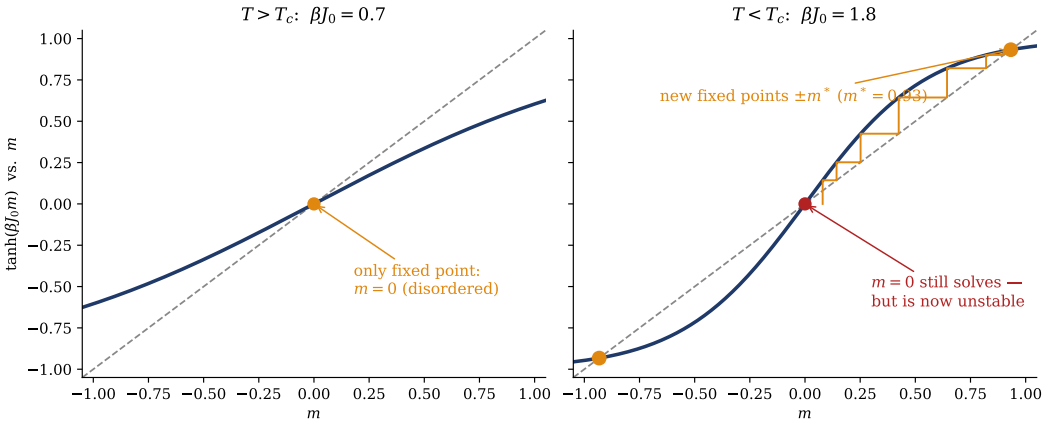

Figure 10.1: The graphical solution of $m = \tanh(\beta J_0 m)$, the engine's first picture. Left, $T > T_c$ (drawn at $\beta J_0 = 0.7$): the tanh leaves the origin with slope $\beta J_0 < 1$, stays below the diagonal, and the only fixed point is $m = 0$ — disorder. Right, $T < T_c$ ($\beta J_0 = 1.8$): the origin slope exceeds one, the curve rises above the diagonal and bends back, and two new fixed points $\pm m^*$ appear (here $m^* = 0.93$); the cobweb shows the previous lecture's coordinate-ascent iteration $m \leftarrow \tanh(\beta J_0 m)$ flowing away from the now-unstable origin and into the ordered solution. The transition is the moment the origin slope crosses one: $T_c = J_0$.

Below T_c, then, three solutions coexist: 0 and $\pm m^*$. Which does the system take? The cobweb in the figure already suggests the answer — the previous lecture's fixed-point iteration flows *away* from the origin and *into* $\pm m^*$ — but "the iteration goes there" is dynamics, not thermodynamics. The previous lecture's variational principle provides the answer: among multiple self-consistent solutions, take the one of lowest variational free energy. So we must now look at F_q itself — and looking at it is the second half of the engine, because the transition turns out to be a change in the *shape of a landscape*.

10.3 Landau's landscape

The free energy as a function of the order parameter. For the homogeneous ansatz, the previous lecture's variational free energy per spin is a function of the single number m:

$$f(m) \;=\; \frac{\langle E\rangle_q}{N} - T\,s(m) \;=\; -\tfrac{1}{2}J_0\,m^2 \;-\; T\,s(m),$$

with the binary entropy $s(m)$ of Section 9.5 (the $\frac{1}{2}$ is the shared-bond bookkeeping, as in the previous lecture: $\langle E\rangle_q = -\frac{1}{2}\sum_{i\neq j} J_{ij}m^2 = -\frac{N}{2}J_0 m^2$). You have met this exact function before, in a different context: it is the $f(m) = \varepsilon(m) - Ts(m)$ whose saddle Section 2.5 evaluated for the all-to-all ferromagnet, back when it was a Week 1 exercise in Laplace's method. That recognition returns below.

Expand near the transition. Everything near T_c happens at small m, so expand. The entropy's Taylor series follows from $\partial S/\partial m = -\operatorname{artanh}(m) = -(m + m^3/3 + \cdots)$:

$$s(m) \;=\; \log 2 \;-\; \frac{m^2}{2} \;-\; \frac{m^4}{12} \;-\; \cdots$$

and assembling f:

$$\boxed{f(m) \;\approx\; -T\log 2 \;+\; \frac{T - T_c}{2}\, m^2 \;+\; \frac{T}{12}\, m^4 \;+\; \cdots} \tag{10.2}$$

Landau free energy

This is the **Landau free energy**, and its entire content sits in the quadratic coefficient $a(T) = (T - T_c)/2$, which **changes sign at** T_c — the same $T_c = J_0$ as the graphical argument, reassuringly. For $T > T_c$: a single well at $m = 0$; disorder is the minimum. For $T < T_c$: the origin curves *downward* — the symmetric solution still exists but is now a local *maximum* — and the positive quartic term ($b = T/12 > 0$) catches the landscape on both sides, creating two symmetric minima at $\pm m^*$. Figure 10.2 draws the morphing landscape and the *pitchfork* of solutions it implies. The transition is a change in the shape of a landscape, with temperature as the control parameter — the same picture as Figure 3.1, now derived from a variational principle.

Spontaneous symmetry breaking. Look at what has happened to the symmetry. The Hamiltonian at $h = 0$ is exactly invariant under the global flip $s \to -s$, so $f(m) = f(-m)$: the *law* is Z_2-symmetric at every temperature. But below T_c the *state* is not: the system sits in one well, $+m^*$ or $-m^*$, and which one is decided by an infinitesimal stray field or the memory of history. The symmetry of the law is not the symmetry of the solution — *spontaneous symmetry breaking*, the concept Landau (1937) built the modern theory of phases around. One caveat, kept precise in the script: at finite N the system tunnels between the two wells and the literal average $\langle m \rangle$ vanishes; broken symmetry is a statement about noncommuting limits ($h \to 0$ *after* $N \to \infty$), i.e. about *ergodicity breaking* — the barrier between the wells becomes extensive and the two ordered states become separate worlds. Week 2's stored memories were exactly such separate worlds; the vocabulary is the same because the phenomenon is.

The critical exponent. How fast does order grow below the transition? Minimize Equation 10.2: $0 = (T - T_c)\, m + \frac{T}{3} m^3$ gives

$$m^{*2} = \frac{3\,(T_c - T)}{T} \qquad \Longrightarrow \qquad \boxed{m^* \;\sim\; (T_c - T)^{1/2}} \tag{10.3}$$

order-parameter onset, exponent $\beta_{\text{exp}} = \frac{1}{2}$

— the square-root onset: steep (infinite slope at T_c), continuous (no jump), universal. (The same result falls out of the self-consistency directly — expand $\tanh x \approx x - x^3/3$ in $m = \tanh(\beta J_0 m)$ and solve; the script runs both routes and they agree, as they must, since the equation *is* the stationarity condition of the landscape.) A notation flag, once and firmly: the exponent is conventionally called β, which this course cannot write without a collision — we write β_{exp} for the exponent, and β remains $1/T$. Its companions, derived in the script and stated here: the susceptibility diverges as $\chi = \partial m/\partial h \sim |T - T_c|^{-1}$ (exponent $\gamma = 1$ — this is the **Curie–Weiss law**: Curie measured the paramagnetic $\chi \sim 1/T$, and Weiss's molecular field bent it into the

$1/(T-T_c)$ that diverges at the transition); on the critical isotherm $m \sim h^{1/3}$ ($\delta = 3$); and the specific heat has a finite jump ($\alpha = 0$). These are the *mean-field critical exponents*, and their most remarkable property is what they do not depend on: neither the lattice, nor the material, nor the microscopic details — every system, treated in this approximation, shares them. That is the first sighting of *universality*, and it survives (in corrected form) far beyond mean field.

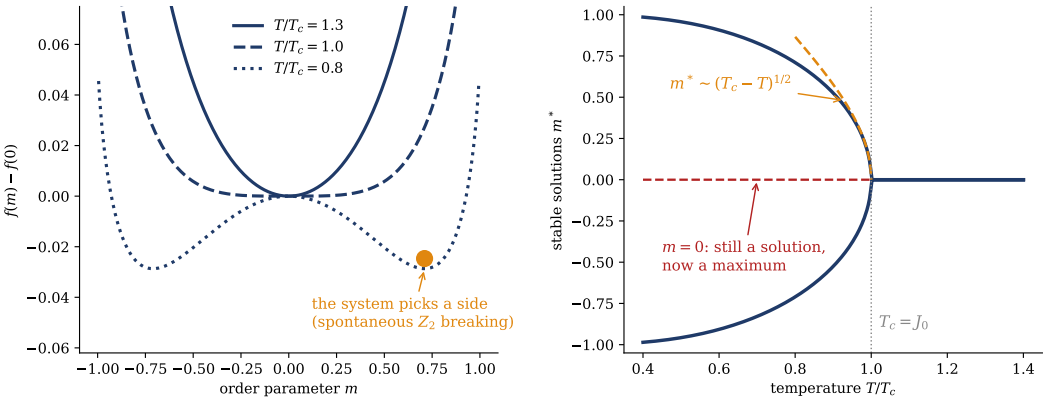

Figure 10.2: The transition as a landscape, exactly (not the truncated expansion): the variational free energy per spin $f(m) = -\frac{1}{2}J_0 m^2 - T\,s(m)$, drawn relative to $f(0)$, at three temperatures ($J_0 = 1$). Left: above T_c a single well at $m = 0$; at $T = T_c$ the well bottom flattens into a quartic; below T_c the origin becomes a local maximum and two symmetric minima appear at $\pm m^*$ — the system must choose a side (ball drawn at $+m^* = 0.71$ for $T = 0.8$), though the law remains exactly $m \to -m$ symmetric: spontaneous Z_2 symmetry breaking. Right: the resulting pitchfork — the stable solutions (navy) against temperature, with the square-root onset $m^* \sim (T_c - T)^{1/2}$ of Equation 10.3 (orange, asymptotic) and the destabilized $m = 0$ branch (red dashed). Compare Section 2.5, where this same double well appeared as a Week 1 exercise: this is mean-field theory, performed before it was named as such.

10.4 The limits of mean field

The previous lecture told us exactly what the approximation discarded (every connected correlation), and physics has had a century to measure what that costs.

Mean field over-predicts order. The fluctuations it discards work against order: local reversals eat away at alignment. Ignore them and you systematically *overestimate* the ordered phase. Quantitatively: on the 2D square lattice, mean field predicts $T_c = 4J$, while Onsager's exact solution (Onsager 1944) gives $T_c = 2J/\log(1+\sqrt{2}) \approx 2.27\,J$ — an overestimate of 76%. Qualitatively, and worse: on the 1D chain, mean field predicts $T_c = 2J$, and the exact answer is that **there is no transition at all** — in one dimension, thermal fluctuations destroy long-range order at any $T > 0$ (a single flipped domain costs finite energy and buys extensive entropy). The approximation does not merely misplace the transition; it invents one.

Mean field gets the exponents wrong in low dimension. Where a transition genuinely exists, the mean-field *location* is off but the *exponents* — the universal

part — are also wrong: the 2D order parameter grows as $(T_c - T)^{1/8}$ (announced by Onsager in 1949, proved by Yang in 1952), not our square root. The resolution, stated and pointed at rather than derived (the Ginzburg criterion; Goldenfeld (1992)): fluctuations dominate the neighborhood of a critical point in low dimension, and only above the *upper critical dimension* $d_c = 4$ do they become negligible enough for the mean-field exponents to be exact. Below it, computing them correctly is the renormalization group's job — a course this course respectfully is not.

And where mean field is *exact*. For the **fully-connected model** — $J_{ij} = J/N$, every spin coupled to all others — mean field is not an approximation at all in the thermodynamic limit. There are two ways to see it, each worth one sentence. The physical one: the effective field on a spin is an average over $N-1$ neighbors, and its relative fluctuation vanishes as $1/\sqrt{N}$ — in the thermodynamic limit the average field coincides with the environment each spin actually feels. The structural one, which closes the Week 1 callback: for this model the exact free energy collapses, by Laplace's method, onto $\min_m[-\frac{1}{2}Jm^2 - Ts(m)]$ — the calculation of Section 2.5 — and that is *literally the same function* our variational principle minimizes. The exact saddle and the variational optimum coincide; the bound is tight; Week 1's toy calculation was mean-field theory, performed before we knew its name. The practical corollary for this course: mean field is the theory of the sharp average field, best exactly where each unit sees many others — which is why a "crude" approximation is routinely excellent for the large, dense models of machine learning, and worst on chains, thin lattices, and anything critical.

The error, quantified. All of this is the previous lecture's gap: $F_q - F = T\,D_{\mathrm{KL}}(q\|p)$, the discarded correlations, large exactly where fluctuations dominate (low dimension, the vicinity of T_c) and negligible where the field is sharp (high connectivity, away from criticality). The error cannot be fixed by iterating harder — the factorized family simply does not contain distributions with correlations, and no amount of optimization within the family restores what the family lacks. (One more echo from the previous lecture, left for the tutorial to develop: below T_c the variational problem has *multiple optima* — the two wells — and which one an iteration finds depends on initialization. The physicist calls that symmetry breaking; the machine learner calls it the non-convexity of the ELBO; the tutorial puts both vocabularies on the same landscape.)

Trap

A clean calculation can be confidently, qualitatively wrong. Mean field overestimates T_c always (it ignores exactly the fluctuations that destroy order), and in 1D it predicts a transition where none exists — not an error of a few percent but an invented phenomenon. The lesson generalizes well beyond spins: when an approximation *discards a mechanism* (here, correlated fluctuations), no internal consistency check will reveal the damage — the equations remain internally consistent and the physics remains absent. Validate against the regime, not the algebra.

Take-home 2

Mean field over-predicts order: T_c overestimated ($4J$ vs. Onsager's $2.27J$ in 2D), a spurious transition in 1D, and wrong exponents below $d_c = 4$ — because it discards the fluctuations that fight order. It is *exact* for the fully-connected

model, where the effective field's fluctuations vanish as $1/\sqrt{N}$ and the variational optimum coincides with the exact saddle of Section 2.5. Trust it dense and away from criticality; distrust it low-dimensional and critical.

10.5 Example: the transition mean field gets wrong

We run the audit rather than assert it — one reassuring case and one damning one, side by side in Figure 10.3.

The reassuring case: the fully-connected model, where the exact answer is computable at real size. The trick is that the energy depends on the configuration only through the total magnetization $M = \sum_i s_i$, so the 2^N-term partition function collapses onto $N+1$ magnetization sectors weighted by binomial degeneracies — Week 1's "sum over states becomes a sum over macrostates" move, executed exactly. At $N = 1000$ (2^{1000} configurations, 1001 sectors) the exact $\langle|m|\rangle$ and the mean-field prediction lie on top of each other through the entire ordered phase — at $T = 0.5\,J$ they agree to three decimal places (0.957) — with the only discrepancy a finite-size rounding of width $\mathcal{O}(N^{-1/2})$ near and above T_c, where the exact finite system's $\langle|m|\rangle$ cannot quite vanish. This is mean field's home turf, verified.

The damning case: the 1D chain, $z = 2$. Mean field confidently magnetizes at $T_c^{\mathrm{MF}} = 2J$. The exact chain (transfer matrix, or Week 1's two-state machinery applied bond by bond) has $m = 0$ at every positive temperature — and, drawn alongside, the quantity mean field is constitutionally blind to: the exact nearest-neighbor correlation $\langle s_i s_{i+1}\rangle = \tanh(\beta J)$, growing smoothly toward 1 as the chain cools. *Local* order builds continuously; *long-range* order never arrives; and mean field — which cannot tell the two apart, having set all correlations to zero — mistakes the first for the second. The discarded quantity is not a refinement; in one dimension it is the whole story.

model	T_c^{MF}	T_c^{exact}	verdict
1D chain ($z = 2$)	$2J$	0 — no transition	invented order
2D square lattice ($z = 4$)	$4J$	$2.27\,J$ (Onsager 1944); $\beta_{\mathrm{exp}} = \frac{1}{8}$, not $\frac{1}{2}$	wrong place, wrong exponent
fully connected ($J_{ij} = J/N$)	J	J	**exact** (as $N \to \infty$)

Mean field predicts order from an approximation — and misplaces it; the discrepancy identifies where next week's corrections are needed.

The tutorial (16–18, Ph12 106) is computation-led: three treatments of one small Ising lattice, side by side: **exact enumeration** (the truth, affordable only because the lattice is small — Week 3's ledger set the ceiling), **mean field** (cheap, deterministic, correlations identically zero), and **Monte Carlo sampling** (the other road out of an intractable Z, introduced here two weeks before Module 3 makes it a subject). Today's lecture provides the complete theory of where the three agree and where the mean-field gap opens. The comparison is run in the room by the groups,

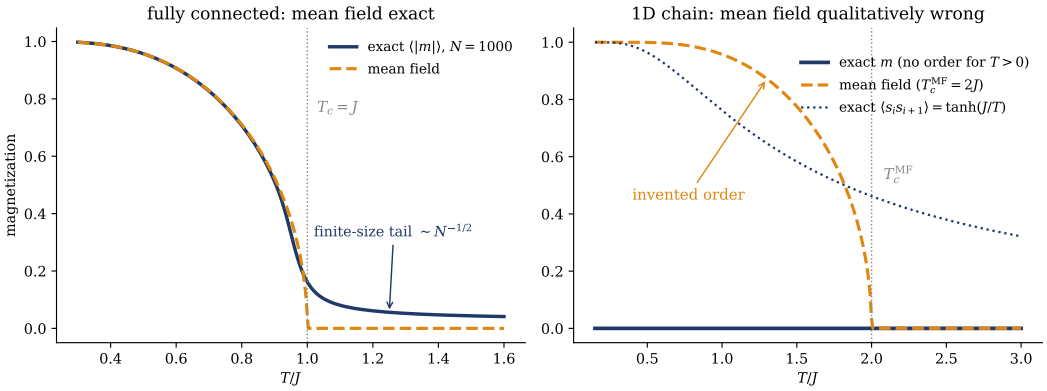

Figure 10.3: The audit, run. Left: the fully-connected model at $N = 1000$ — the exact $\langle|m|\rangle$, computed from the 1001 magnetization sectors that the 2^{1000}-state sum collapses onto (navy), against the mean-field prediction (orange). The curves coincide through the ordered phase (both 0.957 at $T = 0.5$); the small exact tail above T_c is finite-size rounding of order $N^{-1/2}$, vanishing as $N \to \infty$. Mean field is exact here. Right: the 1D chain — mean field magnetizes at $T_c^{\mathrm{MF}} = 2J$ (orange); the exact chain never does ($m = 0$ for all $T > 0$, navy). The dashed curve is what mean field cannot see: the exact nearest-neighbor correlation $\langle s_i s_{i+1} \rangle = \tanh(J/T)$, building smoothly while long-range order never arrives. Mistaking local correlation for global order is precisely the error of setting correlations to zero.

on their own machines, once each group has written down where it expects the gap to appear.

> **Take-home 3**
>
> Run, the audit confirms the theory: on the fully-connected model mean field coincides with the exact sector-sum answer through the ordered phase; on the 1D chain it magnetizes at $2J$ while the truth stays disordered forever, mistaking smoothly growing *local* correlation ($\tanh \beta J$ — the quantity it set to zero) for long-range order. The gap is the discarded correlations, largest where fluctuations dominate — and *where mean field fails is where variational inference will need corrections.*

10.6 Outlook: corrections, and the other road

Module 2's first method now stands complete. A variational approximation, minimizing a bound, predicted collective order, located it at $T_c = J_0$, gave the transition a universal shape, and founded the language — order parameter, symmetry breaking, Landau landscape — in which this course discusses every collective phenomenon it meets, from Week 2's memory collapse (a *first-order* cousin: the order parameter jumped, where today's grew continuously) to the transitions that lie ahead in sampling dynamics and training. But the method's one structural act — setting correlations to zero — over-predicts order everywhere, catastrophically so in low dimension, and no optimization within the factorized family can repair it.

Week 6 addresses this directly. If the disease is discarded correlations, the cure is to *put them back, systematically.* Two restorations, named now and derived next week. The **TAP equations** — Thouless, Anderson, and Palmer (1977), the same lineage as Week 2's spin-glass machinery — add to the mean-field equation the leading fluctuation correction, the *Onsager reaction term*: the part of a spin's local field that is merely its own influence reflected back by its neighbors, which honest bookkeeping must subtract. And **belief propagation** — message passing on the interaction graph — generalizes the variational family from independent sites to consistent pairs, is *exact on trees*, and comes with its own free energy (Bethe's) standing where Gibbs–Bogoliubov stood today. Both are still road one: approximate the distribution, bound or estimate the free energy, restore correlations order by order.

The tutorial's third method — Monte Carlo — does not approximate p at all; it *samples* it, trading determinism and speed for asymptotic exactness and avoiding Z entirely. That road is Module 3's subject, three weeks from now; today's comparison is an introduction. The course's three strategies for the partition function remain: bound the free energy (Module 2), sample the distribution (Module 3), and eliminate the normalizer from the gradient (Week 10).

11 The TAP equations: the term mean field forgot

Recommended reading

Core (~1 h). Thouless, Anderson, and Palmer (1977) — six pages, and the best-titled paper in this syllabus: the corrected mean-field theory of the Sherrington–Kirkpatrick model, containing the reaction term this lecture derives; read for the equations and for the quotation marks in the title. Alongside it, re-read Mézard and Montanari (2009), Ch. 2, whose cavity formulation our physical route follows. (The belief-propagation chapter, ch. 14, is *the next lecture's* warm-up pre-read — the keen may start it early, but stop before the messages become algorithms.)
Optional background. Plefka (1982) — the systematic expansion behind the engine, naive mean field and TAP as its first two orders; Georges and Yedidia (1991) — the expansion continued, every higher correction generated mechanically; Onsager (1936) — the original *reaction field*, in dielectrics, sixty years before machine learning rediscovered it; Mézard, Parisi, and Virasoro (1987) — the cavity method and the many-states picture, for depth; Opper and Saad (2001) for TAP in inference.
Prerequisite reminder. All of Week 5 — the variational bound and its ELBO reading, the factorized ansatz and Equation 9.3, the audit and its verdict — plus Week 2's disordered couplings (Hebb, the AGS feedback factor r) and one fact from Week 1's two-state spin that we will re-derive in a line: its susceptibility. New content: the reaction/cavity argument, the Plefka expansion, the TAP equations, the $\sum_j J_{ij}^2$ scaling analysis, and the estimate-versus-bound trade.

11.1 The debt, and a cheeky title

Last week's audit ended in a verdict — mean field discards every correlation, overpredicts order, and *where it fails is where variational inference needs corrections* — and the tutorial made the gap visible on a projector. Today we begin the correction. Naive mean field will turn out to be the *first order of a controlled expansion*, its leading correction is a single term with a crisp physical meaning, and the size of that term will tell us — quantitatively, model by model — when Week 5's theory was trustworthy and when it was quietly wrong.

The history behind that term frames the whole of Module 2, and it runs through Week 2's own lineage. In 1975, Sherrington and Kirkpatrick proposed a mean-field spin glass — Gaussian random couplings, every spin coupled to every other — and, with disarming confidence, titled the paper *Solvable Model of a Spin-Glass* (Sherrington and Kirkpatrick 1975). They then solved it, by the replica method, and their

solution carried a defect they themselves flagged: at low temperature the entropy goes **negative** — for discrete spins, an impossibility, the thermodynamic equivalent of a proof ending in $1 = 2$. Two years later, Thouless, Anderson, and Palmer published the corrected mean-field theory under the title *Solution of "Solvable Model of a Spin Glass"* (Thouless, Anderson, and Palmer 1977) — the quotation marks serving as a referee report. Their correction to the mean-field equations is *one term*. Today we earn it, twice: once by a short physical argument, and once by an expansion that tells us exactly what order of what series the argument computed. (The SK saga's full resolution — Parisi's replica symmetry breaking, and a 2021 Nobel prize — is a mountain this course walks beside rather than over; the outlook says precisely how our road relates to it.)

The driving question, point first: *what is the first correction to mean field, and when is it large?* We preview both answers. The correction is **subtracting your own echo** — removing from each spin's effective field the part that the spin itself put there, reflected back by its neighbors. And its size is governed by $\sum_j J_{ij}^2$: of order $1/N$ for the uniform ferromagnet Week 5 audited — invisible, which is *why* naive mean field passed that audit — but of order *one* for disordered couplings, which is to say for the SK model, for Week 2's Hebb-built networks, and for essentially every model whose couplings were written by data rather than by symmetry.

> Take-home 1
>
> Naive mean field is not a dead end but the *first order* of a systematic expansion. Its leading correction — the Onsager reaction term, Thouless, Anderson, and Palmer (1977) — subtracts from each spin's field the part that spin itself put there. The correction is $\mathcal{O}(1/N)$ for uniform dense couplings and $\mathcal{O}(1)$ for disordered ones: the models that matter live on the corrected side.

11.2 The echo: the field you feel versus the field you cause

The physical argument first, because it fits in a picture that carries the central idea. The naive effective field of Equation 9.3, $\sum_j J_{ij} m_j + h_i$, treats each neighbor's magnetization as *independent evidence* about what spin i should do. But look at where m_j comes from: j sits, in part, in the field of i itself. Spin i polarizes its neighbor through the bond; the polarized neighbor reports back through the same bond; and the naive field counts that report as one more vote for what i already is. A consistent theory must not let a spin respond to its own echo.

One line of Week 1 makes the echo quantitative. A single spin in a field h has $m = \tanh(\beta h)$, so its *susceptibility* — its readiness to respond — is

$$\chi_j \;=\; \frac{\partial m_j}{\partial h_j} \;=\; \beta\,(1 - m_j^2)\,:$$

single-site linear response

an undecided spin ($m_j \approx 0$) responds maximally; a saturated one ($m_j \to \pm 1$) barely responds at all. That is the only new ingredient the correction needs, and it is not new. Assemble the echo (Figure 11.1): spin i shifts its neighbor by $\delta m_j = \chi_j J_{ij} m_i$; the shifted neighbor feeds back the field $\delta h_i = J_{ij}\,\delta m_j = \beta(1 - m_j^2)\,J_{ij}^2\,m_i$. Note the *square* of the coupling — echoes travel out along the bond and back along the same

bond, so they always carry J^2. Summing over neighbors and subtracting gives the corrected field:

$$h_i^{\text{cav}} \;=\; h_i + \sum_j J_{ij}\, m_j \;-\; \beta\, m_i \sum_j J_{ij}^2 (1 - m_j^2) \tag{11.1}$$

the cavity field: i's field with i's own influence removed

— the *cavity field*, so called because it is the field at site i computed as if i sat in a cavity: the neighbors' magnetizations taken as they would be *in i's absence*. (Conceptually, the correct inputs are the cavity magnetizations $m_j^{(i)}$ — j's value with i removed — and the reaction term is exactly the bookkeeping that converts the full m_j into them, to the order computed: $m_j = m_j^{(i)} + \chi_j J_{ij} m_i + \mathcal{O}(J^3)$. That distinction matters loosely today and exactly in the next lecture, because keeping "who was removed" straight is precisely what message passing does.) The idea is much older than spin glasses: Onsager (1936) introduced the *reaction field* for dielectrics — a molecular dipole polarizes the liquid around it, and the returning polarization field must not be counted when computing the dipole's own alignment (Figure 11.1, right). TAP's move was to import a forty-year-old piece of electrostatics into disordered magnetism; ours, this week, is to import it into inference.

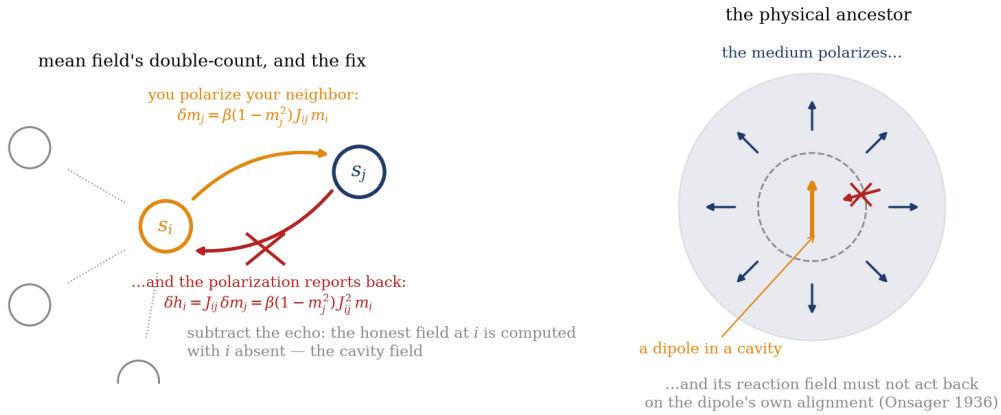

Figure 11.1: The central picture of the lecture. Left: mean field's double-count — spin i polarizes its neighbor j by $\delta m_j = \beta(1-m_j^2)J_{ij}m_i$ (orange), and the polarization reports back as a field $\propto J_{ij}^2 m_i$ (red, struck out): a spin's own influence, returning as fake evidence. Subtracting the echo yields the *cavity field* Equation 11.1 — the field at i computed with i absent. Right: the physical ancestor (Onsager 1936) — a dipole polarizes the surrounding medium, and the medium's *reaction field* must not act back on the dipole's own alignment.

Writing $m_i = \tanh(\beta h_i^{\text{cav}})$ already gives the corrected self-consistency — the TAP equations, in fact. But at this point they are physics, not yet a controlled approximation: we do not know what the argument neglected, what "order" it computed, or how to go further. The Plefka expansion answers all three.

11.3 The engine: the Plefka expansion

Setup. We want an expansion whose zeroth order is free spins, whose knob is the interaction strength, and whose variables are the magnetizations — so that each

order corrects the *theory of* m, not just the free energy. The right object is the *constrained* (Gibbs) free energy $G(\lambda, m)$: the free energy of the Hamiltonian with its exchange part scaled by an interpolation parameter $\lambda \in [0,1]$ ($J \to \lambda J$; the external fields ride along unscaled), evaluated *with the magnetizations pinned* to prescribed values m_i. The pinning is done by auxiliary Lagrange fields b_i adjusted at every λ so that $\langle s_i \rangle = m_i$ exactly — the Legendre-transform bookkeeping is spelled out in Plefka (1982) and Georges and Yedidia (1991), and the script keeps only what the board needs: at $\lambda = 0$ the constrained measure *factorizes*, so every average is computable, and derivatives of G in λ are averages of the interaction in that computable ensemble. Expand around $\lambda = 0$, then set $\lambda = 1$. (One notational flag: the literature writes this parameter α; we rename it λ because α is this course's storage load, and that collision is not worth carrying.)

Order zero — independent spins. At $\lambda = 0$, pinned free spins are Week 5's factorized ansatz exactly, and G is pure entropy: $\beta G_0 = -\sum_i S(m_i)$, the binary entropy per site, minus the field term $\beta \sum_i h_i m_i$.

Order one — Week 5, recovered. The first derivative is the average of the exchange energy in the factorized reference, where $\langle s_i s_j \rangle = m_i m_j$:

$$G_1 \;=\; \Big\langle -\sum_{(ij)} J_{ij}\, s_i s_j \Big\rangle_0 \;=\; -\sum_{(ij)} J_{ij}\, m_i m_j .$$

Assemble $G_0 + \lambda G_1$ at $\lambda = 1$ and you are holding, symbol for symbol, Week 5's mean-field free energy. This reframes the whole of last week: **naive mean field is the first-order truncation of this expansion.** It was never an isolated ansatz; it was rung one of a ladder (Figure 11.2), and the ladder continues.

Order two — the Onsager term, derived. The second derivative of G in λ is (minus β times) a *variance*: the fluctuation of the exchange energy about its mean, in the constrained reference. Two lines of moment algebra do the evaluation. The fluctuating object is $\sum_{(ij)} J_{ij}(s_i s_j - m_i m_j)$; squaring and averaging over independent spins, distinct bonds decorrelate, and each bond contributes $\langle (s_i s_j)^2 \rangle - \cdots = 1 - m_i^2 m_j^2$ — from which the constraint (the b_i readjusting at each order so the magnetizations stay pinned) removes the pieces where only *one* spin of the bond fluctuates, $m_j^2(1-m_i^2) + m_i^2(1-m_j^2)$, leaving exactly

$$(1 - m_i^2 m_j^2) - m_j^2(1-m_i^2) - m_i^2(1-m_j^2) \;=\; (1-m_i^2)(1-m_j^2)$$

per bond: *the product of the two ends' single-site variances* — that is, of their susceptibilities, the very objects that carried the echo. The second-order term is therefore $G_2 = -\frac{\beta}{2}\sum_{(ij)} J_{ij}^2 (1-m_i^2)(1-m_j^2)$, and truncating the ladder here gives the **TAP free energy**:

$$F_{\mathrm{TAP}}(m) \;=\; \underbrace{-\sum_{(ij)} J_{ij}\, m_i m_j - \sum_i h_i m_i - T\sum_i S(m_i)}_{F_{\mathrm{naive}}(m)\ \text{(Week 5)}} \;-\; \frac{\beta}{2}\sum_{(ij)} J_{ij}^2 (1-m_i^2)(1-m_j^2) \tag{11.2}$$

TAP free energy

Stationarity gives the equations. Differentiate in m_i: the naive part gives Week 5's terms ($-h_i^{\text{eff}}$ from the energy, $T\,\text{artanh}(m_i)$ from the entropy), and the new term contributes $+\beta\, m_i \sum_j J_{ij}^2(1-m_j^2)$. Setting the sum to zero and inverting:

$$\boxed{m_i \;=\; \tanh\Big(\beta\Big[\sum_j J_{ij}\, m_j + h_i\Big] \;-\; \beta^2\, m_i \sum_j J_{ij}^2(1-m_j^2)\Big)} \tag{11.3}$$

TAP equations

— the **TAP equations** (Thouless, Anderson, and Palmer 1977): Week 5's self-consistency with one new term inside the tanh, and that term is precisely the echo of Equation 11.1. The two derivations serve different purposes: the physical route says *why* (a spin must not respond to its own reflection), the expansion says *what* (second order in the coupling) and *what next* (the third and higher rungs are generated mechanically — Georges and Yedidia (1991) give the algorithm, and nothing but patience stands between you and G_3).

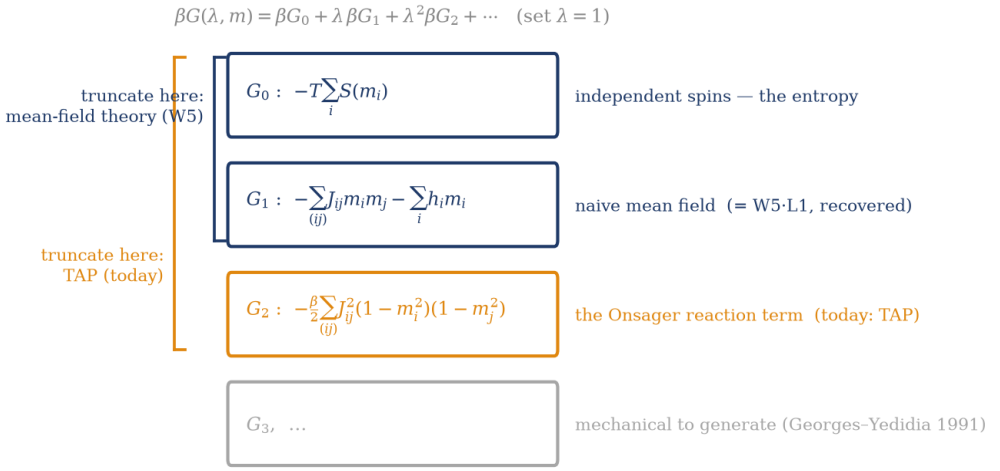

Figure 11.2: The engine's map: mean field is not a guess but a truncation. The constrained free energy expands in the interaction strength as $\beta G = \beta G_0 + \lambda\,\beta G_1 + \lambda^2 \beta G_2 + \cdots$: order zero is the entropy of pinned free spins, order one is Week 5's mean-field theory — recovered, not assumed — and order two is the Onsager reaction term, whose per-bond weight $(1-m_i^2)(1-m_j^2)$ is the product of the two ends' susceptibilities. Truncate after G_1 and you hold W5 · L1; truncate after G_2 and you hold TAP; the ladder continues mechanically (Georges and Yedidia 1991).

Take-home 2

The Plefka expansion of the fixed-m free energy in the coupling: order one *is* Week 5's mean field; order two adds $-\frac{\beta}{2}\sum J_{ij}^2(1-m_i^2)(1-m_j^2)$, whose stationarity gives the TAP equations — the effective field minus the spin's own echo, reflected through J_{ij}^2 and weighted by the neighbors' susceptibilities. Higher orders are mechanical.

11.4 When the echo matters — and what TAP costs

The scaling verdict. The reaction term rides on one number, $\sum_j J_{ij}^2$, and its value across coupling ensembles is the central scaling result (Figure 11.3):

- **Uniform ferromagnet,** $J_{ij} = J/N$**:** $\sum_j J_{ij}^2 = J^2(N-1)/N^2 \to 0$. The echo *vanishes in the thermodynamic limit* — and Week 5's audit, which found naive mean field exact on precisely this model, is now *explained* rather than observed. Nothing was lucky about it: the model's couplings are individually so weak that no spin meaningfully polarizes any neighbor.
- **SK disorder,** $J_{ij} \sim \mathcal{N}(0, J^2/N)$**:** $\sum_j J_{ij}^2 \to J^2$ by the law of large numbers — **order one, at every temperature.** Individual couplings are strong enough ($\sim 1/\sqrt{N}$) to polarize neighbors, and the echoes, though random in sign *as fields*, add their *squares*. For disordered couplings the reaction term is not a refinement; it is the difference between a wrong theory and a usable one — which is why SK's "solvable" model needed TAP. On the SK model the term takes a famously compact form: $\sum_j J_{ij}^2(1-m_j^2) \to J^2(1-q)$ with $q = \frac{1}{N}\sum_j m_j^2$ — the *Edwards–Anderson overlap*, Week 2's q, returning to the course in its home country.
- **Sparse graph, degree** z**, bonds of fixed strength** J**:** $\sum_j J_{ij}^2 = zJ^2$ — the correction matters whenever individual bonds stay strong, however few.

The same physics appeared in Week 2, without its name. The AGS retrieval theory's noise-amplification factor $r = (1-C)^{-2}$ in Equation 4.5 — the term that moved the capacity from $2/\pi$ to 0.138 — is the network responding to its own errors: flipped bits polarize the fields that flipped them. That was a reaction effect, quoted as a result then, derived as a principle now. Hebb couplings at extensive load are exactly the disordered, $\mathcal{O}(1)$-echo kind — *learning writes disordered couplings*, so the models this course cares about live generically on the corrected side of the verdict.

What was traded. Week 5's one-sided guarantee does not survive the correction. $F_q \geq F$ held because F_q was the *exact* variational functional evaluated on a restricted family. F_{TAP} is a *truncated expansion* — the exact functional of no family at all — and the guarantee does not survive truncation: F_{TAP} is generally much closer to the truth and can sit on **either side** of it (in the example below it sits below the exact F at every temperature shown). This is a real loss, and it recurs across the course's methods: *contrastive divergence traded unbiasedness for tractability (Week 3); TAP trades the bound for accuracy.* Each successive method in the course gains precision at the expense of formal guarantees. (The next lecture partially restores the variational character at the pairwise level, via the Bethe free energy; the loss is specific to the Plefka truncation, not to the correction program.)

Many solutions, and trouble iterating. A further cost is physical rather than formal. At high temperature the TAP equations have one solution; lower the temperature into the glass region of a disordered model and they develop *many* — exponentially many for SK — each a metastable valley of the corrected free-energy landscape, the *many states* of Mézard, Parisi, and Virasoro (1987). This is Week 2's ruggedness surfacing inside a deterministic approximation, and it has an inference-side reading Week 5 prepared: multiple TAP solutions are multiple local optima of a corrected ELBO-like objective — the two ferromagnetic wells of last week, multiplied by disorder into a landscape. Practical corollary: naive fixed-point iteration of Equation 11.3 can oscillate or diverge exactly where the physics is interesting (our

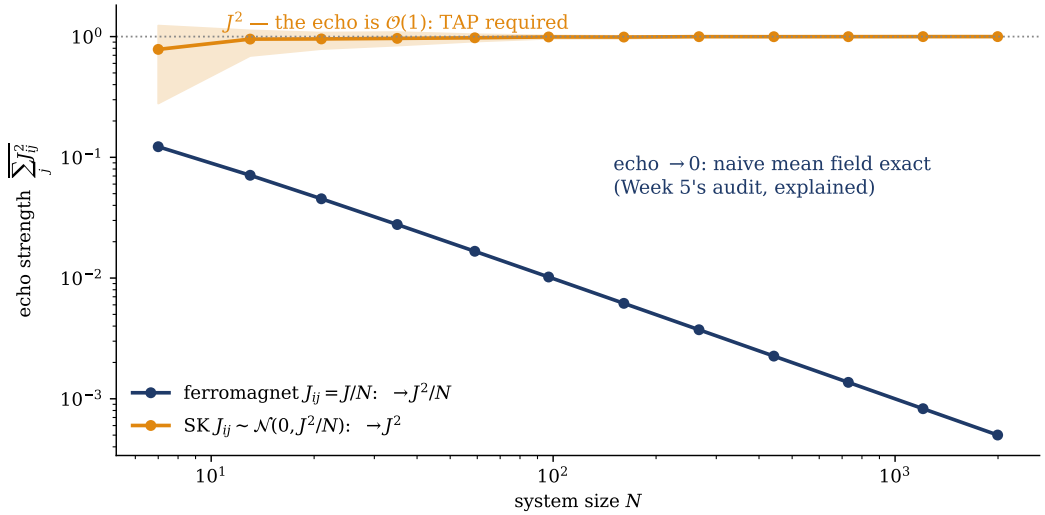

Figure 11.3: The scaling verdict, sampled: the echo's driver $\frac{1}{N}\sum_i \sum_j J_{ij}^2$ against system size for the two coupling ensembles (20 instances per size; bands span the instances). For the uniform ferromagnet $J_{ij} = J/N$ (navy) the sum decays as J^2/N — the reaction term vanishes in the thermodynamic limit, which is why Week 5's audit found naive mean field exact on exactly this model. For SK disorder $J_{ij} \sim \mathcal{N}(0, J^2/N)$ (orange) it converges to J^2: order one at every size and every temperature. One number decides who needs TAP — and learned, data-built couplings are of the disordered kind.

example uses damping), and the iteration whose error can be tracked exactly, step by step, arrives in the next lecture under an algorithm's name.

The ML echo. The same machinery runs inference in learning systems directly: Kappen and Rodríguez (1998) showed that a Boltzmann machine's intractable statistics — Week 3's negative phase $\langle s_i s_j \rangle_{\text{model}}$ — can be estimated *deterministically* by solving the TAP equations and reading correlations from linear response, replacing the sampling chains of contrastive divergence with a corrected mean-field computation. The correction term remains a working component of inverse-Ising and Boltzmann-machine inference.

Trap

"Second order" does not mean small — and F_{TAP} is not a bound. ① For disordered couplings the second-order term is $\mathcal{O}(1)$ — the same size as the first — at every temperature; order in the expansion is position on a ladder, not magnitude, and the magnitude is set by $\sum_j J_{ij}^2$ alone. ② Truncating the ladder destroys Week 5's one-sidedness: F_{TAP} can dip below the true F and promises no error sign. Bound versus estimate — always know which you are holding, and never report a TAP solution as *the* solution inside the glass phase, where there are exponentially many.

Take-home 3

The echo's size is $\sum_j J_{ij}^2$: vanishing for the uniform ferromagnet (J^2/N — Week 5's exactness, explained), order one for SK-type disorder ($J^2(1-q)$, with Week 2's Edwards–Anderson q returning) — and learning writes disordered couplings. The price of the correction: the bound is gone (TAP is an estimate — the CD trade again), and inside the glass phase the equations have exponentially many solutions with real convergence trouble for naive iteration.

11.5 Example: one term, from wrong to right

One disordered instance, small enough to enumerate, so that every claim can be checked directly. Take $N = 10$ spins with SK couplings $J_{ij} \sim \mathcal{N}(0, J^2/N)$, $J = 1$, and small random fields ($h_i \sim \mathcal{N}(0, 0.2^2)$, to break the global flip symmetry); $2^{10} = 1024$ states gives the exact F and $\langle s_i \rangle$ by brute force. Solve Week 5's naive equations and today's TAP equations by damped iteration from the same start, and compare — the same three-column protocol as Week 5's example, now with disorder doing the judging:

T	RMS error, naive MF	RMS error, TAP	F_{exact}	F_{naive}	F_{TAP}
2.0	0.051	0.009	−15.06	−14.01	−15.15
1.5	0.227	0.027	−11.96	−10.69	−12.10
1.2	0.419	0.064	−10.22	−9.01	−10.43
0.8	0.460	0.225	−8.19	−7.49	−8.65

The columns track the predictions above. The **error columns** are the headline: at $T = 1.5$ the single reaction term cuts the magnetization error *eightfold* (0.227 → 0.027); at $T = 1.2$, sixfold. One term takes the theory from wrong to right, on a model whose $\sum_j J_{ij}^2$ is order one — exactly as the verdict predicts. The **free-energy columns** show both costs at once: F_{naive} sits *above* the truth at every temperature (Week 5's bound, holding faithfully — off by 1.3 at $T = 1.5$), while F_{TAP} sits far closer (0.15 at $T = 1.5$) but *below* the truth — the lost bound, visible directly. And the **last row** makes the third cost numerical: at $T = 0.8$, inside the glass phase of this model, TAP's own error grows to 0.23 — the territory of many solutions and higher-order physics, where no truncation of the ladder is safe and the next lecture's algorithmic ideas become necessary. Figure 11.4 draws the $T = 1.5$ magnetizations and the free-energy curves. For contrast, run the same pipeline on the uniform ferromagnet ($J_{ij} = J/N$, same N): the naive and TAP magnetizations agree to 0.03 — a gap that is pure $1/N$, shrinking with size — and the echo term is invisible, as the verdict says it must be.

A spin must not respond to its own echo — subtracting the echo is the entire first correction to mean field, and on disordered couplings the echo is $\mathcal{O}(1)$.

The week continues from here. The evening before the next lecture: the warm-up card, on the belief-propagation introduction of Mézard and Montanari (2009), ch. 14 — the cavity field derived above is the object that chapter passes along the edges

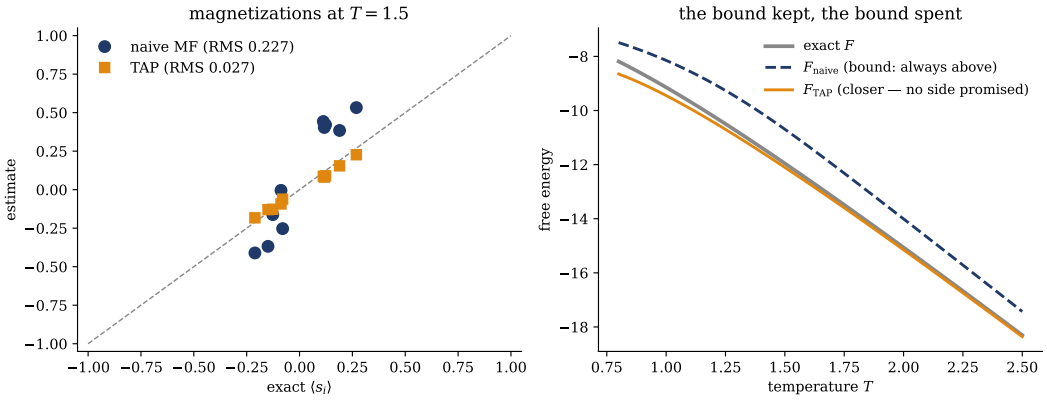

Figure 11.4: The correction at work, on an enumerable SK instance ($N = 10$, $J_{ij} \sim \mathcal{N}(0, 1/N)$, small random fields; exact values from all 1024 states). Left: per-spin magnetizations at $T = 1.5$, estimate against truth — naive mean field (navy) scatters off the diagonal (RMS error 0.227); adding the one reaction term (TAP, orange) collapses the points onto it (RMS 0.027). Right: free energies across temperature. F_{naive} (navy dashed) sits above the exact F everywhere — Week 5's variational bound, intact. F_{TAP} (orange) hugs the truth an order of magnitude closer — and from *below*: the truncated expansion keeps no side of the truth, the guarantee spent for the accuracy. Inside the glass phase (left edge) all truncations degrade — the many-states territory.

of a graph. In the next lecture, the cavity goes onto a sparse graph and becomes an algorithm with a variational functional of its own; the tutorial is derivation-led, deriving that algorithm at the board alongside its modern descendant from signal processing.

11.6 Outlook: from cavity to messages

Module 2's program now stands as follows. The variational road now has two stations: the bound (Week 5 — exact functional, restricted family, guaranteed sign) and the correction (today — truncated expansion, better numbers, no sign), with the ladder above them mechanical (Georges and Yedidia 1991). The next lecture adds the third and, algorithmically, the most consequential: put the cavity field on a *sparse graph*, where "compute j's magnetization with i removed" becomes a quantity passed along an edge, and the self-consistency becomes **belief propagation** — exact on trees, equipped with its own free energy (Bethe's, standing where Gibbs–Bogoliubov stood), and the ancestor of algorithms that decode your phone's error-correcting codes. The tutorial derives it at the board. The convergence trouble flagged today has an algorithmic resolution there.

One connection to Week 2 remains open. When W2 · L2 quoted the AGS phase diagram, it deferred the machinery behind quenched averages of $\log Z$ to Module 2 and named the replica method as the technology. That connection can now be made precise. The replica route and the cavity route are the two great roads through disordered systems; they give the same answers where both apply. This course's acquisition is the **cavity route** — Onsager's echo, TAP, and from the next lecture message passing — because it is the road that became *algorithms*: belief propagation,

approximate message passing, the inference engines of modern high-dimensional statistics. The replica method remains, in this course, a named landmark: you now know what it computes (the quenched average W2 could not take), what went wrong in its naive form (SK's negative entropy, $S(0) = -1/(2\pi) \approx -0.16$ per spin in the replica-symmetric solution — the impossibility that provoked TAP), and what fixed it (Parisi's replica symmetry breaking (Parisi 1979), the many-states structure that TAP's multiple solutions shadow from our side of the mountain, honored with the 2021 Nobel prize; Nishimori (2001) carries the full story, and Mézard, Parisi, and Virasoro (1987) is its book). The AGS diagram stays a quoted result — but the *physics* inside its equations, feedback and reaction and self-consistency, has now been derived within Module 2's framework.

12 Belief propagation: the cavity becomes an algorithm

Recommended reading

Core (~1 h). Mézard and Montanari (2009), Ch. 14 — belief propagation built from the cavity method, the Bethe free energy, and the physics/coding/inference unification; this was the warm-up reading, and the engine below follows its logic. Alongside it, Yedidia, Freeman, and Weiss (2003) — the definitive statement of this lecture's variational result, that BP fixed points are stationary points of the Bethe free energy, plus the generalized (region-graph) extensions we only name.
Optional background. Donoho, Maleki, and Montanari (2009) — AMP and its Onsager correction, the tutorial's second hard problem: skim the algorithm and the role of the correction term, and *stop before* the state-evolution analysis so the crack stays fresh; Pearl (1988) — belief propagation's birth in AI; Bethe (1935) — the approximation that named the free energy; Gallager (1962) — the same algorithm decoding codes, twenty-six years early.
Prerequisite reminder. All of Chapter 11 — the cavity field Equation 11.1, the Onsager reaction term, the TAP equations, and the $\sum_j J_{ij}^2$ verdict — plus Week 5's mean-field functional. New content: messages on a graph, exactness on trees and its chain instance (the forward–backward algorithm), the Bethe free energy, the BP→TAP reduction, and AMP as message passing for the dense linear model.

12.1 Passing the cavity field

The previous lecture's cavity field — the field on a spin computed *in the cavity where that spin is removed* — is, in structure, a *message*. Today the identification becomes precise, and the consequences run through the rest of the week. Compute the cavity field not once, as a correction, but *recursively, along the edges of a graph*: every node tells every neighbor what it believes in that neighbor's absence, each such report built from the reports the node itself received from everyone else. Iterate to consistency. That is **belief propagation**, and on a tree it is not an approximation at all — it returns the *exact* marginals of a 2^N-state distribution in a number of operations linear in N. (One question suggests itself immediately: *what is a message, if not a marginal?* The difference between the two is the entire method, and it gets a direct answer below.)

The history is the course's thesis in miniature, because this algorithm was invented **three times, in three fields that were not talking to each other**. Bethe (1935) invented it in statistical physics, as an approximation for alloys — the "Bethe approximation" whose free energy we meet today. Gallager (1962) invented it in

coding theory, as an iterative decoder for the error-correcting codes that were decades ahead of their hardware — the LDPC codes that now run in your phone's radio. And Pearl (1988) invented it in artificial intelligence, as exact inference on the graphical models that organized modern probabilistic reasoning. Three communities and three notations converged on one algorithm — and the demonstration that they are one thing, assembled in the book you read for the warm-up (Mézard and Montanari 2009), is as sharp a statement of this course's premise as any: physics, coding, and inference are one subject speaking three languages.

The driving question picks up where the previous lecture's verdict left off. TAP's reaction term was the right correction *for a dense graph* — the Plefka expansion said so. But the graphs of real inference problems are not all dense: they are sparse, structured, tree-like, loopy. *What is the right correction for an arbitrary graph?* Today's answer is structural rather than perturbative: Pass the cavity field along each edge and let the graph itself decide — on a tree the result is exact, and on a dense graph, at the end of the lecture, the messages collapse and their correction *is* the Onsager term. Message passing contains TAP.

> Take-home 1
>
> Belief propagation is the previous lecture's cavity field, passed recursively along a graph's edges: each node tells each neighbor what it believes in that neighbor's absence. On a tree it returns the *exact* marginals — and it was invented three separate times (Bethe 1935, Gallager 1962, Pearl 1988) because physics, coding, and inference are one problem.

12.2 Node beliefs versus edge messages

One structural dichotomy organizes the method and is worth stating before the algebra. Week 5's mean field tracked *one number per node* — the magnetization m_i — and paid for that economy with zero correlations. Belief propagation tracks *one number per directed edge*: the message $u_{k\to i}$, a different report for every recipient, because what k believes *in i's absence* genuinely depends on who i is. That per-edge bookkeeping is exactly what lets correlation flow along the graph — and it is the previous lecture's cavity distinction (m_j versus $m_j^{(i)}$), promoted from a second-order correction to the method's primary variable.

Figure 12.1 draws the central picture, both halves. Left: a tree, with one message flowing along one directed edge — $u_{k\to i}$, everything the subtree behind k has to say to i, condensed into a single effective field. Right: the reason trees are special. Remove node i, and a tree *falls apart* — the neighbors' subtrees are genuinely, exactly independent of one another, because the only path between them ran through i. The cavity assumption that the previous lecture's TAP invoked approximately ("the neighbors are independent in my absence") is, on a tree, simply *true*. The results of this lecture follow from that picture: exactness where removing a node disconnects, correction terms where it does not.

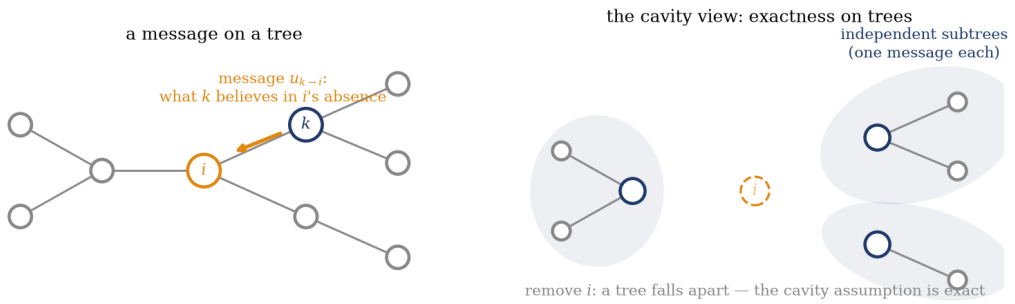

Figure 12.1: The central picture of the lecture. Left: a message on a tree — $u_{k\to i}$ carries everything the subtree behind k has to say to i, condensed into one effective field: *what k believes in i's absence*. Right: why trees are special — remove i and the tree falls apart into independent subtrees, one per neighbor, each summarized by one incoming message. The cavity assumption ("my neighbors are independent in my absence") is exactly true on a tree; every loop in a graph is a way for it to fail.

12.3 The engine: belief propagation

Setup. A pairwise model on a graph — Ising, for concreteness and continuity: spins $s_i = \pm 1$, couplings J_{ij} on the edges, fields h_i, temperature T live as it has been all module. Write ∂i for the set of i's neighbors. Two cavity objects, and then the recursion closes on itself.

The cavity field, per edge. Let $h_{i\to j}$ be the effective field on i *with the edge to j cut* — the previous lecture's cavity field, now directed. In the cavity, i's remaining neighbors act on it independently (exactly so on a tree), so their contributions add.

The message. What does neighbor k contribute to i? Trace k out. Spin k sits in its own cavity field $h_{k\to i}$ (its field from everything *except* i) and couples to i through J_{ki}; summing over $s_k = \pm 1$ produces an effective field on i, and two lines of algebra give it in closed form. Define $u_{k\to i}$ by $\sum_{s_k} e^{\beta J_{ki} s_k s_i + \beta h_{k\to i} s_k} \propto e^{\beta u_{k\to i} s_i}$; taking the ratio of the $s_i = \pm 1$ cases,

$$e^{2\beta u_{k\to i}} = \frac{\cosh\beta(h_{k\to i} + J_{ki})}{\cosh\beta(h_{k\to i} - J_{ki})} \qquad \Longrightarrow \qquad \tanh(\beta u_{k\to i}) = \tanh(\beta J_{ki})\,\tanh(\beta h_{k\to i}),$$

that is,

$$\boxed{u_{k\to i} \;=\; \frac{1}{\beta}\,\operatorname{artanh}\Big[\tanh(\beta J_{ki})\,\tanh\left(\beta\, h_{k\to i}\right)\Big]} \tag{12.1}$$

BP message: what k tells i

In words: the message is k's cavity magnetization $\tanh(\beta h_{k\to i})$, *attenuated by the bond* through $\tanh(\beta J_{ki})$ — a weak bond transmits almost none of k's belief, a strong bond passes it on nearly verbatim, and the artanh converts the result back into field units so that messages can be *added*.

The recursion, and the marginals. The cavity field on i toward j collects the messages from all of i's *other* neighbors,

$$h_{i\to j} \;=\; h_i + \sum_{k\in\partial i\setminus j} u_{k\to i},$$

and the definition closes on itself: messages determine cavity fields determine messages. Iterate to a fixed point (sweep the directed edges; damping helps on hard instances, as it did in the previous lecture). Once converged, the *full* marginal at a node combines **all** incoming messages:

$$m_i \;=\; \tanh\Big(\beta\Big[h_i + \sum_{k\in\partial i} u_{k\to i}\Big]\Big) \tag{12.2}$$

BP marginal: all messages combined

And here is the promised answer to the opening question. **A message is not a marginal.** The message $u_{k\to i}$ is a belief *in i's absence* — computed from $h_{k\to i}$, which deliberately excludes what i itself has been telling k. The marginal m_i is what you believe once *every* neighbor has told you what they believe without you. Keeping the two straight — always asking "who was removed?" — is not pedantry; it is precisely the discipline that prevents the double-counting the previous lecture was about. A node that fed its own message back into the reply it receives would be responding to its own echo.

Exactness on trees. Now the theorem, and after Figure 12.1 it is one paragraph. The only assumption anywhere above is the cavity assumption: that i's neighbors are statistically independent once i is removed. On a tree, removing i *disconnects* the neighbors' subtrees — there is no path left between them — so the assumption is exactly true, at every node, at every step of the recursion. By induction from the leaves (whose cavity fields are just their local fields, no assumption needed), every message is exact, hence every marginal Equation 12.2 is **exact**. A 2^N-sum inference problem, solved in $\mathcal{O}(N)$ message updates, with no approximation: belief propagation on a tree is an exactly solvable algorithm. And the same paragraph locates the failure mode in advance: on a graph *with* loops, removing i does not disconnect its neighbors — influence can travel around the loop and return — which is exactly the self-reaction that trees lack and the previous lecture subtracted. Loops are where messages become echoes.

The oldest special case: the chain. The simplest tree has no branching at all, and on it the theorem lands on an algorithm invented yet a fourth time, in yet a fourth field (Welling, Lu, and Holdijk 2026, ch. 6 on belief propagation). Consider a *hidden Markov model*, the workhorse of speech recognition and biological sequence analysis: hidden states z_t evolving under a transition law $p(z_t \mid z_{t-1})$, each emitting an observation x_t with likelihood $p(x_t \mid z_t)$. With the observations clamped, the posterior over the hidden sequence is a chain of pairwise factors,

$$p(z_0,\dots,z_T \mid x) \;\propto\; p(z_0)\,p(x_0 \mid z_0)\prod_{t=1}^{T}\underbrace{p(z_t \mid z_{t-1})\;p(x_t \mid z_t)}_{\equiv\,\psi_t(z_{t-1},\,z_t)}.$$

The graph is a chain — a tree without branches — so belief propagation on it is exact, and it simplifies twice. Each interior node has exactly two neighbors, so "all neighbors but the recipient" is a single term and the recursion loses its product: messages simply sweep, once left to right and once right to left. And because z_t need not be binary, the message travels as a function over states rather than as one number — the Ising field of Equation 12.1 was the binary compression of this object,

the artanh algebra doing nothing more than storing a two-state message in a single scalar. Writing α_t for the rightward message into z_t and β_t for the leftward one, the sweeps read

$$\alpha_t(z_t) \;=\; \sum_{z_{t-1}} \psi_t(z_{t-1}, z_t)\, \alpha_{t-1}(z_{t-1}), \qquad \beta_t(z_t) \;=\; \sum_{z_{t+1}} \psi_{t+1}(z_t, z_{t+1})\, \beta_{t+1}(z_{t+1}),$$

forward–backward: BP on a chain

seeded at the ends by the leaf initialization of the induction above ($\alpha_0 = p(z_0)\, p(x_0 \mid z_0)$, $\beta_T = 1$), and the exact posterior marginal combines the two incoming messages exactly as Equation 12.2 prescribes, $p(z_t \mid x) \propto \alpha_t(z_t)\, \beta_t(z_t)$. These two sweeps are the **forward–backward algorithm** of hidden-Markov-model inference (Baum et al. 1970), running in speech recognizers and genome annotators for half a century, and the tree-exactness theorem is the reason it returns exact posteriors in $\mathcal{O}(T)$ operations. The count from the opening section was in fact low: Bethe found this algorithm in alloys, Gallager in codes, Pearl in expert systems — and Baum, eighteen years before Pearl, in time series, a chain being the simplest tree, narrow enough that its users did not recognize the underlying message-passing structure.

12.4 The Bethe free energy — and BP contains TAP

Two results complete the theory: one gives BP a variational foundation, and the other connects it back to the previous lecture's TAP equations.

BP is variational: the Bethe free energy. Is BP just a clever iteration, or does it optimize something? The answer starts from a tree identity worth knowing on its own: on a tree, the full Gibbs distribution *factorizes exactly* over its node and edge marginals,

$$p(s) \;=\; \frac{\prod_{(ij)} b_{ij}(s_i, s_j)}{\prod_i b_i(s_i)^{\,d_i - 1}},$$

where b_i, b_{ij} are the single-node and pairwise marginals and d_i is node i's degree — each node over-counted once per extra edge and divided back out. Insert this form into $F = \langle E \rangle + T\langle \log p \rangle$ and the free energy assembles itself from the beliefs alone:

$$F_{\text{Bethe}}[b] \;=\; \sum_{(ij)} \sum_{s_i, s_j} b_{ij}\, E_{ij} \;+\; \sum_i \sum_{s_i} b_i\, E_i \;+\; T \sum_{(ij)} \sum_{s_i, s_j} b_{ij} \log b_{ij} \;-\; T \sum_i (d_i - 1) \sum_{s_i} b_i \log b_i, \tag{12.3}$$

Bethe free energy

with $E_{ij} = -J_{ij} s_i s_j$ and $E_i = -h_i s_i$. This is the **Bethe free energy**: the *pairwise-correlation upgrade* of Week 5's mean-field functional, tracking beliefs on edges and not just nodes. Note where the approximation lives — the *energy* term is exact on any graph, because the Hamiltonian only ever touches pairs; it is the *entropy* whose node-and-edge counting is borrowed from the tree. On a tree, by the identity above, F_{Bethe} equals the true free energy exactly. Now treat the beliefs as free variables, constrained to normalize and to be *consistent* — each pairwise belief must marginalize to its nodes' beliefs, $\sum_{s_j} b_{ij}(s_i, s_j) = b_i(s_i)$ — and ask what stationarity delivers.

The constrained optimization runs in three steps (Yedidia, Freeman, and Weiss 2003). First, attach a Lagrange multiplier $\lambda_{j\to i}(s_i)$ to each marginalization constraint — one function per *directed edge*, the same bookkeeping unit as a message, which is no accident. Second, vary the pairwise belief: setting the derivative of the Lagrangian with respect to $b_{ij}(s_i, s_j)$ to zero gives $E_{ij} + T(\log b_{ij} + 1) - \lambda_{j\to i}(s_i) - \lambda_{i\to j}(s_j) = \text{const}$, i.e.

$$b_{ij}(s_i, s_j) \;\propto\; e^{-\beta E_{ij}}\, e^{\beta \lambda_{j\to i}(s_i)}\, e^{\beta \lambda_{i\to j}(s_j)}$$

— a Boltzmann factor for the bond, tilted at each end by that end's multiplier. Third, vary the node belief, whose entropy enters Equation 12.3 with the *negative* weight $-(d_i - 1)$ — which flips the usual signs — while every incident edge contributes its multiplier; the same bookkeeping gives

$$b_i(s_i) \;\propto\; \exp\Big(\frac{\beta}{d_i - 1}\Big[E_i(s_i) + \sum_{j\in\partial i} \lambda_{j\to i}(s_i)\Big]\Big).$$

(For a leaf, $d_i = 1$, the node-entropy term drops out of Equation 12.3 — its weight is $d_i - 1 = 0$, though the node-energy term remains — and the constraint alone fixes b_i, consistent with the leaf-initialized recursion above.) On binary spins any function of s_i is affine, so each multiplier is one number per directed edge, $\lambda_{j\to i}(s_i) = \mu_{j\to i}\, s_i$ up to a constant. Substitute the ansatz $\mu_{j\to i} = h_{i\to j}$ — *the multiplier is the cavity field* — and both stationarity conditions close. The pairwise belief becomes $b_{ij} \propto e^{\beta[J_{ij} s_i s_j + h_{i\to j} s_i + h_{j\to i} s_j]}$, and imposing the marginalization constraint on it is exactly the two-line cosh trace that produced the message Equation 12.1. The node exponent collects, per unit s_i,

$$\frac{1}{d_i - 1}\Big[-h_i + \sum_{j\in\partial i} h_{i\to j}\Big] \;=\; \frac{1}{d_i - 1}\Big[-h_i + \underbrace{d_i\, h_i + (d_i - 1)\textstyle\sum_k u_{k\to i}}_{\text{each } k \text{ missed exactly once over the } j\text{-sum}}\Big] \;=\; h_i + \sum_{k\in\partial i} u_{k\to i},$$

reproducing the marginal Equation 12.2. The consistency constraints *are* the message equations; their multipliers *are* the cavity fields. The identification is therefore:

BP fixed points = stationary points of the Bethe free energy

BP is variational (Yedidia–Freeman–Weiss)

One scope note: the negative node-entropy weight in Equation 12.3 means F_{Bethe} is not jointly convex in the beliefs off trees — stationary points need not be minima, which is one root of loopy BP's convergence trouble. Module 2's organizing principle — approximate the distribution by optimizing a functional — therefore extends through the entire week: mean field optimized a node functional (W5, with a bound), TAP optimized a truncated expansion (the previous lecture, bound spent), and BP optimizes the Bethe functional — exact on trees, and on loopy graphs a principled approximation ("loopy BP") whose functional, like TAP's, promises no side of the truth.

BP contains TAP. The week's thesis can now be made rigorous. Put belief propagation on the previous lecture's *dense* graph — every pair coupled, weakly, $J \sim 1/\sqrt{N}$ — and expand. For weak bonds the message Equation 12.1 linearizes,

$u_{k\to i} \approx J_{ki}\, m_{k\to i}$, so the marginal reads $m_i = \tanh(\beta[h_i + \sum_k J_{ki}\, m_{k\to i}])$: naive mean field, except that the inputs are *cavity* magnetizations. And the cavity magnetization differs from the full one by exactly the neighbor's linear response to i's removal:

$$m_{k\to i} \;\approx\; m_k - \underbrace{\beta(1-m_k^2)}_{\chi_k} J_{ki}\, m_i \qquad \Longrightarrow \qquad m_i = \tanh\Big(\beta\Big[h_i + \sum_k J_{ki} m_k - \beta\, m_i \sum_k J_{ki}^2 (1-m_k^2)\Big]\Big)$$

— the TAP equations Equation 11.3, term for term: **the message correction is the Onsager reaction term** (Figure 12.2; the systematic $1/\sqrt{N}$ bookkeeping is script-level, but the two lines above are the whole idea). The week's thesis is now a theorem rather than a slogan: *message passing is mean-field plus Onsager*. On a sparse graph, BP is the richer object — a full message per edge; on a dense one, the many tiny messages collapse and what survives of them, collectively, is precisely the reaction term the previous lecture derived from the Plefka ladder. One idea — the cavity — takes whatever form the graph topology dictates.

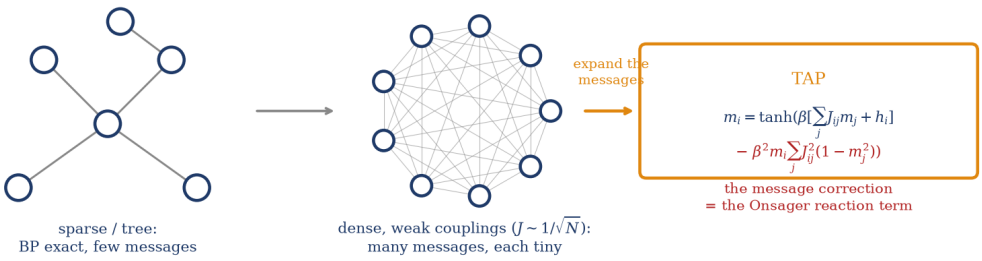

Figure 12.2: The week's thesis, made visual. On a sparse graph (left) belief propagation carries a full message per edge and is exact on trees. Densify the graph — many couplings, each weak, $J \sim 1/\sqrt{N}$ (center) — and the messages linearize and collapse: expanding the cavity magnetizations to leading order (right) yields exactly the TAP equations, the messages' collective correction to naive mean field being the Onsager reaction term of Chapter 11. Belief propagation contains TAP; the graph topology chooses the form.

Trap

BP is exact only on trees — and "works" is not "exact." On loopy graphs, belief propagation may fail to converge, and its fixed points can return wrong marginals: a short loop lets a node's influence travel around and return as fresh evidence — the self-reaction trees structurally lack (the example below measures the damage growing with the loop's coupling strength). The Bethe free energy is likewise *not a bound* off trees. Loopy BP nonetheless works astonishingly often — it is decoding error-correcting codes in your pocket right now — but that is an empirical grace with known failure modes (short loops, strong couplings), not a guarantee. Generalized BP on region graphs (Yedidia, Freeman, and Weiss 2003) repairs loops systematically; we name it here without developing it.

Take-home 2

BP fixed points are stationary points of the Bethe free energy — the pairwise upgrade of Week 5's functional, exact on trees, with the messages as Lagrange multipliers (Yedidia, Freeman, and Weiss 2003). And on a dense weak-coupling graph the messages collapse: the cavity correction $m_{k\to i} \approx m_k - \chi_k J_{ki} m_i$ reproduces the TAP equations exactly. **Message passing = mean-field + Onsager.** On loops, BP is usually excellent and never guaranteed; Bethe is not a bound.

12.5 AMP: TAP for the dense linear model

One more graph, chosen because you will derive message passing on it yourselves in the tutorial — and because it is where this fifty-year-old physics became a modern algorithm with a proof. The problem is **compressed sensing**: observe $y = Ax + w$, with A an $M \times N$ dense random matrix (i.i.d. Gaussian entries), $M < N$ measurements of an N-dimensional but *sparse* signal x, noise w. Undersampled linear regression, rescued by sparsity — the workhorse of modern signal recovery (accelerated MRI being the textbook application). As an inference problem it lives on a dense *bipartite* factor graph (Figure 12.3, left): every measurement touches every signal entry, so literal BP would pass $2MN$ messages per iteration — computationally impractical, but structurally familiar: a dense graph of weak individual influences, exactly the regime where this lecture just taught you messages collapse.

They do. Because each message is a sum of many weak, nearly independent contributions, the central limit theorem collapses the $2MN$ messages onto a few *vector* updates — **approximate message passing** (Donoho, Maleki, and Montanari 2009). Its structure (stated here; the derivation is deferred to the tutorial): iterate an *estimate* step, $x^{t+1} = \eta_t(x^t + A^T z^t)$, where η_t is a scalar *denoiser* (for sparse signals, a soft-threshold that shrinks small entries to zero), and a *residual* step,

$$z^t \;=\; y - Ax^t \;+\; \underbrace{\frac{1}{\delta}\, z^{t-1} \left\langle \eta'_{t-1} \right\rangle}_{\text{the Onsager correction}} \quad , \qquad \delta = M/N,$$

AMP: the reaction term, again

whose last term is the important one: an extra piece of the residual, proportional to the *previous* residual and to the average sensitivity $\langle \eta' \rangle$ of the denoiser — a memory term that naive iterative thresholding does not have. It is called, in the AMP literature and not by coincidence, the **Onsager correction term**, and its job is the previous lecture's job: the estimate x^t was built *from* the residual, so the naive residual $y - Ax^t$ is contaminated by the estimate's own influence reflected back through A — an echo, through a matrix. Subtracting it *decorrelates* the effective noise, keeping $x^t + A^T z^t$ statistically equivalent to *signal plus fresh Gaussian noise* at every iteration — which is precisely what makes AMP analyzable: a one-dimensional recursion (*state evolution*) then predicts the algorithm's error at every step, **exactly** in the large-system limit. Drop the term and the iteration derails; keep it and the algorithm comes with a proof. The reaction term is not a refinement here — it is the difference between a heuristic and a theorem, exactly as it was the difference between SK's "solution" and TAP's.

Two further results connect AMP to the module's themes. State evolution has a *sharp threshold*: below a critical measurement rate δ, recovery fails; above it, in the noiseless limit AMP reconstructs the signal essentially perfectly (with measurement noise the transition is a jump in the reconstruction *error* rather than to perfection) — a **phase transition in an inference problem**, the same order-parameter physics that magnetized Week 5's ferromagnet and collapsed Week 2's memory, now deciding whether an MRI scan can be undersampled. The derivation of AMP from the collapsing messages, and the full demonstration of how the Onsager correction emerges from the message bookkeeping, are deferred to the tutorial.

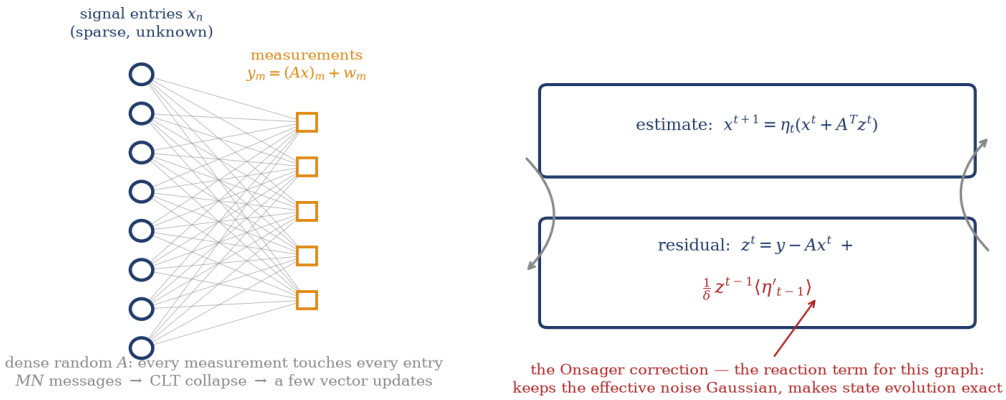

Figure 12.3: AMP in one picture. Left: compressed sensing as a graphical model — a dense bipartite graph in which every measurement y_m touches every unknown signal entry x_n; literal BP would pass $2MN$ messages, but dense-and-weak is exactly the collapsing regime. Right: the collapsed algorithm (Donoho, Maleki, and Montanari 2009) — an estimate step through a denoiser η_t, and a residual step carrying the *Onsager correction* (red): the memory term that subtracts the estimate's own influence reflected back through the measurement matrix, keeps the effective noise Gaussian, and makes the algorithm's state-evolution analysis exact. The reaction term of Chapter 11 in signal-processing form — you derive it in the tutorial.

Take-home 3

AMP is belief propagation for the dense compressed-sensing graph, collapsed by the CLT to vector updates — and its residual carries an **Onsager correction term** that decorrelates the effective noise and makes state evolution exact. Drop it and the iteration fails; keep it and the algorithm has a proof, and a sharp recovery *phase transition* in the measurement rate. It is TAP for another graphical model.

12.6 Example: exact on the tree, wobbling on the loop, TAP on the dense

The three claims are tested numerically on three graphs (Figure 12.4). **The tree:** a seven-node Ising tree with random couplings and fields — BP converges in a handful of sweeps and matches the enumerated marginals to 4×10^{-15}: machine precision,

the exactness theorem realized numerically. **The loop:** add *one* edge to that tree, closing a single cycle, and rerun. BP still converges here, but the marginals are now wrong — by 0.03 at weak coupling, growing to 0.65 as the loop's couplings strengthen — while the tree's error stays at machine zero throughout. One edge is the entire difference, and the mechanism is the one the trap named: around a loop, a node's influence returns as fake evidence, and the stronger the loop's couplings, the louder the echo. **The dense graph:** run full BP on the previous lecture's SK instance ($N = 10$, the seed and fields of Section 11.5, $T = 1.5$) and set it beside the previous lecture's TAP solution: the two agree to 0.02 across all ten spins — the BP→TAP reduction, verified on the very instance the previous lecture used to test TAP (both, for the record, within 0.03 RMS of the enumerated truth).

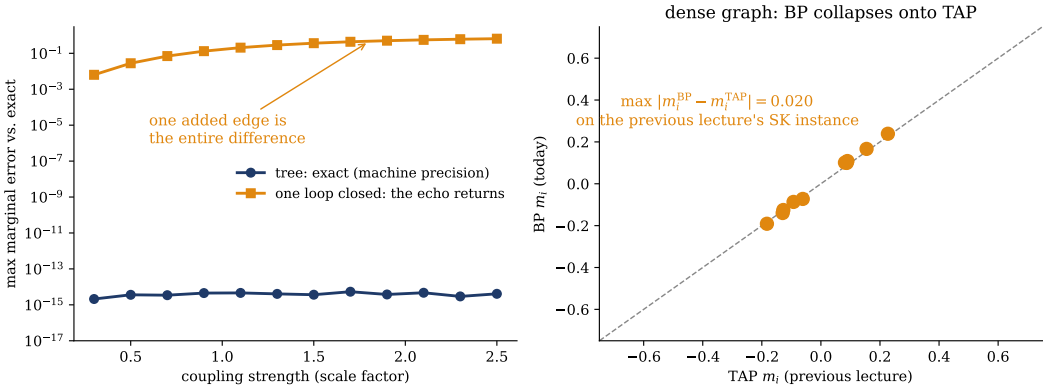

Figure 12.4: The lecture's three claims, run. Left: maximum marginal error of BP against exact enumeration, versus the strength of the couplings, on a seven-node tree (navy — flat at machine precision, $\sim 10^{-15}$: the exactness theorem) and on the same graph with *one* added edge closing a single loop (orange — the error grows from 0.03 to 0.65 as the loop's couplings strengthen: influence returning around the cycle, the self-reaction trees lack). Right: on the dense graph, BP collapses onto TAP — the BP marginals of the previous lecture's $N = 10$ SK instance (Section 11.5, same seed, $T = 1.5$) against the previous lecture's TAP solution, on the diagonal to within 0.02. One idea, three graphs: exact where removal disconnects, echoing where it does not, and TAP where the graph is dense.

> *The cavity field is a message; passed on a tree it is exact, passed on a dense graph it is TAP — message passing is mean-field plus Onsager.*

The tutorial (16–18, Ph12 106) is derivation-led. Two problems, both applications of today's machinery: run belief propagation on a specific Ising tree by hand and check it against the exact answer, then derive approximate message passing for compressed sensing from the collapsing messages.

12.7 Outlook: the week's thesis, complete

Week 6 closes with a single idea spanning four apparently different objects. The previous lecture derived one term — the Onsager reaction, a spin's self-echo subtracted from its effective field — twice, and found its size governed by $\sum_j J_{ij}^2$. Today the

same cavity idea, run recursively, became an algorithm: exact on trees, variational through the Bethe free energy, and collapsing — on dense graphs — back onto the previous lecture's equations, the messages' correction *being* the reaction term. In the tutorial it becomes a second algorithm, on a second graph, with the same term reappearing in a residual update and earning, this time, an exactness proof. Mean field, TAP, belief propagation, and AMP are one idea at four levels of graph structure, with a progressively weaker set of guarantees — a bound (W5), an estimate (TAP), tree-exactness (BP), asymptotic exactness with state evolution (AMP) — each relaxed in exchange for broader applicability.

The two routes through disordered systems can now be compared. Week 2 introduced the replica method's results; The previous lecture presented the cavity route as the alternative. The cavity route is the one that became *algorithms*: belief propagation decodes the LDPC codes in every modern radio (Gallager 1962), and AMP reconstructs undersampled signals with performance guarantees (Donoho, Maleki, and Montanari 2009). The cavity method is how disordered-systems physics entered algorithm design.

The module now turns toward machine learning proper. Week 7 takes the variational thread to its ML destination: latent-variable models, the auxiliary-field (Hubbard–Stratonovich) trick for manufacturing latents, and the **variational autoencoder** — where Week 5's ELBO stops being a physicist's identity and becomes the literal training loss of a deep generative network, with Week 5's reverse-KL warnings manifesting as posterior collapse. After that, Module 3 finally takes the other road out of the intractable Z — not approximating the distribution but *sampling* it — and the two roads will meet again in Module 4, where Z is neither approximated nor sampled but removed from the objective altogether.

13 Hubbard–Stratonovich: interactions become latent variables

Recommended reading

Core (~1 h). Mézard and Montanari (2009), the Curie–Weiss treatment of Ch. 2 once more — this time for the auxiliary-field route: the transformation and the saddle-point recovery of mean field, in the notation this lecture follows. Alongside it, Mehta et al. (2019), the sections on latent-variable and generative models — the machine-learning framing of $p(x) = \int \mathrm{d}z\, p(z)\, p(x \mid z)$ that today's reframe makes precise.
Optional background. Stratonovich (1957) and Hubbard (1959) — the transformation's original two papers, each about a page; skim for the Gaussian identity and its use on a quadratic interaction. Dempster, Laird, and Rubin (1977) — the EM algorithm, the classical route to learning a latent-variable model and the ancestor of the next lecture's training objective; MacKay (2003), Ch. 33–34, for latent variables and variational methods on the inference side.
Prerequisite reminder. The Gaussian integral Equation 2.1 and Laplace's method Equation 2.2 from Week 1 — today they carry the entire lecture; the Curie–Weiss ferromagnet and its self-consistent equation from Week 5 (Equation 10.1 and the landscape of Section 10.3); and the ELBO identity Equation 9.2 from Section 9.4. New content: the Hubbard–Stratonovich transformation, mean field as a saddle point, the reading of the auxiliary field as a latent variable, and the linear-Gaussian model as the template's one solvable instance.

13.1 Remove the interaction, don't approximate it

Module 2 has so far worked around the intractable interacting distribution. Week 5 bounded it, restricting the variational family to factorized distributions and accepting the gap Equation 9.1 as the price; Week 6 corrected the bound's blindness with the reaction term of Chapter 11 and then reorganized the whole computation into messages on a graph (Chapter 12). All three methods *live with* the interaction and manage its consequences. Today we do something categorically different: we remove the interaction — exactly. A one-line Gaussian identity trades the coupling between spins for a coupling of each spin to a single shared auxiliary field, so that the interacting problem becomes an independent-spin problem at the price of one ordinary integral. No approximation is made anywhere.

The identity is the **Hubbard–Stratonovich transformation**, and its history runs through two short papers a world apart: Stratonovich (1957), in the Soviet literature on quantum distribution functions, and Hubbard (1959), who used it to rewrite partition functions of interacting systems as functional integrals over auxiliary fields

— the move that became the daily workhorse of quantum field theory, where "integrating in a field" is how interacting particles become field theories in the first place. Machine learning, meanwhile, built the same structure from the opposite shore without knowing it: mixture models, factor analysis, and the EM algorithm of Dempster, Laird, and Rubin (1977) all posit a hidden variable z whose value renders the observations independent, and fit it to data. Today's lecture proves that these are one construction. The auxiliary field that decouples an interacting system *is* a latent variable, and a latent-variable generative model *is* an interacting system with the interaction integrated out.

The driving question follows: *what is the auxiliary field, really?* It enters as a bare integration variable — introduced by hand, with no physical meaning of its own. Yet by the end of Section 13.3 its saddle-point value will be the magnetization, Week 5's order parameter, satisfying Week 5's equation; and by the end of Section 13.4 the field itself will be the latent variable of a generative model, with a prior, a conditional, and a graphical model attached. One object serves two communities, and the transformation is the dictionary between them. Where Week 5 derived mean field as a variational bound, today derives the same equation as the *leading order of something exact* — which is why mean field is simultaneously an approximation and the beginning of a systematic truth, and why its corrections were computable at all.

> Take-home 1
>
> The Hubbard–Stratonovich transformation removes an interaction exactly: interacting spins become independent spins in a single shared auxiliary field, at the cost of one integral. That auxiliary field is both the physicist's order parameter (its saddle value is the magnetization) and the machine-learner's latent variable (the hidden common cause of the observations) — the same object, seen from two shores.

Figure 13.1 shows the central picture: a fully connected graph of interacting spins collapses to a *star*, every spin attached only to a central auxiliary node z, with no spin–spin edge surviving. The left half is Week 5's problem; the right half is, literally, the graphical model of a latent-variable generative model. The mathematics below is the proof that the arrow between them is an equality.

13.2 The engine: decoupling by a Gaussian

Setup. We work on the model where the transformation is cleanest: the Curie–Weiss ferromagnet of Section 10.2, fully connected with $J_{ij} = J/N$ and $J > 0$, in a field h. Its Hamiltonian is a perfect square,

$$H = -\frac{J}{2N}\Big(\sum_i s_i\Big)^2 - h\sum_i s_i, \qquad \text{so} \qquad Z = \sum_{\{s\}} \exp\left[\frac{\beta J}{2N}\Big(\sum_i s_i\Big)^2 + \beta h \sum_i s_i\right].$$

The obstacle is the square: expanded, $(\sum_i s_i)^2 = \sum_{ij} s_i s_j$ couples every spin to every other, and the sum over $\{s\}$ does not factorize. Everything Week 5 and Week

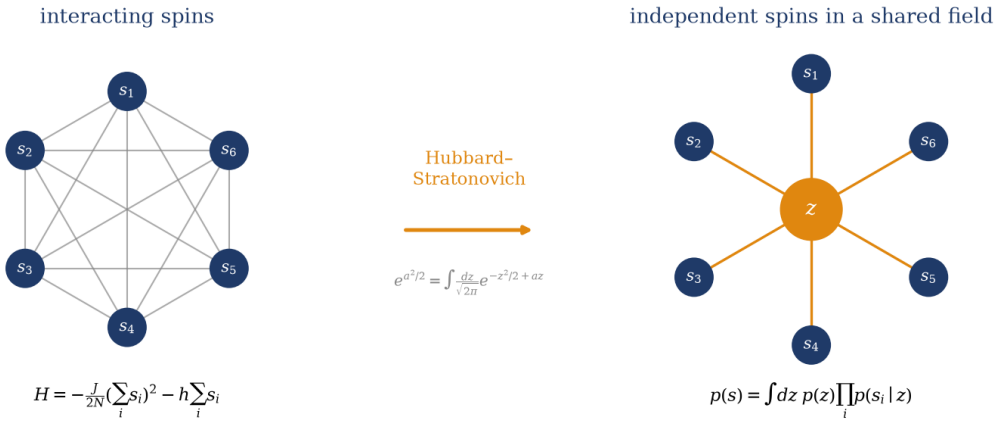

Figure 13.1: The lecture in one picture. Left: the Curie–Weiss ferromagnet — every spin coupled to every other through $(\sum_i s_i)^2$, the all-to-all interaction that makes Z a sum over 2^N coupled configurations. Right: the same model after the Hubbard–Stratonovich transformation — each spin couples *linearly and independently* to one shared auxiliary field z, and no spin–spin edge remains. The star graph is exactly the graphical model of a latent-variable generative model: observations conditionally independent given a hidden common cause.

6 did was a strategy for living with that square. The question is: *can a square be traded for something linear?*

The identity. It can, and the tool is one we have carried since Week 1. Complete the square in the exponent of a Gaussian integral: $-\frac{z^2}{2} + az = -\frac{(z-a)^2}{2} + \frac{a^2}{2}$, so by Equation 2.1 (shift $z \to z + a$, which sweeps out the same real line),

$$\boxed{e^{a^2/2} \;=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{d}z\; e^{-z^2/2 + az}} \tag{13.1}$$

Hubbard–Stratonovich identity

Read it as a trade: on the left, the *square* of a quantity a; on the right, a appears only *linearly*, coupled to a new integration variable z that is averaged over a Gaussian. If a is a sum over spins, its square couples them all — but a linear term does not couple anything. The identity converts an interaction into an average over fields, exactly. (This is Week 1's Gaussian integral run in reverse: instead of evaluating a Gaussian, we *introduce* one to linearize an exponent. The same completed square, read right to left.)

Apply it to the interaction. Set $a = \sqrt{\beta J/N} \sum_i s_i$ in Equation 13.1, so that $a^2/2$ is precisely the interaction term of Z. The exchange factor becomes an integral over z in which the spins enter linearly; rescaling the integration variable, $z \to \sqrt{N\beta J}\, z$, puts the result in its physical form:

$$\exp\left[\frac{\beta J}{2N}\Big(\sum_i s_i\Big)^2\right] \;=\; \sqrt{\frac{N\beta J}{2\pi}} \int \mathrm{d}z\; e^{-N\beta J z^2/2}\; \exp\Big[\beta J z \sum_i s_i\Big].$$

Insert this into Z and exchange the sum and the integral. Under the integral, each spin sees only the field $Jz + h$ — the auxiliary field plus the external one — and *no other spin*:

$$Z \;=\; \sqrt{\frac{N\beta J}{2\pi}} \int \mathrm{d}z\; e^{-N\beta J z^2/2} \sum_{\{s\}} \prod_i e^{\beta(Jz+h)\, s_i}.$$

The spin sum now factorizes — this is the decoupling. Each factor is Week 1's two-state spin in a field, $\sum_{s_i=\pm 1} e^{\beta(Jz+h)s_i} = 2\cosh(\beta(Jz+h))$, so the 2^N-term interacting sum collapses to the N-th power of a single-site object:

$$Z \;=\; \sqrt{\frac{N\beta J}{2\pi}} \int \mathrm{d}z\; e^{-N\beta f(z)}, \qquad f(z) \;=\; \frac{J}{2}\, z^2 \;-\; \frac{1}{\beta} \ln\!\left[2\cosh(\beta(Jz+h))\right] \tag{13.2}$$

decoupled partition function

The interpretation is direct: an N-body sum over 2^N coupled configurations has become a *single one-dimensional integral*, exactly, and the cost of removing the interaction is one dimension of integration regardless of how many spins participate. The exponent $f(z)$ is a free energy per spin as a function of the auxiliary field — Gaussian cost of the field, minus the free energy that N independent spins gain by living in it.

Two remarks before we evaluate anything. First, the identity required a *positive* coefficient on the square: for $J > 0$, the interaction is $+a^2/2$ and Equation 13.1 applies as written. A general coupling matrix J_{ij} needs its eigendecomposition $J = \sum_\mu \lambda_\mu v_\mu v_\mu^{\mathsf{T}}$, which splits the interaction into independent squares — one Hubbard–Stratonovich field per eigenvalue, real fields for $\lambda_\mu > 0$ and *imaginary* ones (a rotated integration contour) for $\lambda_\mu < 0$. Curie–Weiss is the rank-one case, $J_{ij} = (J/N)\,\mathbf{1}\mathbf{1}^{\mathsf{T}}$, which is why one field suffices; the general count returns in Section 13.4 with a machine-learning meaning. Second, nothing has been approximated: Equation 13.2 is an identity, valid at every N and every temperature. What follows next is where the approximation *chooses to enter* — and where Week 5 falls out.

13.3 The saddle point is mean field

The decoupled partition function Equation 13.2 has exactly the shape that Week 1 taught us to read: an integral of $e^{-N\beta f(z)}$ with N large in the exponent. Laplace's method Equation 2.2 applies verbatim — as $N \to \infty$ the integral is dominated by the minimum of f, and the free energy per spin converges to $f(z^*)$ where $f'(z^*) = 0$. Differentiate the exponent of Equation 13.2:

$$f'(z) \;=\; Jz \;-\; J\tanh(\beta(Jz+h)) \;\stackrel{!}{=}\; 0,$$

and divide by J:

$$z^* \;=\; \tanh\!\big(\beta(Jz^*+h)\big) \tag{13.3}$$

saddle point = mean-field equation

This is Week 5's self-consistent equation, symbol for symbol — compare Equation 10.1 and the graphical solution of Section 10.2, with $J_0 = J$ for the fully connected model. But look at the route: no variational family was chosen, no bound was invoked, and nothing was discarded. The equation emerged as the *stationarity condition of an exact integral*. Mean-field theory is the saddle point of the

Hubbard–Stratonovich representation, and the mean-field free energy is the leading term of an exact large-N evaluation.

The saddle value is the magnetization. The identification $z^* = m$ is forced by the structure of Equation 13.2, not merely suggested by the matching equation. Differentiate $\ln Z$ with respect to h under the integral: $m = \frac{1}{N\beta}\,\partial_h \ln Z = \langle \tanh(\beta(Jz+h)) \rangle$, the average taken over z with weight $e^{-N\beta f(z)}$. As $N \to \infty$ that weight concentrates on z^*, so $m \to \tanh(\beta(Jz^*+h)) = z^*$ by Equation 13.3 itself. The integration variable we introduced by hand acquires, *at its saddle*, the meaning of the order parameter — off the saddle it remains what it always was, a dummy variable being integrated over. One further identity closes the loop with Week 5: evaluating f at the saddle and using $S(m) = \ln[2\cosh(\beta(Jm+h))] - \beta(Jm+h)\,m$ for $m = \tanh(\beta(Jm+h))$, one finds $f(z^*) = -\frac{J}{2}m^2 - hm - T\,S(m)$ — exactly the variational free energy per spin that Section 10.3 minimized. The two functions $f(z)$ and Week 5's landscape differ away from their stationary points, but at the physical solution they coincide, value and all.

So the same equation now has two derivations, and Figure 13.2 records the fork. From the left, mean field is a *variational upper bound*: the exact functional evaluated on a restricted family, with the one-sided guarantee $F_q \geq F$ of Equation 9.1. From the right, it is a *saddle point*: the leading order of an exact integral, with no guarantee but with a systematic sequence of corrections attached. The two views are complementary in precisely the way Week 6 needed: it is because mean field is the leading order of something exact that its corrections (the fluctuations the bound discards) are computable rather than mysterious.

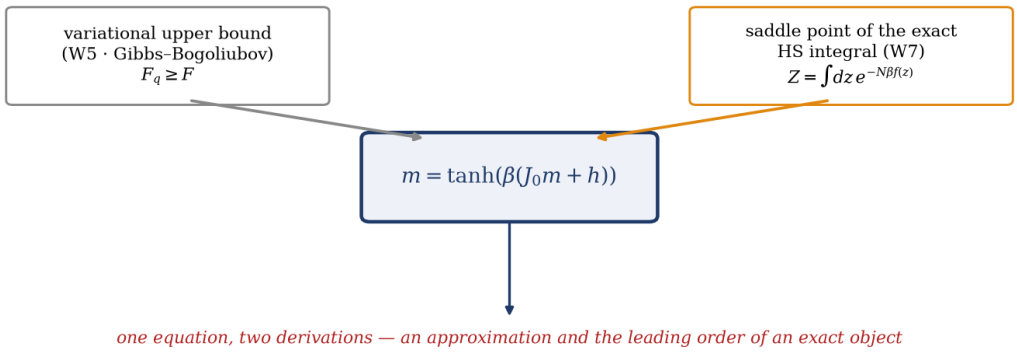

Figure 13.2: Two derivations, one equation. Week 5 reached $m = \tanh(\beta(J_0 m + h))$ by restricting the variational family and minimizing the Gibbs–Bogoliubov bound; today reaches it as the stationary point of the exact Hubbard–Stratonovich integral. The bound explains why mean field is an approximation with a guaranteed sign; the saddle explains why it is the leading order of a systematic expansion, with corrections that are the auxiliary field's fluctuations.

Corrections are the field's fluctuations. Laplace's method carries its own error estimate, and here the error has a name. Expanding f to second order around the saddle and using Equation 2.2, the prefactors combine into

$$Z \;\approx\; e^{-N\beta f(z^*)}\sqrt{\frac{J}{f''(z^*)}}, \qquad f''(z^*) \;=\; J[1-\beta J(1-z^{*2})] \;=\; \frac{\beta J(1-z^{*2})}{\chi},$$

where $\chi = \partial m/\partial h$ is the susceptibility of the model, obtained by differentiating Equation 13.3. The Gaussian width of the auxiliary field around its saddle is therefore set by $1/f''$ — by the susceptibility. Far from the transition, f'' is of order one, the field barely fluctuates, and the corrections to mean field are $\mathcal{O}(1/N)$; as $T \to T_c$ the susceptibility diverges (the Curie–Weiss law of Section 10.3), $f''(z^*) \to 0$, the integrand flattens, and the fluctuations of the auxiliary field grow without bound. This is Week 5's audit and Week 6's correction program, re-derived from one integral: *mean field is exact when the field cannot fluctuate* — the fully connected model as $N \to \infty$, where the saddle is infinitely sharp — *and fails where the field fluctuates strongly*, near criticality. What the variational route called "discarded correlations" the saddle route calls "fluctuations of z": the same physics, carried by f''.

Take-home 2

Evaluating the exact integral Equation 13.2 by Laplace's method gives $z^* = \tanh(\beta(Jz^* + h))$ — Week 5's mean-field equation, with z^* equal to the magnetization and $f(z^*)$ equal to the variational free energy at its optimum. Mean field is the saddle point of an exact representation; its corrections are the Gaussian fluctuations of the auxiliary field, of size $1/f''(z^*) \propto \chi$ — small at $\mathcal{O}(1/N)$ generically, divergent at the critical point.

13.4 The auxiliary field is a latent variable

Everything so far *evaluated* the integral. The reframe that closes Module 2 comes from refusing to: instead of integrating over z, read the integrand of Equation 13.2 as a probability distribution. The object under the integral, before the spin sum is performed, is a perfectly well-formed *joint* distribution over the spins and the field together,

$$p(s, z) \;=\; \frac{1}{Z}\sqrt{\frac{N\beta J}{2\pi}}\;\exp\Big[-\frac{N\beta J}{2}z^2 + \beta(Jz+h)\sum_i s_i\Big],$$

whose marginal over z is, by construction, exactly the Curie–Weiss Boltzmann distribution — that is what Equation 13.2 asserts. Now factor this joint the other way. Conditioned on z, the spins are independent, each governed by the normalized single-site factor

$$p(s_i \mid z) \;=\; \frac{e^{\beta(Jz+h)s_i}}{2\cosh(\beta(Jz+h))} \;=\; \sigma\big(2\beta(Jz+h)\,s_i\big),$$

a *sigmoid* $\sigma(u) = 1/(1+e^{-u})$ in the field — the logistic unit of every neural network, appearing here as the exact conditional of a spin given the auxiliary field. (The symbol σ is the logistic function throughout this lecture, unrelated to the encoder standard deviation σ_ϕ of the next lecture's VAE.) And the marginal of the field itself, obtained by summing the joint over the spins, is $p(z) \propto e^{-N\beta f(z)}$: the very integrand whose saddle we just took, now in the role of a *prior* over z. Assembling the two factors,

$$p(s) \;=\; \int \mathrm{d}z\; p(z) \prod_i p(s_i \mid z), \qquad p(z) \propto e^{-N\beta f(z)}, \qquad p(s_i \mid z) = \sigma(2\beta(Jz + h)\, s_i) \tag{13.4}$$

the latent-variable reframe

This is a **latent-variable generative model**, and it is the Curie–Weiss ferromagnet with no approximation: draw a field z from its prior, then draw each spin independently from a sigmoid conditioned on z. The recipe even samples correctly through the phase transition — below T_c the prior $e^{-N\beta f(z)}$ concentrates on the two wells $\pm m$ of Section 10.3, so drawing z first amounts to choosing a symmetry-broken branch, after which the spins fall independently in line. The interaction has become a *hidden common cause*: spins look correlated to anyone who cannot see z, and become independent the moment z is revealed. The star graph of Figure 13.1 was the claim; Equation 13.4 is the proof. (A bookkeeping remark for the careful reader: one may equally keep the bare Gaussian weight $e^{-N\beta J z^2/2}$ as the prior, at the price of unnormalized conditionals — normalizing the sigmoids is what moves the factor $[2\cosh]^N$ into $p(z)$. Both factorizations describe the same joint; Equation 13.4 is the one in which every factor is a distribution.)

The general model. Strip the physics and the structure that remains is

$$p(x) \;=\; \int \mathrm{d}z\; p(z)\, p(x \mid z),$$

a prior over a hidden variable and a conditional — machine learning calls it the *decoder* — over the observations. This one template covers a remarkable fraction of generative modeling (Figure 13.3): a *Gaussian mixture* is a discrete z with one Gaussian per component; *factor analysis* and *probabilistic PCA* are a Gaussian z with a linear-Gaussian decoder; the *variational autoencoder* of the next lecture (Kingma and Welling 2014) is a Gaussian z with a neural-network decoder. The instances differ only in the two distributions plugged into the template; the conditional-independence structure — the star — is common to all of them. And the physics supplies what the template alone does not: an answer to *where the latent comes from* and *how many dimensions it should have*. The eigendecomposition remark of Section 13.2 says that a general interaction $J = \sum_\mu \lambda_\mu v_\mu v_\mu^\mathsf{T}$ decouples into one field per eigenvalue: the latent dimension counts the independent modes of the interaction. Curie–Weiss is rank one and needs a single latent; data whose correlations have many strong modes need as many decoupling fields — which is, from this shore, why the latent spaces of real generative models are high-dimensional.

The solvable instance: a linear-Gaussian decoder. One member of the zoo is worked out in full, because it is the only one whose inference problem closes in a formula — and because it turns the eigenmode count above into numbers (Welling, Lu, and Holdijk 2026, ch. 4 on latent-variable models). Take a Gaussian latent and a linear decoder with isotropic noise,

$$p(z) = \mathcal{N}(0, \mathbb{1}_K), \qquad p(x \mid z) = \mathcal{N}(Wz,\, \sigma^2 \mathbb{1}),$$

so that a sample is assembled as $x = Wz + \sigma\,\varepsilon$: K hidden modes, mixed by the columns of W, plus noise — factor analysis with equal noise variances, better known as *probabilistic PCA* (Tipping and Bishop 1999). Everything the general template leaves intractable is here explicit. The joint log-density is quadratic in z, so the

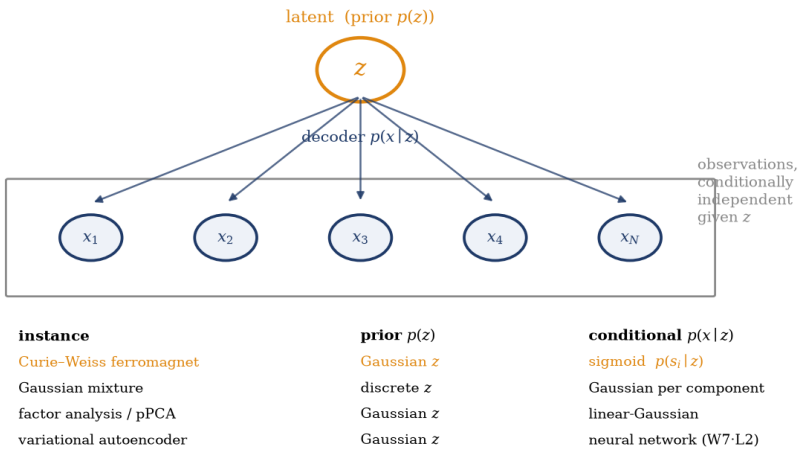

Figure 13.3: One structure, many models. The latent-variable graphical model — a hidden z with conditionally independent observations $x_1, \dots, x_N$ — realized by different choices of prior and conditional. The Curie–Weiss ferromagnet, decoupled by Hubbard–Stratonovich, is an exact instance with a one-dimensional latent; the variational autoencoder (the next lecture) is the same structure with a neural decoder.

posterior is Gaussian and completing the square — the same move that proved Equation 13.1 — is the entire computation:

$$\log p(z \mid x) \;=\; -\frac{1}{2\sigma^2}\|x - Wz\|^2 - \frac{1}{2}\|z\|^2 + \text{const} \;=\; -\frac{1}{2}\, z^{\mathsf{T}}\Big(\mathbb{1} + \frac{W^{\mathsf{T}}W}{\sigma^2}\Big) z + \frac{1}{\sigma^2}\, z^{\mathsf{T}} W^{\mathsf{T}} x + \text{const},$$

whence

$$p(z \mid x) = \mathcal{N}(\mu(x),\, C), \qquad C = \Big(\mathbb{1} + \frac{W^{\mathsf{T}}W}{\sigma^2}\Big)^{-1}, \qquad \mu(x) = \frac{1}{\sigma^2}\, C\, W^{\mathsf{T}} x.$$

the one closed-form posterior in the zoo

(The evidence follows just as cheaply: x is a linear map of jointly Gaussian variables, so $p(x) = \mathcal{N}(0,\, WW^{\mathsf{T}} + \sigma^2 \mathbb{1})$ — the marginal likelihood, in closed form, for this one model.) Read the posterior along the decoder's own axes and the eigenmode claim becomes quantitative. Take the columns of W orthogonal, $W_\mu = w_\mu u_\mu$ with orthonormal u_μ: then $W^{\mathsf{T}}W$ is diagonal, the posterior factorizes over modes, and each latent coordinate is a *shrunken projection* of the data onto its mode,

$$\mu_\mu(x) \;=\; \frac{w_\mu^2}{w_\mu^2 + \sigma^2}\, \frac{u_\mu^{\mathsf{T}} x}{w_\mu}, \qquad C_{\mu\mu} \;=\; \frac{\sigma^2}{w_\mu^2 + \sigma^2}.$$

The bracket in front is a per-mode signal-to-noise weight: a strong mode ($w_\mu^2 \gg \sigma^2$) is read off the data essentially verbatim, while a weak one ($w_\mu^2 \ll \sigma^2$) is shrunk toward the prior's zero, its posterior variance staying close to the prior's one — the inference is effectively suppressed where noise dominates. The claim of Section 13.2 — that the latent dimension counts the interaction's *strong* eigenmodes — is here a formula rather than a principle. The instance matters for the module because of what happens when it breaks: replace Wz by a neural network and both closed forms

vanish, since no conjugacy survives the first nonlinearity. That loss is exactly the intractability the next lecture's encoder is trained to patch — the VAE's variational posterior $q_\phi(z \mid x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$ keeps the *shape* of the formula above and learns the map $x \mapsto (\mu, \sigma)$ that the linear model derives.

Learning the model, and the loop back to Week 5. The physics handed us $p(z)$ and $p(x \mid z)$ complete, because we knew the Hamiltonian. Machine learning faces the inverse task: given samples of x and a parameterized decoder $p_\theta(x \mid z)$, fit θ by maximizing the *evidence* $\log p_\theta(x) = \log \int \mathrm{d}z\, p(z)\, p_\theta(x \mid z)$ — and both the integral and the posterior $p(z \mid x)$ it hides are intractable, for the same reason Z always was. The resolution is the one this module has rehearsed since Week 5: maximize a lower bound, the ELBO, which Equation 9.2 identified as minus the variational free energy, with the posterior gap $D_{\mathrm{KL}}(q(z) \| p(z \mid x))$ as the exact deficit. The classical algorithm built on this bound is EM (Dempster, Laird, and Rubin 1977): the E-step sets q to the current posterior (closing the gap), the M-step maximizes $\langle \log p_\theta(x, z) \rangle_q$ over θ (raising the bound) — coordinate ascent on the ELBO, forty years before that name existed. What the next lecture adds is scale: when the posterior is not computable even per data point, a neural *encoder* learns to approximate it for all x at once, and the result is the variational autoencoder. The physics that creates latent variables (Hubbard–Stratonovich) and the inference that learns them (the ELBO) are now both on the table; only the neural realization remains.

Trap

The auxiliary field is not physical — but its saddle is. Under the integral, z ranges over all real values and carries no physical meaning; attributing reality to its fluctuating value is a category error. Only the *saddle* z^* equals the magnetization, and only as $N \to \infty$. Two corollaries follow. ① Mean field is the saddle, not the integral: equating them silently drops the fluctuation corrections, which is precisely the error Weeks 5–6 spent two lectures diagnosing. ② The transformation's simple real form needs a positive-definite interaction; applying Equation 13.1 blindly to a coupling with negative eigenvalues produces a divergent Gaussian — those modes need imaginary fields, and the bookkeeping is not optional.

Take-home 3

Read rather than evaluated, the Hubbard–Stratonovich integrand is a latent-variable model: $p(s) = \int \mathrm{d}z\, p(z) \prod_i p(s_i \mid z)$, with the integrand-as-prior $p(z) \propto e^{-N\beta f(z)}$ and sigmoid conditionals — the Curie–Weiss ferromagnet exactly. The general template $p(x) = \int \mathrm{d}z\, p(z)\, p(x \mid z)$ covers mixtures, factor analysis, and the VAE; the latent dimension counts the interaction's eigenmodes; and learning the model is Week 5's ELBO again, with EM as its classical coordinate ascent.

13.5 Example: the saddle against the truth

The two claims — the integral is exact, and mean field is only its saddle — can be tested against each other on one small model. Take Curie–Weiss at $J = 1$ (so $T_c = 1$), $h = 0$, and $T = 0.7$, comfortably in the ordered phase. The saddle

equation Equation 13.3 gives $z^* = 0.8286$ with $f(z^*) = -0.5480$ and curvature $f''(z^*) = 0.552$; the exact free energy per spin comes from the magnetization-sector sum of Section 10.5, affordable at any N. Figure 13.4 draws the two halves of the story: the exponent $f(z)$ developing the double well of Section 10.3 as T crosses T_c — the same landscape, now living on the auxiliary field — and the integrand $e^{-N\beta f(z)}$ sharpening onto the saddles as N grows.

N	f_{exact}	f_{saddle}	gap	saddle + fluctuation	residual
5	-0.6819	-0.5480	0.134	-0.6866	0.0047
20	-0.5859	-0.5480	0.038	-0.5827	0.0032
100	-0.5551	-0.5480	0.0070	-0.5550	0.0001
1000	-0.54872	-0.54802	0.00069	-0.54871	5×10^{-6}

The columns confirm the analysis. The **gap column** shrinks in proportion to $1/N$ — tenfold between $N = 100$ and $N = 1000$: the saddle approximation converges to the truth as $1/N$, which is the precise sense in which mean field is exact for the fully connected model — the verdict of Section 10.4, now visible inside a single integral. The **last two columns** identify the gap completely: adding the Gaussian-fluctuation prefactor of Section 13.3, together with a factor of 2 because *both* wells contribute below T_c, predicts the free energy as $f(z^*) - \frac{T}{N}\ln(2\sqrt{J/f''(z^*)})$, and this lands within 10^{-4} of the exact answer already at $N = 100$. The correction to mean field is the width of the peak (and the number of peaks), and the formula is known — the same lesson Section 2.6 taught for Stirling's series, returning at the end of the module.

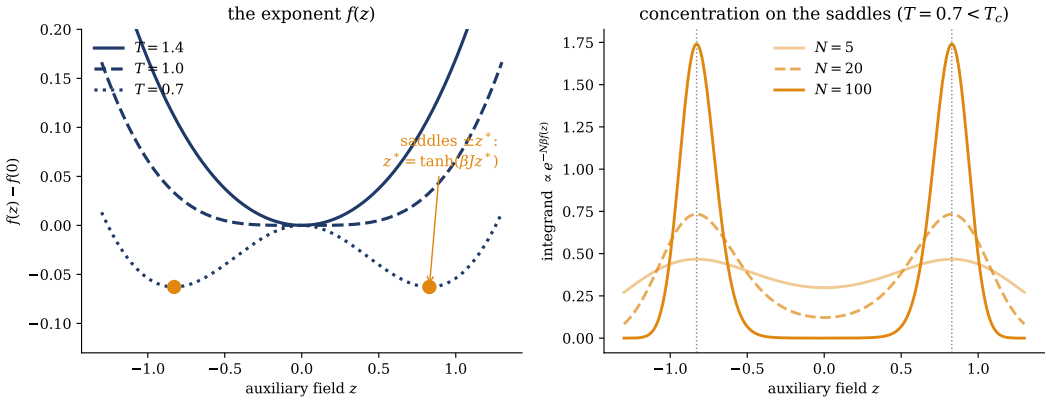

Figure 13.4: The exact partition function as one integral, concentrating on mean field ($J = 1$, $h = 0$, so $T_c = 1$). Left: the exponent $f(z)$ of Equation 13.2 at three temperatures, drawn relative to $f(0)$ — a single well at $z = 0$ above T_c, a flattening at T_c, and the double well of Section 10.3 below it, now as a function of the auxiliary field; the orange dots mark the saddles $\pm z^*$ from Equation 13.3 ($z^* = 0.83$ at $T = 0.7$). Right: the normalized integrand $e^{-N\beta f(z)}$ at $T = 0.7$ for growing N — the weight concentrates onto the saddles, with width shrinking as $1/\sqrt{N\beta f''(z^*)}$, and Laplace's method Equation 2.2 becomes exact. Mean field is the $N \to \infty$ limit of this picture; everything it misses is the remaining width.

One consistency check ties the module together. The double well in the left panel

of Figure 13.4 and the double well of Figure 10.2 are drawn from *different functions* — Week 5's landscape lives on the trial magnetization, today's on the auxiliary field — yet their minima sit at the same $\pm m$ with the same depth, as Section 13.3 proved they must. The phase transition of Section 10.2 can therefore be read a third way: it is the moment the exact integrand of Z changes from one peak to two, and spontaneous symmetry breaking is the statement that as $N \to \infty$ the integral is captured entirely by whichever peak the field $h \to 0^+$ selects.

> *A latent variable is a decoupling field — a generative model is an interacting system with the interaction integrated out.*

The week continues from here. The evening before the next lecture: the warm-up card, on the ELBO and reparameterization sections of the VAE review by Kingma and Welling (2019) — every object in that paper now has a physical name: the prior is a decoupling field's distribution, the decoder is the conditional the field induces, and the training objective is Week 5's variational free energy with its sign flipped. In the next lecture the model becomes trainable: neural encoder and decoder, the reparameterization trick that makes the latent integral differentiable, and the variational autoencoder in full (Kingma and Welling 2014). The tutorial is computation-led, training a VAE on MNIST and decomposing its ELBO term by term.

13.6 Outlook: Module 2 closes

Module 2 set out to tame one intractable object, the interacting Boltzmann distribution, and it left four methods on the table. Week 5 *bounded* it: the Gibbs–Bogoliubov inequality on a factorized family, with the ELBO as its inference-side name. Week 6 *corrected* the bound (the reaction term of Chapter 11) and then *reorganized* it into messages on the interaction graph (Chapter 12). Today *transformed* it: the interaction removed exactly, the system rewritten as a latent-variable generative model, and mean field re-derived as the saddle of the exact representation. The unifying object throughout has been a single functional — the variational free energy, equal to minus the ELBO by Equation 9.2 — optimized over different families, corrected by different terms, and returning in the next lecture as the training loss of the variational autoencoder. That a 1950s identity from quantum field theory (Stratonovich 1957; Hubbard 1959) and the objective of a 2014 deep generative model (Kingma and Welling 2014) belong to the same three-week story is a convergence worth emphasizing.

A broader principle generalizes beyond the ferromagnet: wherever a latent-variable model fits data well, an interacting system stands behind it, with the latent as its decoupling field and the latent's typical values as its order parameters. The correlations in data play the role the interaction played here; the latent dimensions count its strong eigenmodes; and inference over the latent is the physics of a field near its saddle. Machine learning rediscovered this structure independently, because marginalizing a shared cause and decoupling an interaction are the same mathematical act read in opposite directions.

Every method of this module — bound, correction, messages, transformation — manipulated the *form* of the distribution without visiting its configurations. Module 3, beginning in Week 8, takes the complementary approach and *samples*: Monte Carlo and the Metropolis rule, simulated annealing, and the Langevin dynamics

whose stationary measure is the Boltzmann distribution itself — the machinery that will let us treat stochastic gradient descent as a finite-temperature physical process. The distribution approximated over three weeks will now be sampled directly.

14 The variational autoencoder: minimizing free energy by backpropagation

Recommended reading

Core (~1 h). Kingma and Welling (2014) — the founding VAE paper, and a rare case of a field-launching result that fits in eight pages: the ELBO estimator, the amortized encoder, and the reparameterization trick, all first stated here; read §2–§3 for the ideas and let the lecture carry the derivations. Alongside it, the ELBO and reparameterization sections of Kingma and Welling (2019), the warm-up review, which retells the same construction with a decade of hindsight.
Optional background. Rezende, Mohamed, and Wierstra (2014) — the simultaneous, independent invention of the same model from the generative-model side; that two groups arrived at the reparameterized ELBO within months of each other says something about how ready the pieces were. Higgins et al. (2017) — the -VAE, which reweights the KL term and maps the tradeoff this lecture ends on. Dayan et al. (1995) — the Helmholtz machine, the recognition/generative-network ancestor, with Hinton in the author list once again. Mehta et al. (2019), the variational-autoencoder section, for the physicists' telling in notation close to ours.
Prerequisite reminder. Week 5 in full — the variational free energy and the identity $F_q = -\text{ELBO}$ (Equation 9.2), the dictionary of Table 9.1, and the reverse-KL mode-seeking warning (Figure 9.3); the previous lecture's latent-variable reframe Equation 13.4, $p(x) = \int \mathrm{d}z\, p(z)\, p(x \mid z)$, with the latent as a decoupling field (Section 13.4); and basic feed-forward networks trained by SGD, from the companion *Machine Learning and Physics* course. New content: amortized inference, the reparameterization trick, and the VAE as the trainable form of the variational free energy.

14.1 The objective we already have

Module 2 ends with a trainable model. The previous lecture closed the circle it had opened: a latent-variable model $p_\theta(x) = \int \mathrm{d}z\, p(z)\, p_\theta(x \mid z)$ is an interacting system with the interaction integrated out — the latent z is a decoupling field (Section 13.4) — and *learning* such a model runs immediately into the posterior $p_\theta(z \mid x)$, which is intractable for the same reason Z always was. Week 5 told us what to do about that: replace the posterior by a tractable q, and optimize the evidence lower bound, which is nothing but the negative variational free energy (Equation 9.2). So the objective is settled. What is not settled is whether anyone can *train* it. Between the ELBO on the blackboard and a generative model fitted to a million images stand two concrete obstacles, and today's lecture is the removal of both.

The first obstacle is bookkeeping that does not scale. Classical variational inference — and its ancestor, the EM algorithm — optimizes a *separate* variational distribution q for every data point: one free-standing optimization per image, re-run from scratch for any image not seen before. The second obstacle is a gradient that does not exist where we need it. The ELBO is an expectation over $q_\phi(z \mid x)$, and its parameters ϕ sit *inside the distribution being sampled*; back-propagation, which differentiates compositions of deterministic functions, cannot pass through the random draw of a z.

In 2014 both obstacles fell at once, twice. Kingma and Welling (2014) and, independently and nearly simultaneously, Rezende, Mohamed, and Wierstra (2014) proposed the same pair of fixes: replace the per-datapoint q by a single neural network — an *encoder* — that outputs the variational parameters for any x in one forward pass, and rewrite the random draw as a deterministic, differentiable function of the parameters plus external noise. The resulting model, the **variational autoencoder** (VAE), launched deep generative modeling as a field. Its lineage is worth a sentence: the idea of pairing a *recognition network* with a *generative network* and training them together is the *Helmholtz machine* of Dayan et al. (1995) — a name chosen, twenty years before this course, precisely because the objective being optimized is a free energy.

The driving question of the lecture: *how do you train Week 5's free-energy minimization on a dataset too large to enumerate and a model too deep to solve?* The answer occupies the two sections below; the closing sections read the trained object back in the language of Week 5.

> Take-home 1
>
> The ELBO ($= -F_{\text{var}}$, Week 5) is the right objective for a latent-variable model, but two obstacles block training it at scale: classical VI optimizes a fresh q per data point, and the gradient in the variational parameters cannot pass through a random sample. The VAE (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014) removes both — *amortized inference* (one encoder network for all data) and the *reparameterization trick* (a differentiable sampler) — turning free-energy minimization into a network trained by SGD.

14.2 Two obstacles, two fixes

The structure is a two-by-two correspondence, worth stating before any formula. Obstacle one, *per-datapoint inference*, is removed by **amortization**: a single network $q_\phi(z \mid x)$ learns the map from data to variational parameters, so that inference on a new x costs one forward pass rather than one optimization. Obstacle two, *non-differentiable sampling*, is removed by **reparameterization**: the sample z is rewritten as a deterministic function of ϕ and a parameter-free noise ε, so that gradients flow through the sampler like through any other layer. The first provides scale; the second provides gradients; together they make the ELBO an objective like any other in deep learning — evaluated on minibatches, differentiated by backprop, optimized by SGD.

Figure 14.1 assembles the machine these fixes produce, and everything that follows is a derivation of one of its parts. Data x enters the *encoder* $q_\phi(z \mid x)$, which outputs the mean $\mu_\phi(x)$ and standard deviation $\sigma_\phi(x)$ of a Gaussian over the latent;

a sample $z = \mu_\phi + \sigma_\phi \odot \varepsilon$ is drawn with external noise ε ($\odot$ is the elementwise product); the *decoder* $p_\theta(x \mid z)$ maps the latent back to a distribution over data space, scored against the input. Two terms tie the loop together: the reconstruction term compares the decoder's output to x, and a KL term holds the encoder's distribution near the prior $p(z) = \mathcal{N}(0, I)$. That KL term is what separates this machine from a plain autoencoder, and the distinction is the whole point: because every encoder output is pulled toward the *same* prior, a z drawn from that prior is a legitimate input to the decoder — the model can *generate*, not only compress. A deterministic autoencoder has no prior to sample from and cannot.

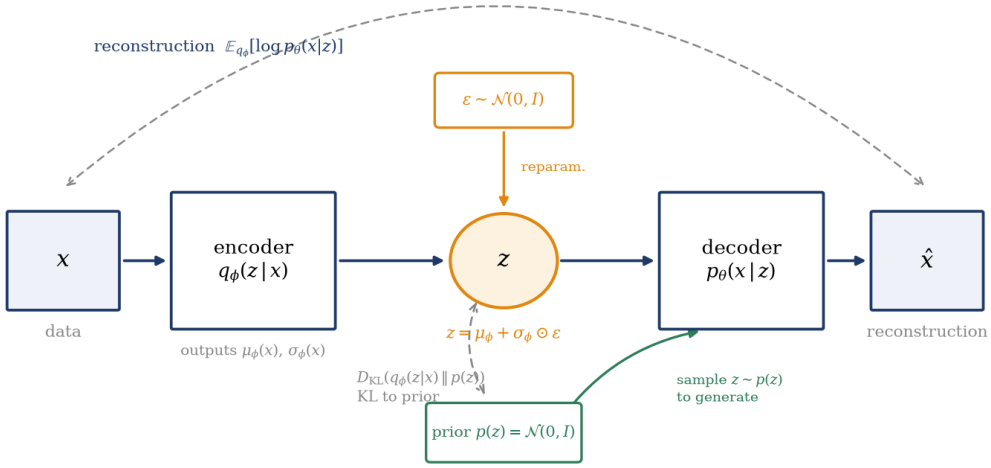

Figure 14.1: The central picture of the lecture: the variational autoencoder. Data x enters the encoder $q_\phi(z \mid x)$, which outputs the variational parameters $\mu_\phi(x), \sigma_\phi(x)$; the latent is sampled through the reparameterization $z = \mu_\phi + \sigma_\phi \odot \varepsilon$ with external noise $\varepsilon \sim \mathcal{N}(0, I)$ (orange), and the decoder $p_\theta(x \mid z)$ scores the reconstruction against the input. The two loss terms are drawn in gray: the reconstruction term (top) compares $\hat{x}$ to x; the KL term (bottom) holds the encoder near the prior $p(z) = \mathcal{N}(0, I)$. Generation (green) bypasses the encoder entirely: sample $z \sim p(z)$ and decode.

One notational flag before the derivations, because two parameter sets now live in one objective. Throughout, θ denotes the *decoder* (generative) parameters and ϕ the *encoder* (inference) parameters; the ELBO is optimized over both, but the two gradients raise entirely different issues — ∇_θ is routine, and ∇_ϕ is the subject of Section 14.4. The latent z is the previous lecture's auxiliary field (Chapter 13), here continuous and Gaussian; ε is the reparameterization noise, not an energy scale. And we work at $T = 1$ with the manufactured energy $E(z) = -\log p_\theta(x, z)$ of Section 9.4, so the machine-learning literature's $\mathbb{E}_{q_\phi}[\,\cdot\,]$ is our $\langle \cdot \rangle_{q_\phi}$; we keep the course's angle brackets.

14.3 The engine, part I: two terms, one encoder

The ELBO as Week 5 left it is one number — a lower bound on the log-evidence. To train a network we need it in the two pieces the network will trade off, and the

split takes four lines. Start from the master identity, Equation 9.2, specialized to an x-dependent variational family:

$$\log p_\theta(x) \;=\; \mathrm{ELBO}(\theta,\phi;x) + D_{\mathrm{KL}}(q_\phi(z\mid x)\,\|\,p_\theta(z\mid x)), \qquad \mathrm{ELBO} \;=\; \big\langle \log p_\theta(x,z)\big\rangle_{q_\phi} - \big\langle \log q_\phi(z\mid x)\big\rangle_{q_\phi}.$$

Since the gap on the right is a KL divergence, it is non-negative, and the ELBO bounds the evidence from below — that much is Week 5 verbatim. Now split the joint along the generative structure of Equation 13.4, $p_\theta(x,z) = p(z)\,p_\theta(x\mid z)$, and regroup:

$$\mathrm{ELBO} \;=\; \big\langle \log p_\theta(x\mid z)\big\rangle_{q_\phi} + \big\langle \log p(z)\big\rangle_{q_\phi} - \big\langle \log q_\phi(z\mid x)\big\rangle_{q_\phi} \;=\; \big\langle \log p_\theta(x\mid z)\big\rangle_{q_\phi} - \Big\langle \log \frac{q_\phi(z\mid x)}{p(z)}\Big\rangle_{q_\phi},$$

and the last average is again a KL divergence — this time between the encoder and the *prior*:

$$\boxed{\mathrm{ELBO}(\theta,\phi;x) \;=\; \underbrace{\big\langle \log p_\theta(x\mid z)\big\rangle_{q_\phi(z|x)}}_{\text{reconstruction}} \;-\; \underbrace{D_{\mathrm{KL}}(q_\phi(z\mid x)\,\|\,p(z))}_{\text{KL to prior}}} \tag{14.1}$$

ELBO decomposition

The two terms pull against each other, and their tension is the model. The **reconstruction** term is the log-likelihood the decoder assigns to the data given its inferred latent: it grows when z carries information about x, so it pushes the encoder to be *informative*. The **KL regularizer** measures how far the encoder strays from the prior: it shrinks when z carries nothing, so it pushes the encoder to be *forgettable*. A trained VAE sits where the two forces balance — enough structure in the latent to reconstruct, little enough to keep the latent space close to the prior it must share with generation. And by Week 5's identity, the whole expression is $-F_{\mathrm{var}}(q_\phi)$: reconstruction plays the energy term, the KL plays the entropy-and-prior term, and we will make that mapping exact in Section 14.6 once the objective is trainable.

Amortization. As it stands, Equation 14.1 still expects a separate q per data point: classical variational inference would now optimize the variational parameters of q for *this* x, then start over for the next one. For a dataset of 10^5 images that is 10^5 optimizations per gradient step, and for a test image it is one more. The fix is to make the variational parameters a *function* of x and learn the function once:

$$q_\phi(z\mid x) \;=\; \mathcal{N}(z;\; \mu_\phi(x),\; \mathrm{diag}\,\sigma^2_\phi(x)), \tag{14.2}$$

amortized Gaussian encoder

where $\mu_\phi(x)$ and $\sigma_\phi(x)$ are the outputs of one neural network with weights ϕ. This is *amortized inference*: the cost of inference is paid once, during training, and thereafter every data point — including data never seen — gets its approximate posterior in a single forward pass. The trade-off is explicit. A single network cannot match the optimal per-datapoint q on every x, so the amortized ELBO sits below what classical VI could reach; that shortfall is the *amortization gap*, distinct from the approximation gap $D_{\mathrm{KL}}(q_\phi\|p_\theta(z\mid x))$ that any restricted family pays. The two gaps stack — the family's first, then the function's — and both reappear in the traps below.

The objective now scales, but it is not yet differentiable where it matters: the reconstruction term is an average over $q_\phi(z \mid x)$, and training the encoder needs ∇_ϕ of that average. That gradient is the subject of the next section.

14.4 The engine, part II: the reparameterization trick

The problem, stated precisely: we need $\nabla_\phi \langle f(z) \rangle_{q_\phi}$, the gradient of an expectation *with respect to the parameters of the distribution being averaged over*. Backprop differentiates deterministic compositions; a sample $z \sim q_\phi$ is not one. There is a classical estimator for exactly this object, and its failure is what makes the trick necessary. Differentiate the integral directly and insert $\nabla_\phi q_\phi = q_\phi \nabla_\phi \log q_\phi$:

$$\nabla_\phi \big\langle f(z) \big\rangle_{q_\phi} \;=\; \int \mathrm{d}z \, f(z) \, \nabla_\phi q_\phi(z \mid x) \;=\; \big\langle f(z) \, \nabla_\phi \log q_\phi(z \mid x) \big\rangle_{q_\phi} .$$

This is the *score-function* (or REINFORCE) estimator: unbiased, universal — it never differentiates f, so f may be a black box — and, for our purposes, uselessly noisy. The reason sits in plain sight: the estimator multiplies the *value* of f by the score, so its fluctuations track the magnitude of f across the whole distribution rather than the local sensitivity of f to ϕ; every irrelevant fluctuation of f enters at full strength. Baselines and control variates tame it somewhat, but the variance grows with the dimension of z (the example below measures the growth), and deep latent models sit exactly where it hurts.

The reparameterization trick replaces value-times-score by an actual derivative. For the Gaussian encoder of Equation 14.2, write the sample as a deterministic map applied to parameter-free noise:

$$\boxed{z \;=\; \mu_\phi(x) + \sigma_\phi(x) \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I),} \tag{14.3}$$

reparameterization trick

which leaves the distribution of z unchanged but relocates the randomness: the expectation is now over a *fixed* distribution, and the parameters sit inside the integrand, where ∇_ϕ can reach them,

$$\big\langle f(z) \big\rangle_{q_\phi} = \big\langle f(\mu_\phi + \sigma_\phi \odot \varepsilon) \big\rangle_{\varepsilon}, \qquad \nabla_\phi \big\langle f(z) \big\rangle_{q_\phi} = \Big\langle \nabla_z f(z) \cdot \nabla_\phi (\mu_\phi + \sigma_\phi \odot \varepsilon) \Big\rangle_{\varepsilon} .$$

The sampler has become a differentiable layer (Figure 14.2): forward, draw ε and compute z; backward, gradients flow through μ_ϕ and σ_ϕ by the chain rule, exactly as through any affine layer with an input. This *pathwise* gradient uses $\nabla_z f$ — the local slope, not the global value — and its variance is correspondingly small; one draw of ε per data point is standard practice and usually enough. The trick does demand something in return: the latent must admit such a deterministic-map-plus-fixed-noise form, which continuous families do and discrete ones do not — for discrete latents one falls back on score-function estimators or the Gumbel-softmax relaxation, which we name and set aside.

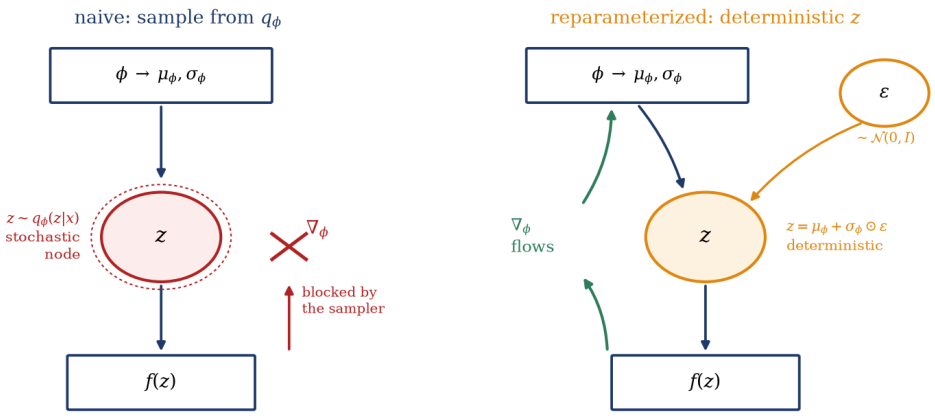

Figure 14.2: The enabling idea as two computation graphs. Left: sampling z directly from $q_\phi(z \mid x)$ makes z a stochastic node, and the backward pass cannot cross it — the gradient in ϕ is blocked at the sampler. Right: the reparameterized graph. The noise $\varepsilon \sim \mathcal{N}(0, I)$ enters as an input, $z = \mu_\phi + \sigma_\phi \odot \varepsilon$ is a deterministic node, and ∇_ϕ flows from $f(z)$ back through μ_ϕ, σ_ϕ (green) by ordinary backpropagation.

The KL term costs nothing. The second term of Equation 14.1 needs no estimator at all: for a Gaussian encoder against the standard-normal prior it is a closed-form function of the encoder outputs. Per latent dimension, with $q = \mathcal{N}(\mu, \sigma^2)$ and $p = \mathcal{N}(0, 1)$,

$$D_{\mathrm{KL}}(q \,\|\, p) = \left\langle \log \frac{q(z)}{p(z)} \right\rangle_q = \left\langle -\frac{(z-\mu)^2}{2\sigma^2} + \frac{z^2}{2} - \log \sigma \right\rangle_q = \frac{1}{2}(\mu^2 + \sigma^2 - 1 - \log \sigma^2),$$

using $\langle (z-\mu)^2 \rangle_q = \sigma^2$ and $\langle z^2 \rangle_q = \mu^2 + \sigma^2$. Summed over dimensions:

$$D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, \mathcal{N}(0, I)) \;=\; \frac{1}{2} \sum_j \left(\mu_j^2 + \sigma_j^2 - 1 - \log \sigma_j^2 \right). \tag{14.4}$$

Gaussian KL, closed form

Assemble the objective. Estimate the reconstruction term by Monte Carlo over ε — a single draw per data point — take the KL in closed form, and average over a minibatch:

$$\mathcal{L}(\theta, \phi) \;=\; \frac{1}{N} \sum_x \Big[\log p_\theta(x \,|\, \mu_\phi(x) + \sigma_\phi(x) \odot \varepsilon) \;-\; D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, p(z)) \Big], \tag{14.5}$$

VAE training objective

maximized over (θ, ϕ) jointly by SGD. The form of the reconstruction term follows from the decoder's likelihood: a Bernoulli decoder (binarized MNIST, the tutorial) makes it a binary cross-entropy, a Gaussian decoder makes it a mean-squared error up to constants — the standard losses of supervised learning, reappearing as likelihood choices. Trained, the network is two machines in one: a *generative model* (draw $z \sim \mathcal{N}(0, I)$, decode) and a *representation learner* (encode x, keep z). Amortization made the objective scale; reparameterization made it differentiable; with both, Week 5's free-energy minimization is a network trained by gradient descent.

Take-home 2

The score-function estimator $\langle f \nabla_\phi \log q_\phi \rangle$ is unbiased but tracks the value of f, and its variance grows with latent dimension. The reparameterization trick $z = \mu_\phi + \sigma_\phi \odot \varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$, moves the randomness onto a fixed distribution so the gradient uses the slope $\nabla_z f$ instead — low variance, one noise draw per data point. With the closed-form Gaussian KL (Equation 14.4), the VAE objective (Equation 14.5) trains by ordinary SGD.

14.5 Example: two estimators of one gradient

The variance claim can be made quantitative on a toy example. Take $f(z) = \|z\|^2$ in d latent dimensions, $q_\phi = \mathcal{N}(\mu, I)$ with $\mu = (1, 0, \dots, 0)$, and estimate one gradient component, $\partial_{\mu_1} \langle f \rangle = 2\mu_1 = 2$ — known exactly, so every estimator can be judged against the truth. Both estimators are unbiased; the question is their spread over single samples. A short calculation with Gaussian moments (worth doing once by hand) gives, per sample,

$$\mathrm{Var}[\hat{g}_{\text{score}}] = (d-1)^2 + 10\,(d-1) + 30, \qquad \mathrm{Var}[\hat{g}_{\text{pathwise}}] = 4\,:$$

the pathwise variance is *independent of dimension* — the estimator $2z_1$ never sees the other $d-1$ coordinates — while the score-function estimator multiplies the full value $\|z\|^2$, fluctuations of every irrelevant dimension included, by the score of z_1, and its variance grows *quadratically* with d. Figure 14.3 samples both. Already at $d = 2$ the score-function estimator is ten times noisier (41 against 4); at $d = 100$ its variance is 1.08×10^4, a factor 2,700 above pathwise, and a latent of realistic size would need thousands of samples to match what reparameterization delivers with one. Figure 14.3 summarizes the case: at realistic latent dimensions, reparameterization reduces the variance by orders of magnitude, making training feasible with a single noise draw per data point.

14.6 What we trained, in Week 5's language

The correspondence with Week 5 is now direct. Fix a data point x and manufacture the energy of Section 9.4, $E(z) = -\log p_\theta(x, z)$ at $T = 1$; then the variational free energy of the encoder is

$$F_{\text{var}}(q_\phi) \;=\; \langle E \rangle_{q_\phi} - S[q_\phi] \;=\; \underbrace{-\,\big\langle \log p_\theta(x \mid z) \big\rangle_{q_\phi}}_{\text{reconstruction} \,=\, -\langle E \rangle \text{ (data term)}} \;+\; \underbrace{D_{\text{KL}}\big(q_\phi(z \mid x) \,\|\, p(z)\big)}_{\text{KL to prior} \,=\, \text{entropy/prior term}} \;=\; -\,\text{ELBO}, \tag{14.6}$$

the VAE loss is the variational free energy

term for term: the reconstruction plays the energy, the KL regularizer bundles the entropy of q_ϕ with the prior, and the training loss $-\mathcal{L}$ of Equation 14.5 *is* F_{var} averaged over the dataset. Training a VAE is variational free-energy minimization with an amortized, neural q — the physicist minimizing F_{var} over a trial family and the engineer running Equation 14.5 through an optimizer are performing the same

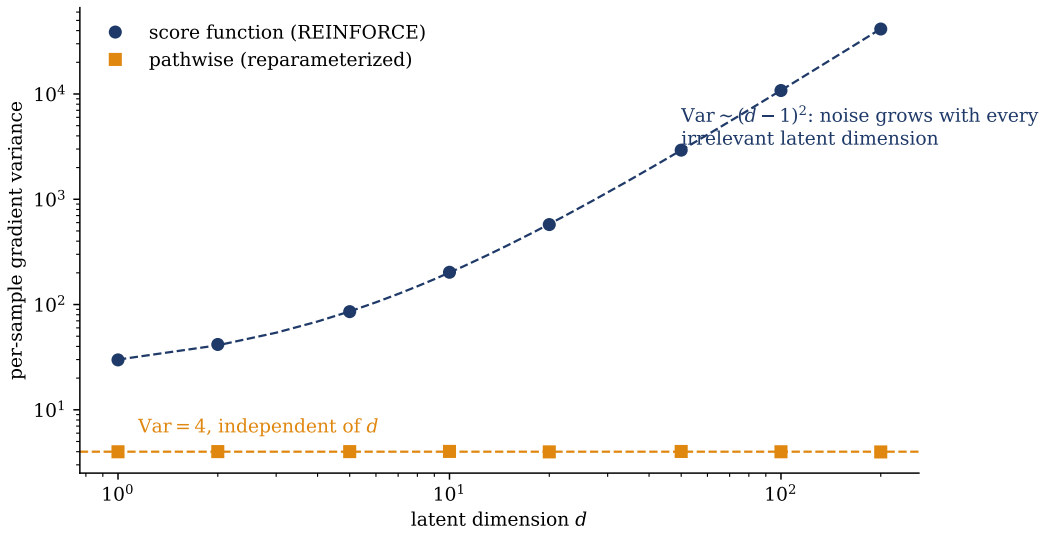

Figure 14.3: One gradient, two estimators: the per-sample variance of $\partial_{\mu_1} \langle \|z\|^2 \rangle_q$ for $q = \mathcal{N}(\mu, I)$, $\mu = (1, 0, \ldots, 0)$, as a function of the latent dimension d (2×10^5 samples per point; dashed lines are the exact Gaussian-moment results). The score-function estimator $\|z\|^2(z_1 - \mu_1)$ (navy) multiplies the full value of f — every irrelevant dimension's fluctuation included — by the score, and its variance grows as $(d-1)^2$: at $d = 100$ it reaches 1.08×10^4. The pathwise (reparameterized) estimator $2z_1$ (orange) uses the slope of f and stays at 4, independent of dimension — a factor 2,700 below score-function at $d = 100$. Both estimators are unbiased; only one is trainable.

optimization on the same functional. The tutorial trains a small VAE on MNIST and tracks reconstruction and KL individually over training.

Posterior collapse. Week 5 flagged that variational inference minimizes the *reverse* KL and warned that its mode-seeking, variance-underestimating character would return (Figure 9.3); here the consequence becomes concrete. Suppose the decoder is expressive enough to model the data with no help from the latent — an autoregressive decoder on images can be. Then the reconstruction term no longer needs z, the KL term is unopposed, and the optimum drives $q_\phi(z \mid x) \to p(z)$ for every x: the KL reaches zero, the latent carries no information, and the "generative model" degenerates into a decoder that ignores its own code (Figure 14.4, left). This is *posterior collapse*, the VAE's characteristic failure, and it is Week 5's zero-forcing made concrete — the reverse KL pays nothing for abandoning latent structure the decoder can fake. The standard fixes reweight the balance: the **-VAE** of Higgins et al. (2017) multiplies the KL term by a coefficient (a KL weight — *not* an inverse temperature, despite this course's reflexes), *KL annealing* ramps that weight from zero during training, and *free bits* exempt a per-dimension quota of KL from the loss. Each adjusts the same balance between reconstruction sharpness and latent structure.

Where the intractability went. Placing the VAE alongside Module 1's models clarifies the comparison. The RBM (Chapter 6) is an *undirected* model: its Z runs over all configurations, the likelihood gradient needs the model average (Equation 5.5), and Week 3 dodged it by sampling a little and accepting CD's bias. The VAE is a *directed* model: the decoder conditional $p_\theta(x \mid z)$ is normalized by construction and the prior is fixed, so there is no Z over x to fight — the hard object is now the *posterior* $p_\theta(z \mid x)$, and the encoder approximates it variationally. The intractability was moved, not removed, and identifying where it sits in each model family is a useful organizing principle: undirected models pay at the partition function, directed models pay at the posterior, and every generative-modeling method in this course is a choice of where to pay.

Trap

Two KLs, one bound, and one trick leave three ways to slip. ① The ELBO's own KL term is $D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))$ — encoder to *prior*, the regularizer inside the objective. The ELBO's *gap* is $D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z \mid x))$ — encoder to *posterior*, the distance to the evidence. Conflating them is the classic error; only the first is computable, and only the second measures approximation quality. ② The ELBO is a lower bound, not the likelihood: a trained VAE sits below its own model's $\log p_\theta(x)$ by the approximation gap *plus* the amortization gap. ③ The reparameterization trick requires a latent that is a differentiable transform of fixed noise — Gaussians qualify, discrete latents do not (Gumbel-softmax or score-function methods take over there).

Take-home 3

The VAE loss *is* F_{var} (Equation 14.6): reconstruction is the data/energy term, the KL to the prior is the entropy/prior term, and training is amortized variational free-energy minimization. Its signature failure, posterior collapse — the KL unopposed, the latent ignored — is Week 5's reverse-KL zero-forcing in practice; -VAE, KL annealing, and free bits reweight the balance. Keep the

two KLs straight: the regularizer points at the prior, the gap points at the posterior.

The VAE is Week 5's variational free energy, minimized by a neural network — reconstruction against a KL regularizer.

14.7 Outlook: the tutorial, and the road to diffusion

Module 2 set out to approximate an intractable distribution and found four ways to do it: the factorized family (Week 5), the corrected expansion (Chapter 11), message passing on the graph (Chapter 12), and the decoupled latent-variable model (this week). The unifying result is that all four optimize the *same functional* — the variational free energy — over different families, and that the last of them is not an approximation scheme at all but a model class, trained by descending that functional with a neural network. The tutorial (16–18, Ph12 106) makes the claim concrete: training a small VAE on binarized MNIST, inspecting reconstructions, prior samples, and a two-dimensional latent space, and decomposing the trained ELBO into its two terms. It is one of the computation-led weeks, so the training runs on the groups' own machines after the predictions are written down.

The connection to the course's title is direct. A VAE has one latent layer; nothing forbids a hierarchy, $x \to z_1 \to \ldots \to z_T$, each layer conditioned on the last. Now make one deliberately strange choice: *fix* the encoder to a chain of Gaussian noising steps $q(z_t \mid z_{t-1})$ — no parameters, no learning, just incremental corruption of the data toward $z_T \approx \mathcal{N}(0, I)$ — and learn only the decoder chain $p_\theta(z_{t-1} \mid z_t)$ that runs the corruption backwards (Figure 14.4, right). That model is a **diffusion model**, and its training loss — the DDPM objective of Week 12 (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015) — is *this lecture's ELBO*, telescoped along the hierarchy into a sum of per-step denoising terms. The free energy that entered the course as a bound in Week 5 and became a training objective today will train the generative models of Module 4 unchanged; from Boltzmann to diffusion is, at the level of the objective, one equation carried forward.

Every method since Week 5 has *approximated or transformed* the distribution; none has *drawn from* it. Module 3 takes the complementary approach and samples: Monte Carlo and the Metropolis rule, annealing on rugged landscapes, and stochastic gradient descent read as a Langevin process — the machinery that turns Week 2's energy landscapes and this module's free energies into working algorithms.

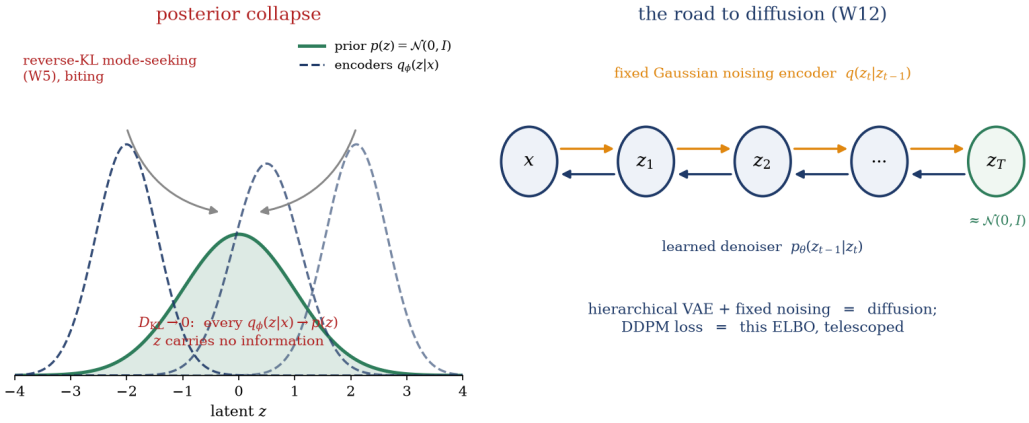

Figure 14.4: Left: posterior collapse. A powerful decoder leaves the KL term unopposed, and every encoder distribution $q_\phi(z \mid x)$ (navy, dashed) is driven onto the prior $p(z)$ (green): the KL reaches zero and the latent carries no information — Week 5's reverse-KL zero-forcing in practice. Right: the road to diffusion. Stack latents $x \to z_1 \to \ldots \to z_T$ with a *fixed* Gaussian noising encoder (orange) and a learned denoising decoder (navy): a hierarchical VAE whose ELBO, telescoped over the steps, is the DDPM training loss of Week 12.

Part III

Module 3 — Sampling and the Fokker–Planck view

15 Markov chain Monte Carlo: sampling what you cannot normalize

Recommended reading

Core (~1 h). Metropolis et al. (1953) — the birth of Markov chain Monte Carlo, on the MANIAC at Los Alamos, and by citation count one of the most consequential algorithms ever published: propose a move, accept with a Boltzmann-ratio probability. Read for the acceptance rule and the Z-free logic; the engine below makes both precise. Alongside it, MacKay (2003), Ch. 29–30 — the inference-side account of Monte Carlo methods, detailed balance, and the mixing problem.
Optional background. Hastings (1970) — the generalization to asymmetric proposals that put the second name in "Metropolis–Hastings"; Geman and Geman (1984) — the Gibbs sampler, and MCMC's arrival in image restoration and Bayesian inference; Mehta et al. (2019), the Monte Carlo sections, for the course's notation.
Prerequisite reminder. The Boltzmann distribution and Z (Week 1); Glauber dynamics and its detailed-balance argument (Section 5.3 — the special case today generalizes); RBM block Gibbs (Section 6.3); rugged landscapes (Weeks 2 and 6). Monte Carlo itself is partly review from the companion *Machine Learning and Physics* course — the first tutorial of this course ran a Metropolis Ising simulation. What is new is the Z-free framing, the general detailed-balance construction and its adjoint-kernel generalization, and the mixing analysis that sets Module 3's central obstacle.

15.1 The other road

Module 2 faced the intractable partition function and *approximated the distribution*: bounded it with a factorized family (Week 5), corrected the bound (Chapter 11), passed messages on the graph (Chapter 12), and finally decoupled it into a trainable latent-variable model (Chapter 14). Every one of those methods lives with Z and manages it — deterministically, with a bias fixed by the chosen family. Module 3 opens the road Week 5's dichotomy left deliberately untaken: **do not approximate p at all. Sample it.**

The fact that makes this possible is the simplest of the course's escapes from Z, and we state it carefully. To *evaluate* $p(x) = e^{-\beta E(x)}/Z$ you need Z, and that is hopeless. But to *sample*, you never need the probability of any single state; you only ever need to *compare two*, and in the ratio

$$\frac{p(x')}{p(x)} = \frac{e^{-\beta E(x')}/Z}{e^{-\beta E(x)}/Z} = e^{-\beta\,\Delta E}, \qquad \Delta E = E(x') - E(x),$$

the partition function cancels. Everything that follows today is the systematic exploitation of that cancellation: *you can draw samples from a distribution you cannot normalize.* The course has now met three distinct ways of defusing Z — Module 1 dodged its parameter-gradient (contrastive divergence), Module 2 bounded its logarithm (the variational free energy), and Module 3 makes it cancel — with the fourth and final way, removal ($\nabla_x \log Z = 0$), still two weeks out.

The history closes a loop the course opened on day one. In 1953, at Los Alamos, Nicholas Metropolis, Arianna and Marshall Rosenbluth, and Augusta and Edward Teller published *Equation of State Calculations by Fast Computing Machines* (Metropolis et al. 1953) — the "fast computing machine" being the MANIAC, one of the first electronic computers, and the calculation being the equilibrium properties of a fluid of hard disks that no closed form could touch. Their idea — walk through configuration space, accepting each proposed move with a probability built from the Boltzmann *ratio* — is Markov chain Monte Carlo, and it now runs everywhere from lattice QCD to Bayesian statistics to the negative phase of Week 3's Boltzmann machines. Hastings (1970) generalized it; the modern world computes with it.

Module 3 is defined as much by its cost as by its trick. Sampling is *asymptotically exact* — run the chain forever and the answers converge to the truth, no family bias, no bound looseness. What it pays instead is **time**: the chain must *mix* — forget its initialization and visit configuration space with the right frequencies — and mixing is slow precisely where this course's landscapes are interesting: near phase transitions, and in the rugged terrain of Weeks 2 and 6. Module 3 swaps the normalization obstruction for the sampling one; today makes both the trick and the price precise.

> Take-home 1
>
> To sample $p \propto e^{-\beta E}$ you never need Z — only energy differences, because Z cancels in every ratio $p(x')/p(x) = e^{-\beta \Delta E}$. Sampling is Module 3's road out of the normalization obstruction Module 2 bounded; its price is mixing.

15.2 Approximate versus sample

The module map frames the fork (Figure 15.1). Road one — Module 2 — replaces p by the best member of a tractable family: deterministic, fast, sometimes armed with a bound, and *biased forever* by whatever the family cannot express (the factorized ansatz never contained a correlation, no matter how long you optimized). Road two — this module — builds a stochastic process whose visits *are* draws from p: no family, no bias in the limit, and instead a stochastic error that shrinks with simulation time — *if* the process equilibrates. The choice is deterministic-and-biased against stochastic-and-exact-eventually; the course will need both, and Module 4 will need them married.

Figure 15.2 draws the method before the mathematics. Left: a walker on an energy landscape takes proposed steps — downhill steps always accepted, uphill steps accepted with probability $e^{-\beta \Delta E}$ — and the histogram of its visits piles up, over time, into exactly $e^{-\beta E}$: the Boltzmann distribution, assembled from nothing but local energy differences. Right: the same walker at low temperature, trapped in one

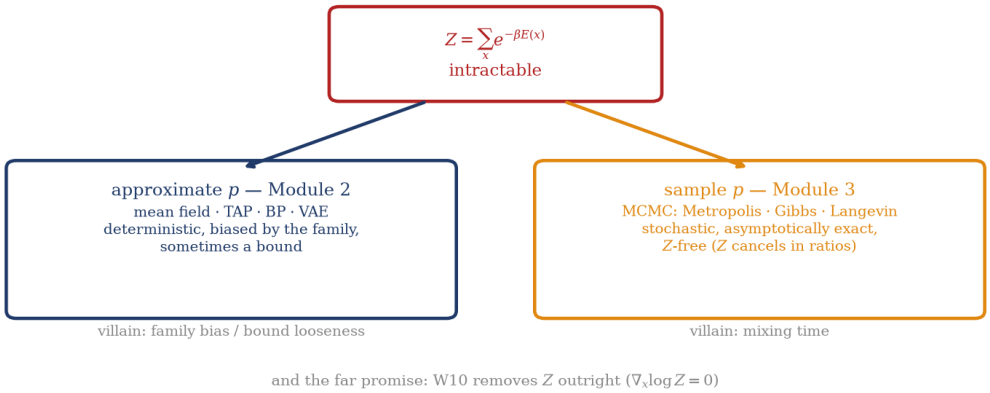

Figure 15.1: The two roads out of an intractable Z. Module 2 approximated the distribution — mean field, TAP, message passing, the VAE — deterministic and biased by the chosen family, sometimes with a bound. Module 3 samples it: Markov chain Monte Carlo is stochastic, asymptotically exact, and Z-free, because the normalizer cancels in every ratio the algorithm ever uses. Each road has its own obstruction: family bias on one, mixing time on the other — and Week 10's score matching still waits at the bottom, where Z is neither bounded nor canceled but removed.

basin behind a barrier it crosses only with exponentially small probability — sampling that basin faithfully, having never seen the rest of configuration space. The left panel is the theorem we now prove; the right is the price we then pay.

15.3 The engine: detailed balance and the Metropolis rule

Setup. We want a stochastic process on configuration space — a *Markov chain*, specified by transition probabilities $W(x \to x')$ — whose long-run visit frequencies are $p(x) \propto e^{-\beta E(x)}$, built using only the ratio $e^{-\beta \Delta E}$. The design condition is *stationarity*: if the chain's states are distributed as p at one step, they must remain so at the next,

$$\sum_x p(x)\, W(x \to x') \;=\; p(x') \quad \text{for all } x'.$$

(That a well-behaved chain then *converges* to this p from any start — uniqueness of the stationary distribution under irreducibility and aperiodicity, Perron–Frobenius — is the same bookkeeping we took as read for Glauber in Section 5.3, and the script keeps it in the margin. Design is today's business; convergence *rate* is the second half of the lecture.)

Detailed balance. Stationarity is a global condition — a sum over all states, awkward to engineer directly. The workhorse is a *local, pairwise* condition that implies it:

$$\boxed{p(x)\, W(x \to x') \;=\; p(x')\, W(x' \to x) \quad \text{for all pairs } x, x'} \tag{15.1}$$

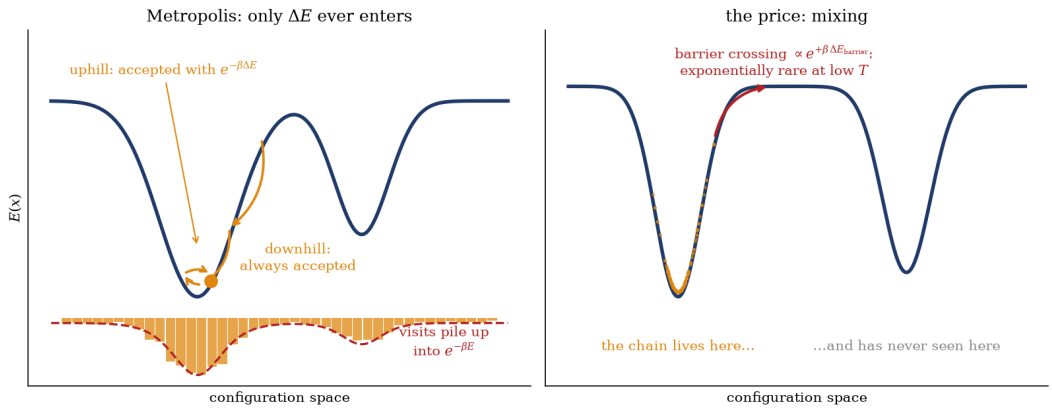

Figure 15.2: The central picture of Module 3's opening. Left: the Metropolis walker — downhill proposals always accepted, uphill ones accepted with probability $e^{-\beta\Delta E}$ — and the histogram of its visits (orange) converging onto $e^{-\beta E}$ (red dashed): the Boltzmann distribution sampled without Z ever being computed. Right: the price. At low temperature a barrier is crossed only with probability $\sim e^{-\beta\,\Delta E_{\text{barrier}}}$ per attempt; the chain samples its home basin perfectly and has never seen the other one. Equilibration within a basin is not equilibration.

detailed balance ⇒ p stationary

— *detailed balance*: at equilibrium, probability flows equally in both directions across every pair of states, not just in aggregate. The implication is one line — sum both sides over x, and $\sum_x W(x' \to x) = 1$ collapses the right side:

$$\sum_x p(x)\, W(x \to x') \;=\; p(x') \sum_x W(x' \to x) \;=\; p(x').$$

You have seen this condition before, doing exactly this job: it is what gave Glauber dynamics its Boltzmann stationary law in Section 5.3, there checked for one specific flip rule. Today it is promoted from a property we verified to a *design principle* we build from. (One margin note, made precise at the end of this section: detailed balance is *sufficient*, not necessary — valid and sometimes faster chains exist that break it. It is the standard tool because it is the checkable one.)

The Metropolis rule. Split each transition into a *proposal* and an *acceptance*: $W(x \to x') = g(x \to x')\, A(x \to x')$, with g symmetric ($g(x \to x') = g(x' \to x)$ — flip a random spin, jiggle a coordinate) and the acceptance chosen as

$$\boxed{A(x \to x') \;=\; \min(1,\; e^{-\beta\,\Delta E})} \tag{15.2}$$

Metropolis acceptance

— accept every downhill move; accept an uphill move with probability $e^{-\beta\Delta E}$ (Figure 15.3). The detailed-balance check, which completes the derivation, takes two lines. With symmetric g,

$$p(x)\, g\, A(x \to x') \;=\; g\, \min(p(x),\, p(x')) \;=\; p(x')\, g\, A(x' \to x),$$

because $p \min(1, p'/p) = \min(p, p')$ — an expression *manifestly symmetric* under $x \leftrightarrow x'$, so Equation 15.1 holds by inspection. And now look at what the algorithm actually consumed: p entered only through the ratio $p(x')/p(x) = e^{-\beta\Delta E}$. Z

never appears. The chain's visit frequencies converge to the full, correctly normalized Boltzmann distribution — normalizer included — and the normalizer was never computed, only canceled. One local rule, using one energy difference per step, samples a globally intractable distribution exactly.

Two standard extensions complete the design. For *asymmetric* proposals, Hastings (1970) rebalances the acceptance, $A = \min\left(1, \frac{p(x')\,g(x'\to x)}{p(x)\,g(x\to x')}\right)$ — still a ratio, Z still gone. And proposing from an *exact conditional* distribution (resample one variable given all others) makes the Hastings ratio identically one — accept always: the **Gibbs sampler** (Geman and Geman 1984), which you have been running since Week 3, because the RBM's block Gibbs (Section 6.3) is precisely this with whole layers as the resampled blocks. Glauber's flip rule, likewise, is just a different valid acceptance function $(1/(1+e^{\beta\Delta E})$ — the "heat bath" choice, gentler than Metropolis's min, satisfying the same detailed balance). One principle generates a family of algorithms.

Sufficient, not necessary: the adjoint kernel. The precise statement identifies what stationarity actually requires — and introduces one of the module's central objects (Welling, Lu, and Holdijk 2026, ch. 11 on Markov chain Monte Carlo). We claim that p is stationary under W *if and only if* there exists a second transition probability $W^\dagger$ — the *adjoint* kernel, nonnegative and normalized — satisfying the *generalized balance* relation

$$p(x)\,W(x\to x') \;=\; p(x')\,W^\dagger(x'\to x) \quad \text{for all pairs } x, x'. \tag{15.3}$$

generalized balance: stationarity's exact requirement

One direction is the same one-line sum as before: if such a $W^\dagger$ exists, summing Equation 15.3 over x gives $\sum_x p(x)\,W(x\to x') = p(x')\sum_x W^\dagger(x'\to x) = p(x')$, which is stationarity. For the converse there is only one candidate — solve Equation 15.3 for

$$W^\dagger(x'\to x) \;=\; \frac{p(x)\,W(x\to x')}{p(x')},$$

manifestly nonnegative, and check that it is normalized:

$$\sum_x W^\dagger(x'\to x) \;=\; \frac{1}{p(x')}\sum_x p(x)\,W(x\to x'),$$

which equals one for every x' exactly when p is stationary. The constructed $W^\dagger$ is a genuine transition kernel precisely when the chain has p as its stationary law, and the equivalence is proved. Detailed balance is the special case $W^\dagger = W$: the chain is its own adjoint.

The adjoint has a physical reading that the course will lean on. By construction, $W^\dagger(x'\to x)$ is the probability that a stationary chain sitting at x' *arrived from* x — Bayes' rule applied to one step — so $W^\dagger$ is the chain watched in reverse. Detailed balance then says the film run backwards is statistically indistinguishable from the film run forwards: no pair of states carries a net probability current, which is the defining property of *equilibrium*. Generalized balance permits more: $W^\dagger \neq W$, probability circulating around closed loops, every state's total inflow and outflow balancing while individual pairs carry steady current. Such a chain is stationary but *irreversible* — a nonequilibrium steady state, the same object whose circulating

currents reappear in Week 9's anisotropic SGD — and irreversibility is not a defect: the persistent current stirs the state space rather than diffusing through it, which is why deliberately non-reversible chains can mix faster than any detailed-balance competitor built from the same moves. You have, in fact, been running one since Week 3: fixed-scan Gibbs — the RBM's $v \to h \to v$ ladder — satisfies detailed balance in each half-step but not in their composition (its adjoint is the scan run in the opposite order), and it leaves p invariant all the same. Why, then, does detailed balance remain the standard tool? Because normalizing $W^\dagger$ is exactly as global a check as stationarity itself — a sum over all states — while detailed balance is verified pair by pair, from energy differences alone. Design locally with Equation 15.1, knowing the license Equation 15.3 grants is wider; Week 10 will use that license out of equilibrium, where the marginals themselves change in time and the reversed kernel becomes the generative object.

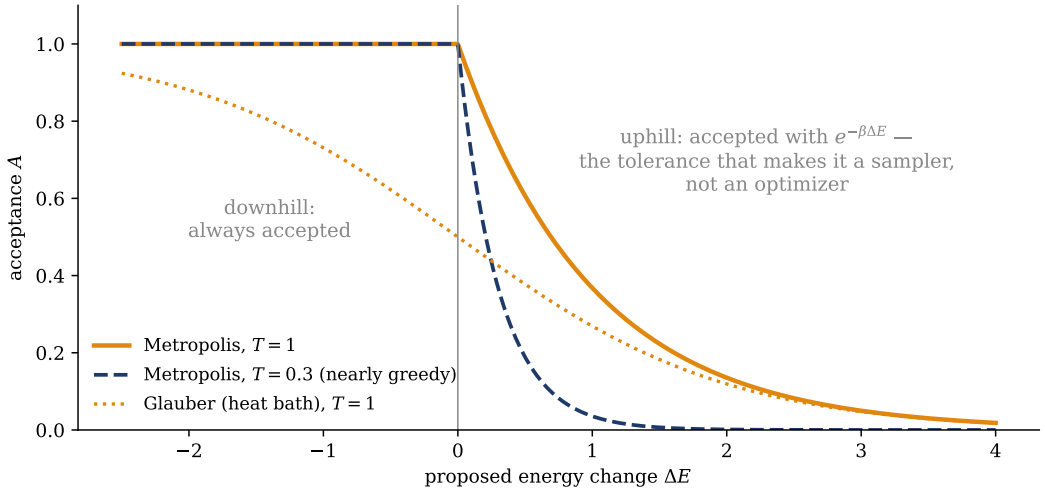

Figure 15.3: The Metropolis acceptance Equation 15.2 against the proposed energy change, at two temperatures, with Glauber's heat-bath rule (Section 5.3) alongside for the warm case. Downhill moves ($\Delta E < 0$) are always accepted; uphill moves survive with probability $e^{-\beta\Delta E}$ — the uphill tolerance that distinguishes a *sampler* from Week 2's greedy descent, and the only place temperature enters the algorithm. Cool the chain and the tolerance narrows toward pure descent (navy dashed); both acceptance functions satisfy detailed balance with respect to the same Boltzmann distribution, Metropolis simply accepting more.

15.4 We have been sampling since Week 3 — and the price is mixing

The unification, first. The method is not new to this course, but it now gets its name. Glauber dynamics — Week 3's stochastic flip rule, from which the Boltzmann machine was built — is single-spin MCMC with the heat-bath acceptance. The RBM's block Gibbs ladder is the Gibbs sampler, layer by layer. And now the result Week 3 could only promise: **contrastive divergence is a truncated Markov chain** — a sampler deliberately run k steps from the data instead of to equilibrium — and *that is the entire content of its bias.* In Section 6.4 we quantified the CD-k gap and watched it decay with k; today's language names the mechanism:

an unequilibrated chain samples the wrong distribution, so the negative phase it estimates is systematically off, by exactly the amount the chain has failed to mix — a corollary of today's framework.

The price: mixing. Everything above says the chain converges; none of it says *when*. Three costs follow, in ascending order of severity. First, MCMC samples are **correlated**: consecutive states differ by one accepted move, so a run of N steps carries not N independent draws but roughly $N/(2\tau)$, where τ is the *autocorrelation time* — the number of steps the chain needs to forget itself. Second, the chain needs **burn-in**: samples taken before stationarity is reached are draws from the wrong distribution and must be discarded (CD's bias again, this time as a lab habit). Third — the main obstruction — τ itself can blow up. It does so in exactly the two regimes this course keeps meeting. **Near a critical point**, fluctuations become system-spanning while the dynamics stays local, and τ grows as a power of the correlation length (*critical slowing down* — the worked example below measures it); physics has cures here (cluster algorithms that flip correlated islands wholesale — named for the script, not developed). **In rugged, glassy landscapes** — Week 2's overloaded memory, Week 6's spin glass — barriers are extensive, crossing times are exponential, and below the glass transition ergodicity *breaks*: the chain is confined to one valley of many, and no polynomial patience rescues it. For the models this course cares about, Figure 15.2's right panel is the typical terrain rather than a pathology.

A question from Week 3 is now settled: normalization and sampling are **two distinct obstructions**. Module 2 fought the first — Z, the normalizer — by bounding and approximating. Module 3 makes the first *cancel* outright, and inherits the second: **mixing**. Neither implies the other (a perfectly known Z would not flatten a single barrier), and the map of the field keeps them separate. MCMC yields *samples and ratios*, never Z itself — if you genuinely need the normalizer (to compare models, to report a likelihood), sampling alone is not enough, and the machinery that extracts free energies from chains (thermodynamic integration, annealed importance sampling) is Week 11's, where it arrives in the language of nonequilibrium thermodynamics.

Trap

A chain that has not mixed samples the wrong distribution — and it will not warn you. MCMC's samples are correlated (effective sample size $N/(2\tau)$, not N), require burn-in, and are only asymptotically exact *given* equilibration. An unmixed chain produces smooth, convergent-looking, reproducibly *wrong* averages — it is why contrastive divergence is biased (Section 6.4), and why every MCMC result owes its reader an autocorrelation time. And do not over-credit the trick: Z cancels in ratios, but MCMC never *gives* you Z — samples are not normalizers (that debt is settled in Week 11).

Take-home 2

Glauber dynamics, block Gibbs, and contrastive divergence are all MCMC — and CD is biased precisely because its chain is deliberately not equilibrated. The price of Z-free sampling is mixing: correlated samples ($N_{\text{eff}} \approx N/(2\tau)$), mandatory burn-in, autocorrelation times that diverge at critical points and grow exponentially in glassy landscapes. Normalization and mixing are distinct

obstructions; Module 3's subject is the second.

15.5 Example: Metropolis on the two-dimensional Ising model

One chain makes both faces of the lecture visible — the method's power and its cost. Run checkerboard Metropolis on a 24×24 Ising lattice (the two sublattices updated in alternating half-sweeps, a vectorized equivalent of single-site updates) across the temperature range straddling Onsager's exact critical point $T_c = 2/\log(1+\sqrt{2}) \approx 2.269$ (Onsager 1944), and measure two things: the magnetization, and the autocorrelation time of its time series (Figure 15.4).

The magnetization curve shows what sampling achieves. The full phase transition — the ordered branch, the collapse at T_c, the disordered tail — emerges from a walker that only ever computed *local energy differences*: no Z, no free energy, no 2^{576}-term sum, just $\min(1, e^{-\beta\Delta E})$ applied a few million times. Where Week 5 obtained this curve from a variational bound (with a manufactured transition in the wrong place) and the exact treatments of Weeks 1–2 needed enumerability, the sampler simply *visits* the truth. And it is genuinely the truth: on a 4×4 lattice, where all 65,536 states can be enumerated, the same chain reproduces the exact $\langle|m|\rangle$ to a few parts in a thousand:

T	exact $\langle\|m\|\rangle$ (4×4, enumerated)	MCMC estimate
2.0	0.9189	0.9193
2.5	0.7647	0.7606
3.0	0.6013	0.5977

The autocorrelation panel shows the cost. Away from the transition, $\tau \approx 1$–3 sweeps — consecutive configurations decorrelate almost immediately, and every sweep is nearly a fresh sample. Approaching T_c, the time series becomes sticky — long correlated excursions as system-spanning fluctuation patterns rearrange through single spin flips — and τ spikes to ≈ 29 sweeps, a twenty-fold collapse in effective sample size, *at fixed computational cost*. Critical slowing down is visible directly, and it scales: on larger lattices the peak grows as a power of the system size, and in the glassy landscapes of Weeks 2 and 6 the analogous times are exponential. The sampler is exact when it mixes; the plot shows where mixing fails — where the physics is most interesting.

> *You can sample a distribution you cannot normalize — Z cancels in every ratio; what you cannot escape is mixing.*

The week continues from here. The warm-up card, due the evening before the next lecture, covers Kirkpatrick, Gelatt, and Vecchi (1983) — ten pages of *Science*, and a landmark idea-transplant between fields; read it with today's Metropolis rule in hand, and notice what the authors do to the temperature. In the next lecture we do it deliberately: temperature stops being a fixed parameter and becomes a *schedule* — run the chain hot to cross barriers, cool it to distill low-energy states — and

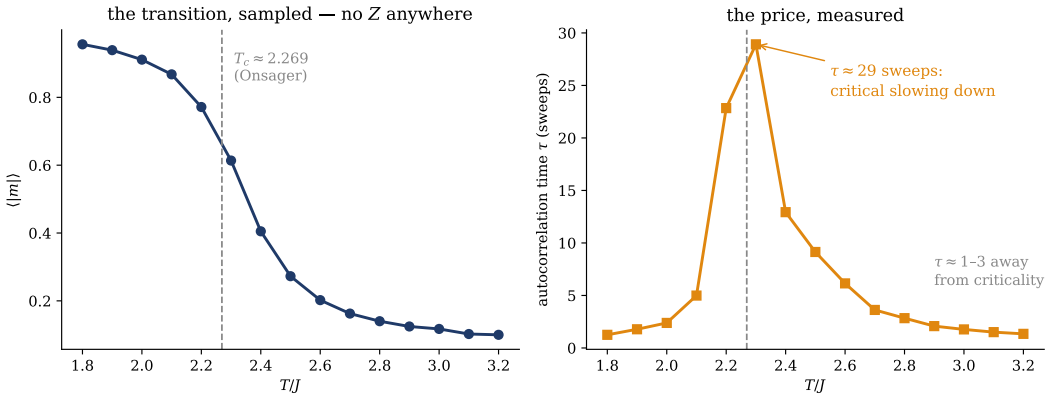

Figure 15.4: The miracle and the bill, on one chain: checkerboard Metropolis on a 24×24 Ising lattice (10,000 measurement sweeps per temperature after burn-in). Left: the sampled magnetization $\langle|m|\rangle(T)$ traces the full phase transition across Onsager's $T_c \approx 2.269$ (dashed) — obtained from local energy differences alone, with Z never computed (and verified against exact enumeration on a 4×4 lattice to a few parts per thousand). Right: the price — the autocorrelation time of the magnetization series spikes from ~ 1 sweep far from criticality to ≈ 29 sweeps near T_c: critical slowing down. The effective sample size collapses exactly where the physics is most interesting; in glassy landscapes the same cost becomes exponential.

the derivation-led tutorial takes that machinery to a famously hard optimization problem, then sets it beside a stochastic-gradient sampler of a Bayesian posterior. What those two have to do with each other is for the tutorial to establish. The tutorial requires no devices.

> Take-home 3
>
> Metropolis samples the 2-D Ising phase transition without ever computing Z — and matches exact enumeration where enumeration is possible — but its autocorrelation time spikes twenty-fold at T_c: critical slowing down, the mixing cost made visible. The method is exact when it mixes and silently unreliable when it does not; whether a chain can be trusted is entirely a matter of equilibration.

15.6 Outlook: temperature becomes a tool

Module 3 is open. A fifty-line algorithm samples the distribution whose normalizer was intractable throughout Weeks 3–7, by canceling it in every ratio. The cost is equilibration, purchased in time, at prices that diverge where landscapes become interesting. The next lecture takes the algorithm's parameter — temperature — and turns it from a parameter into an *instrument*: at high T the walker ignores barriers and roams globally; at low T it commits and refines; and a *cooling schedule* harvests both, using the Boltzmann distribution's own concentration onto low-energy states as an optimization principle. That is **simulated annealing** (Kirkpatrick, Gelatt, and Vecchi 1983) — metallurgy imported into computation — and with it Module 3's second theme: the dynamics of a sampler as a subject in itself.

Week 9 then delivers the module's main consequence for machine learning. Stochastic gradient descent — the algorithm training every model in this course's second half — takes noisy downhill steps on a loss landscape: a proposal (the gradient direction, jittered by the mini-batch) and no explicit acceptance, yet its long-run behavior is that of a Langevin sampler at a temperature set by the learning rate and batch size. Training *is* sampling; the optimizer *is* a Metropolis walker with the thermostat hidden in the hyperparameters; and the stationary distribution it samples — which minima it likes, and why flat ones — is a Boltzmann distribution over the loss. The Fokker–Planck machinery for making all of that precise is Week 9's engine, and it is also — one module later — the mathematical substrate of diffusion models. The spine reads once more, now with three of its four vertebrae in place: Module 1 *dodged* $\nabla_\theta \log Z$, Module 2 *bounded* $\log Z$, Module 3 makes Z *cancel* — and Week 10 removes it.

16 Simulated annealing: optimization by cooling

Recommended reading

Core (~1 h). Kirkpatrick, Gelatt, and Vecchi (1983) — ten pages of *Science*, and the warm-up reading: metallurgical annealing imported into combinatorial optimization, with chip placement and the travelling salesman as the demonstrations. Read for the physical analogy and the role of the schedule; today's §2–§3 make both precise. Alongside it, MacKay (2003), the simulated-annealing passages of the Monte Carlo chapters — the inference-side account, on the footing the previous lecture built.
Optional background. Geman and Geman (1984) — the same paper that gave the previous lecture the Gibbs sampler proves the logarithmic-cooling convergence theorem this lecture states; Černý (1985) — the independent, simultaneous invention of simulated annealing, from Bratislava rather than IBM; Welling and Teh (2011) — stochastic gradient Langevin dynamics, the bridge this lecture plants and the tutorial cracks; Mehta et al. (2019), the optimization sections, for the ML framing.
Prerequisite reminder. All of Chapter 15 — the Z-free ratio, detailed balance, the Metropolis rule Equation 15.2, and the mixing time τ with its divergence at criticality (Section 15.5) — plus the temperature dependence of the Boltzmann distribution Equation 1.2 (Week 1), the phase transition of Section 10.2, and the rugged landscapes of Weeks 2 and 6 (Chapter 4, Section 11.4). New content: the $T \to 0$ concentration on global minima, cooling schedules and their convergence, the quench and its price in excess free energy, and the temperature–learning-rate correspondence.

16.1 Temperature becomes an instrument

The previous lecture's Metropolis chain has one parameter. The Metropolis chain samples $p \propto e^{-E/T}$ at whatever temperature we choose, using only energy differences, and its cost is mixing: the autocorrelation time τ that diverges at critical points and grows exponentially in glassy landscapes. Throughout that lecture the knob stayed fixed — temperature was part of the problem statement, the T at which we wanted equilibrium averages. Today we turn it while the chain runs. Hot, the walker of Figure 15.2 ignores barriers and roams the whole landscape; cold, it commits to a basin and refines downhill. Start hot and cool slowly, and the chain rides the Boltzmann distribution's own concentration onto low-energy states all the way down to the ground state. A sampler, cooled, becomes an optimizer — that is **simulated annealing**, and it is Module 3's second theme: the dynamics of the sampler as a subject in itself.

The name is not a metaphor; it is a lab procedure. A metallurgist who wants a defect-free crystal — the global minimum of the lattice energy — heats the metal and cools it *slowly*, so that at every stage the atoms stay near thermal equilibrium and settle collectively into ever-lower-energy arrangements. Cool the same melt *fast* — a *quench* — and it freezes wherever it happens to be: a glass, disordered, trapped in a local minimum of the energy, held there by barriers it no longer has the thermal energy to cross. In 1983 Kirkpatrick, Gelatt, and Vecchi took this procedure literally (Kirkpatrick, Gelatt, and Vecchi 1983): replace the metal's energy by any cost function, the atoms' thermal motion by a Metropolis chain, and the furnace schedule by a programmed temperature $T(t)$, and the blacksmith's craft becomes a general-purpose optimizer. (The course has met the first author before: this is the same Scott Kirkpatrick who co-wrote *Solvable Model of a Spin-Glass* (Sherrington and Kirkpatrick 1975) — a spin-glass physicist recognizing that the frustrated, rugged landscapes of Section 11.4 and the cost surfaces of hard optimization problems are the same terrain, and that physics already owned a method for navigating them. Independently and almost simultaneously, Černý (1985) built the same algorithm around the travelling salesman.) A year later, Geman and Geman (1984) supplied the theorem: cooled slowly enough, the chain reaches the global minimum with probability one — with a precise meaning of "slowly enough" whose fine print this lecture states in full.

The question, then: *how do you find the global minimum of a rugged landscape?* Gradient descent traps in the first local minimum downhill of its start. Sampling at a fixed low temperature concentrates on the right answer in principle but cannot get there — the previous lecture's mixing failure at its worst. The resolution is the schedule, and the lecture is organized around one trade: cool too slowly and the answer costs forever; cool too fast and you quench into a glass. The bridge to the tutorial and all of Week 9 is already visible: in machine learning the temperature being scheduled has another name — the *learning rate* — and every practitioner who has ever decayed a learning rate has been annealing without knowing it.

Take-home 1

Cooling a Metropolis chain turns a sampler into an optimizer: as $T \to 0$ the Boltzmann distribution concentrates on the global minima, and a chain that tracks equilibrium while cooling is carried onto the ground state. Cool slowly (anneal) and you arrive; cool too fast (quench) and you freeze into a local minimum — a glass. The schedule is the algorithm.

The central picture summarizes the argument (Figure 16.1): one rugged landscape, one family of Boltzmann distributions above it, and two possible fates below.

16.2 The engine, part I: cooling concentrates

One landscape, a family of distributions. Fix the energy $E(x)$ and let the temperature vary: Equation 1.2 becomes a one-parameter family

$$p_T(x) = \frac{e^{-E(x)/T}}{Z_T}, \qquad Z_T = \sum_x e^{-E(x)/T},$$

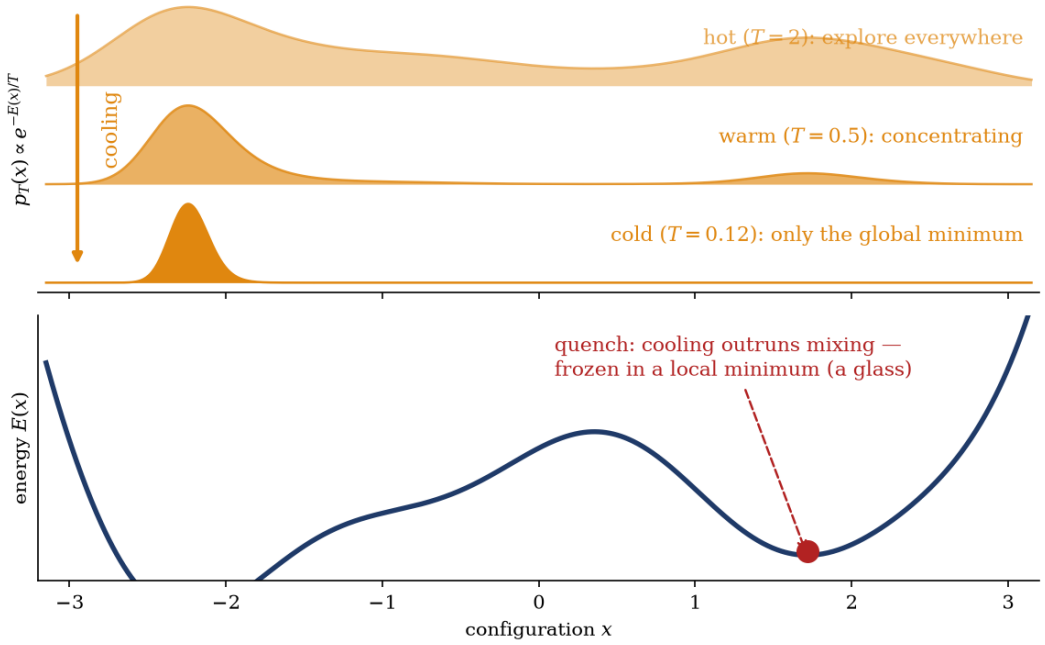

Figure 16.1: The lecture in one picture. Top: the Boltzmann distribution $p_T \propto e^{-E(x)/T}$ over one rugged landscape at three temperatures — hot, the mass spreads over every basin (exploration); warm, it drains toward the deep wells; cold, only the global minimum survives. Bottom: the two fates of a cooled chain. Annealing — cooling slowly enough to stay near equilibrium at every temperature — tracks the concentrating distribution into the global minimum (the crystal); a quench — cooling faster than the chain can mix — strands it in a local minimum (a glass), exactly as in metallurgy.

interpolating from the uniform distribution ($T \to \infty$, where the exponent flattens completely) toward something increasingly selective as T falls. How selective, exactly? The question answers itself if we compare any state to the best one — and the comparison is the previous lecture's central trick, because in a *ratio* the intractable Z_T cancels.

The concentration ratio. Let x^* be a global minimum of the energy, $E^* = \min_x E(x)$, and let x be any state with $E(x) > E^*$. Then

$$\boxed{\frac{p_T(x)}{p_T(x^*)} = e^{-(E(x)-E^*)/T} \xrightarrow{T\to 0} 0} \tag{16.1}$$

the $T \to 0$ concentration

for *every* non-optimal state, however small its excess energy: any fixed gap $E(x) - E^* > 0$ is eventually enormous compared to T. All probability drains onto the global minima, and in the limit

$$p_T \xrightarrow{T\to 0} \text{uniform on } \arg\min_x E(x)$$

— a single point mass if the minimum is unique, and the uniform distribution over all g ground states if it is g-fold degenerate (our spin-glass example below has $g = 2$, the two global spin flips of one configuration). Temperature sets how sharply the Boltzmann distribution prefers low energy, and at $T \to 0$ it prefers *only the lowest* — so a single sample from the zero-temperature limit is a certificate of global optimality. Optimization has become a sampling problem.

The free-energy reading. Week 1's decomposition $F = \langle E \rangle - TS$ says the same thing thermodynamically. At high temperature the entropy term dominates: the free energy is minimized by *spreading* — many states, all basins, exploration favored by the $-TS$ term. As T falls, the entropic reward shrinks and the energy term takes over: the mass must *commit* to the lowest states, however few. Annealing is the walk this balance takes as T moves from one regime to the other, and the concentration Equation 16.1 is its destination. Figure 16.2 makes the statement quantitative on the very spin glass the worked example will anneal: sixteen spins, $2^{16} = 65{,}536$ states enumerated exactly, and the total Boltzmann weight of the two ground states rising from 0.17% at $T = 2$ (already a 55-fold enhancement over the uniform $2/65{,}536 \approx 0.003\%$) through one half at $T \approx 0.34$ to 99% by $T \approx 0.12$.

16.3 The engine, part II: the schedule and the quench

Why you cannot simply set $T = 0$. Put $T = 0$ into the previous lecture's acceptance rule Equation 15.2: every uphill acceptance $e^{-\Delta E/T}$ vanishes, and the chain accepts only moves that lower the energy — *greedy descent*. Such a chain stops, permanently, at the first configuration from which every move goes uphill: a local minimum, and almost never the global one. Our sixteen-spin glass has 28 of these single-flip-stable states, and greedy descent from a random start reaches the true ground state in only 29% of runs — the other 71% end in one of the 26 impostors. Sampling *at* $T = 0$ is precisely the trapping we set out to avoid; the zero-temperature limit is a destination to approach, never an operating point. So the chain must start hot, where barriers are crossable, and *cool*: run Metropolis with

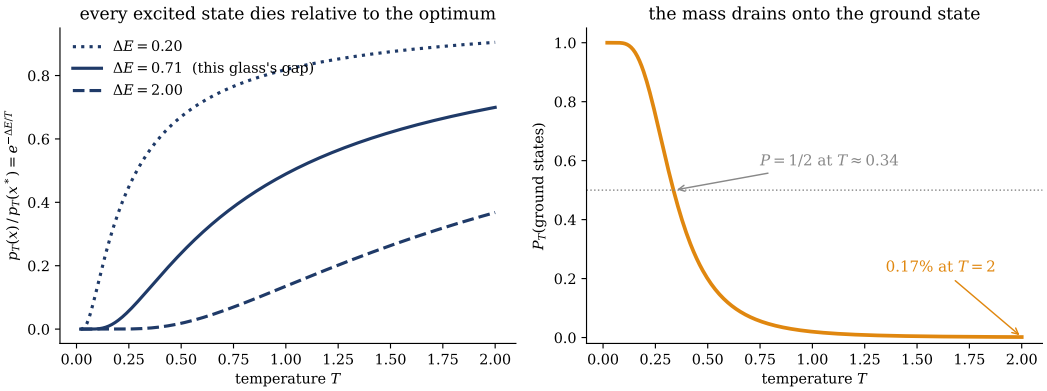

Figure 16.2: Concentration, exactly, on the spin glass of the worked example ($N = 16$ spins, couplings $J_{ij} \sim \mathcal{N}(0, 1/N)$, all 65,536 states enumerated). Left: the boxed ratio Equation 16.1 for three energy gaps — the middle curve uses this instance's actual first excitation gap $\Delta E = 0.71$. Every non-optimal state's relative weight dies as $T \to 0$; bigger gaps die sooner. Right: the total Boltzmann probability of the two degenerate ground states against temperature. At $T = 2$ they carry 0.17% of the mass; the curve crosses $1/2$ at $T \approx 0.34$ and reaches 99% by $T \approx 0.12$ — a sample from the cold distribution is, with overwhelming probability, the exact optimum. The entire remaining question is whether a chain can *follow* this curve down.

a time-dependent temperature $T(t)$, decreasing from exploration to commitment. Everything now hangs on how fast.

The constraint is the previous lecture's mixing time. The concentration argument of Section 16.2 is a statement about *equilibrium* at each temperature; the chain inherits its benefits only if it actually equilibrates as it cools. Equilibration at temperature T costs the mixing time $\tau(T)$ — and $\tau(T)$ *grows* as T falls, because barrier crossings are exponentially suppressed: a barrier of height ΔB takes of order $e^{\Delta B/T}$ attempts (the Arrhenius law W9 will derive from Langevin dynamics; today it is the previous lecture's uphill-acceptance rule read as a waiting time). Cooling therefore races against a clock that runs slower the colder it gets, and the race has exactly two outcomes. Cool within the clock, and the chain stays near $p_{T(t)}$ at every moment and is carried onto the ground state. Outrun the clock, and the chain falls out of equilibrium and freezes wherever it stands — the **quench**, and the frozen result is a *glass* in exactly the metallurgist's sense: a configuration held in place not because it is good but because the moves that would improve it have become too expensive to afford.

The quench, priced. "Frozen wherever it stands" can be made quantitative with Week 1's free-energy decomposition and the previous lecture's adjoint kernel (Welling, Lu, and Holdijk 2026, ch. 10 on stochastic thermodynamics). Define the *nonequilibrium free energy* of whatever distribution ρ the chain currently carries, $F[\rho] = \langle E \rangle_\rho - T\,S[\rho]$ — Week 1's decomposition, now evaluated off equilibrium. Two lines relate it to the equilibrium value $F_T = -T \log Z_T$: since $-\log p_T(x) = E(x)/T + \log Z_T$,

$$F[\rho] - F_T \;=\; \langle E\rangle_\rho + T\langle \log\rho\rangle_\rho + T\log Z_T \;=\; T\,\big\langle \log\rho - \log p_T\big\rangle_\rho \;=\; T\,D_{\mathrm{KL}}(\rho\,\|\,p_T) \;\geq\; 0. \tag{16.2}$$

excess free energy = KL to equilibrium

Every distribution sits above equilibrium by an excess free energy, and the excess is its Kullback–Leibler divergence to the Boltzmann distribution, priced in units of T. Relaxation drains this excess monotonically, and the proof needs only the machinery from the previous lecture. Write the joint law of one chain step in two ways: seeded at ρ, it is $q(x,x') = \rho(x)\,W(x\to x')$; seeded at equilibrium, $r(x,x') = p_T(x)\,W(x\to x')$. In the ratio q/r the kernels cancel, so $D_{\mathrm{KL}}(q\|r) = D_{\mathrm{KL}}(\rho\|p_T)$. Now decompose the *same* divergence by conditioning on the endpoint instead: the marginals are ρW and $p_T W = p_T$, the equilibrium conditional $r(x \mid x')$ is the previous lecture's adjoint kernel (Equation 15.3, making its second appearance), and the chain rule for KL gives

$$D_{\mathrm{KL}}(\rho\|p_T) \;=\; D_{\mathrm{KL}}(\rho W\,\|\,p_T) + \underbrace{\big\langle D_{\mathrm{KL}}(q(\cdot\mid x')\,\|\,r(\cdot\mid x'))\big\rangle_{\rho W}}_{\geq 0} \;\Longrightarrow\; D_{\mathrm{KL}}(\rho W\,\|\,p_T) \;\leq\; D_{\mathrm{KL}}(\rho\,\|\,p_T).$$

Each sweep of the chain lowers the excess free energy, with equality only at equilibrium — a discrete H-theorem, and the information theorist's data-processing inequality in physical units. An anneal is now a sequence of these drains against moving targets: at each stage the chain sheds divergence toward p_{T_k}, then the cooling step sharpens the target and re-prices what remains. End the schedule before the drain completes and the residual $D_{\mathrm{KL}}(\rho_{\mathrm{end}}\|p_{T_{\mathrm{end}}})$ is frozen in: by Equation 16.2 it is exactly the glass's excess free energy over the crystal, and the quench trace of Figure 16.4 flattening above the dotted ground-state line is this residual, seen in its energy component. Week 9 takes this monotone descent to continuous time, where it becomes the H-theorem of the Fokker–Planck equation, and Week 11 refines the accounting to single trajectories, where the drained divergence acquires the name *entropy production* and the frozen residual returns as the *dissipated work* of a finite-time protocol.

The guaranteed schedule — and its price. Geman and Geman (1984) made "slow enough" precise:

$$T(t) \;\geq\; \frac{c}{\log(1+t)} \quad\Longrightarrow\quad \text{convergence to the global minima, almost surely} \tag{16.3}$$

logarithmic cooling (guaranteed)

where the constant c must be at least the depth of the deepest barrier around a non-global minimum — that sharp, necessary-and-sufficient value is Hajek (1988)'s; Geman and Geman (1984)'s original proof guaranteed convergence for a larger $c \sim N\Delta$ (sites times energy range) — and the schedule must additionally send $T(t)\to 0$, since a constant temperature satisfies the inequality as written yet converges to nothing. The theorem is genuine, and its fine print undoes it in practice: inverting the schedule, reaching temperature T takes $t \sim e^{c/T}$ steps — the same exponential as the barrier crossings the schedule must wait for. A guarantee that costs the age of the universe on a modest instance is a guarantee in name only; nobody anneals logarithmically. What practice uses instead is

$$T_{k+1} = \alpha\, T_k, \qquad \alpha \in (0.8,\, 0.99) \tag{16.4}$$

geometric cooling (practical)

— a fixed multiplicative decay per stage, fast, tunable, and backed by no theorem at all. The pairing is a shape this course keeps meeting: the schedule with the proof is unusable, and the schedule in use is unproven. Simulated annealing is a powerful *heuristic* — it beats greedy descent wherever barriers matter — but it is not a guaranteed global optimizer at any practical cooling rate, and results obtained with it should be reported with that in mind. (One notational flag while the symbols are fresh: α here is the cooling factor, not Week 2's storage load; and $T(t)$ is now a *program*, not a parameter — $\beta = 1/T$ climbs as the schedule runs.)

Trap

$T = 0$ is a limit, not a temperature — and the guarantee is not a license. There are two traps. ① Sampling *at* $T = 0$ is greedy descent: on the example glass it fails 71% of the time. The optimizer is the *approach* — the schedule — never the endpoint; a chain parked at any fixed low T simply inherits the previous lecture's exponential mixing time. ② The logarithmic guarantee Equation 16.3 requires $t \sim e^{c/T}$ steps with c set by the deepest barrier — real and unusable. Every practical anneal is a heuristic, and a reported "optimum" from geometric cooling is a best-found, not a certificate.

Take-home 2

You cannot sample at $T = 0$ (greedy descent, trapped); you must cool, and no faster than the chain mixes at each temperature — a clock that itself slows as $e^{\Delta B/T}$. Logarithmic cooling $T \geq c/\log(1+t)$ guarantees the global minimum (Geman and Geman (1984)) at an exponential, unusable price; geometric cooling $T_{k+1} = \alpha T_k$ is the practical, unguaranteed compromise. Cooling faster than mixing is a quench: frozen in a glass, exactly where mixing is worst — through T_c and in rugged landscapes.

16.4 The bridge: the learning rate is a temperature

High temperature buys *exploration* — the chain accepts bad moves and visits basins it would never enter greedily; low temperature buys *exploitation* — it refines what it has. Annealing is the disciplined handover from one to the other, and stated this way the idea is not about spins at all: every stochastic optimizer manages the same trade, and every practitioner of deep learning already schedules a knob that does exactly this. The correspondence is literal.

Consider noisy gradient descent on a loss $L(\theta)$, with the noise injected deliberately:

$$\theta_{t+1} \;=\; \theta_t \;-\; \eta\, \nabla L(\theta_t) \;+\; \sqrt{2\,\eta\, T}\;\xi_t, \qquad \xi_t \sim \mathcal{N}(0, \mathbb{1}),$$

a downhill step of size η plus a Gaussian kick. This is the discretization of *Langevin dynamics*, and Week 9's engine (the Fokker–Planck equation) will prove what today we only state: its stationary distribution is $\propto e^{-L(\theta)/T}$ — a Boltzmann distribution over the loss landscape, with the loss as energy. Run at $T = 1$ with mini-batch

gradients, this is precisely *stochastic gradient Langevin dynamics* (Welling and Teh 2011), a sampler for Bayesian posteriors built from an optimizer; the tutorial compares its samples to an exact posterior. Now delete the injected noise and run plain SGD. The randomness does not vanish — the mini-batch gradient is itself a noisy estimate, with a covariance set by the data and the batch size, *independent of* η. Carried through the same dynamics, drift scales as η while the noise's variance scales as η^2, and the ratio that plays the role of temperature comes out proportional to the step size:

$$\boxed{T_{\text{eff}} \propto \eta \qquad \Longrightarrow \qquad \text{a learning-rate schedule is a cooling schedule}} \tag{16.5}$$

temperature learning rate

Decaying η_t over training *is* annealing on the loss (Figure 16.3): the large early steps are the hot phase, roaming among basins; the small late steps are the cold phase, committing to one and refining it. The warm-up schedules, step decays, and cosine schedules of deep-learning practice are cooling schedules, tuned by exactly the trade this lecture formalized — and the quench has a training-room name too: drop the learning rate too early and the run freezes into whichever basin it happened to occupy, the optimizer's glass. One caveat: for SGLD the Boltzmann stationary distribution is a theorem (in the decreasing-step limit; at a fixed step there is an $\mathcal{O}(\eta)$ discretization bias); for plain SGD, $T_{\text{eff}} \propto \eta$ is today a scaling argument, and turning it into a precise statement — what exactly SGD's mini-batch noise does, when the Boltzmann form holds, and what it says about flat minima and generalization — is the entire business of Week 9. (Two script-only pointers for the curious: *parallel tempering*, the modern cure for glassy annealing — run replicas at a ladder of temperatures and swap them, so cold chains borrow the hot chains' mobility; and *annealed importance sampling*, Week 11, where a cooling schedule is used not to optimize but to estimate Z itself.)

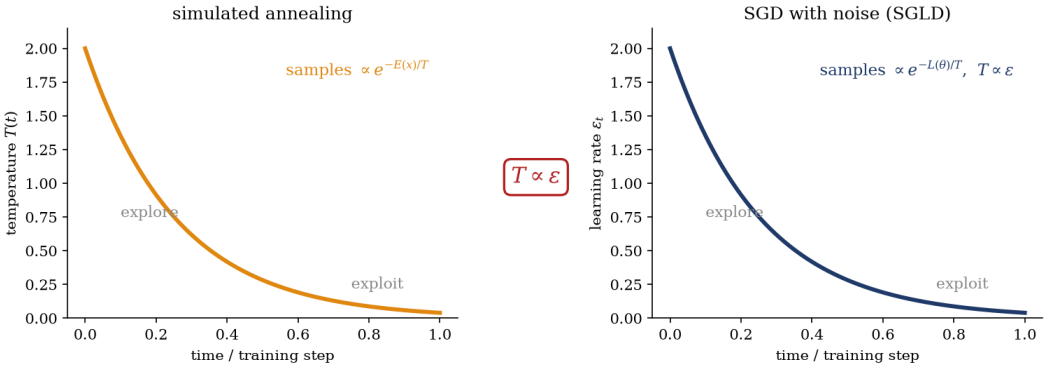

Figure 16.3: The same knob, two rooms. Left: simulated annealing schedules a temperature $T(t)$ downward; the chain samples $\propto e^{-E(x)/T}$ and passes from exploration to exploitation. Right: SGD with a decaying learning rate η_t does the same on a loss landscape — Langevin dynamics samples $\propto e^{-L(\theta)/T}$ with $T \propto \eta$, so the learning-rate schedule is a cooling schedule. Week 9 derives the correspondence; the tutorial runs it.

Take-home 3

Temperature trades exploration for exploitation, and in learning that knob is the *learning rate*: Langevin dynamics with step η and temperature T has

stationary distribution $\propto e^{-L/T}$ (SGLD, Welling and Teh (2011)), and plain SGD's mini-batch noise gives $T_{\text{eff}} \propto \eta$. Decaying the learning rate is cooling; SGD with a decaying step is simulated annealing on the loss; and dropping the rate too early is a quench. Week 9 turns the scaling argument into a theorem.

16.5 Example: anneal versus quench, on sixteen spins

A single enumerable instance illustrates the claims. Take the spin glass of Figure 16.2 — $N = 16$, $J_{ij} \sim \mathcal{N}(0, 1/N)$, all 65,536 states enumerated, ground-state energy $E^*/N = -0.610$ (doubly degenerate, the global flip) and 28 local minima — and anneal it with Metropolis under geometric cooling from $T = 2.0$ to 0.02, giving the schedule different total durations. Every run uses the *same schedule shape*; only the clock changes. For contrast, a second instance from the same ensemble ("glass B", $E^*/N = -0.599$) whose dominant barrier happens to be deeper. Fifty independent runs per point; Figure 16.4 shows the traces and the verdict, and the fractions of runs that end in the true ground state read:

total sweeps	3	10	30	100	300	1000	3000	greedy ($T = 0$)
glass A	0.10	0.28	0.50	0.66	0.88	0.98	1.00	0.29
glass B	0.10	0.16	0.16	0.26	0.24	0.44	0.60	0.20

Slower cooling monotonically improves the crystal: on glass A the success rate climbs from 10% for a three-sweep quench to *certainty* at three thousand sweeps — the anneal–quench crossover of Section 16.3 as a measured curve. The greedy column calibrates both ends: pure descent manages 29%, roughly a third of a patient anneal's success rate, yet still beats the fastest quenches — which end their schedule before they have finished descending at all. And glass B makes Hajek's constant c concrete: same size, same ensemble, one deeper barrier, and every entry drops — at three thousand sweeps, where glass A is solved, glass B still fails four times in ten. The schedule a landscape demands is set by its deepest barrier, exponentially; no single cooling rate is "slow enough" in general, which is why Equation 16.3's guarantee had to be logarithmic and why annealing remains a heuristic. In the traces (left panel), the story is visible directly: the quench's median energy flattens at -0.56 per spin, an excited glass, while the anneal's median lands on the dotted ground-state line at -0.610.

Anneal to optimize: cool slowly enough to stay in equilibrium — a quench traps you in a glass, and a learning-rate schedule is a cooling schedule.

The tutorial (16–18, Ph12 106) is derivation-led. Two problems. First, simulated annealing on the travelling salesman problem — you will watch a tour untangle as the temperature drops, and choose the schedule yourselves; The warm-up card already gave you Kirkpatrick, Gelatt, and Vecchi (1983), where that exact problem earned the method its name. Second, stochastic gradient Langevin dynamics against

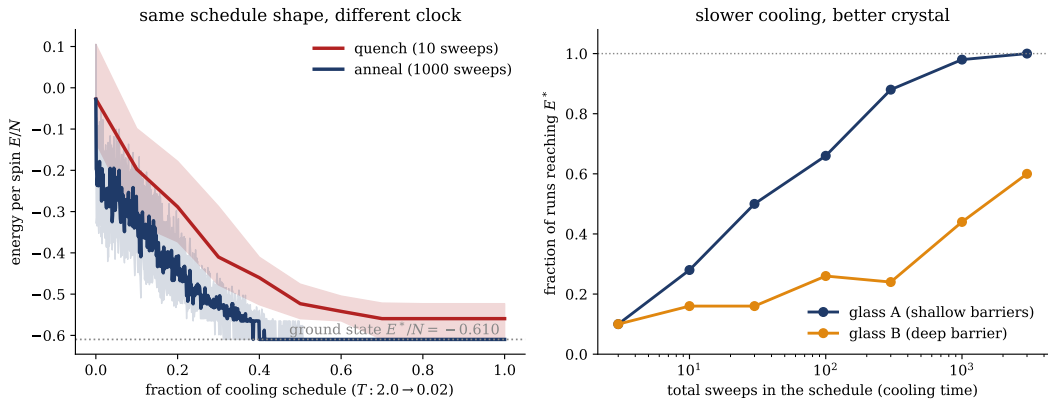

Figure 16.4: Anneal versus quench on glass A ($N = 16$; exact $E^*/N = -0.610$ from enumeration; geometric cooling $T : 2.0 \to 0.02$). Left: median energy per spin with interquartile band over 50 runs, plotted against the *fraction* of the schedule — the quench (red, 10 sweeps total) and the anneal (navy, 1000 sweeps) follow identical schedule shapes with different clocks. The quench freezes at -0.56 per spin, an excited local minimum; the anneal tracks equilibrium and lands on the ground state. Right: the fraction of 50 runs ending in the true ground state against the total schedule length, for glass A and for a second instance with a deeper dominant barrier. Slower cooling always helps; how slow is *enough* is set by the deepest barrier — glass B still fails 40% of the time at a budget that solves glass A completely. The metallurgical claim — slow cooling makes crystals, fast cooling makes glasses — is here a measured curve.

an exact Bayesian posterior — take Equation 16.5's claim into the room and test it on a model small enough to integrate by hand. The tutorial requires no devices.

16.6 Outlook: from cooling to training

Week 8 closes with the sampling road fully open and its first two uses in hand. The previous lecture established that the Boltzmann distribution can be sampled without its normalizer, at the price of mixing; today made the sampler's one knob into an instrument, and the price reappeared as the quench — every claim of this lecture was ultimately a claim about mixing under a deadline. That the resulting algorithm was invented by a spin-glass physicist is not incidental: the rugged landscapes where annealing is both necessary and hard are the same ones Weeks 2 and 6 mapped, and the metallurgical vocabulary — crystal, glass, quench — has been the course's vocabulary for memory and optimization since Figure 3.1.

Week 9 is the module's destination, and today's bridge is its on-ramp. The claim $T_{\text{eff}} \propto \eta$ was offered as a scaling argument; next week supplies the machinery — Langevin dynamics as the continuum limit of noisy descent, the Fokker–Planck equation as its law of motion, and the Boltzmann distribution over the loss as its stationary state — and with it the striking consequences: why SGD's noise selects *flat* minima, what that has to do with generalization, and why the training hyperparameters everyone tunes are thermodynamic variables under other names. Training is sampling; this week built the sampler, and next week reads the training loop as one.

The longer thread runs to the course's title. Cooling as a computational principle does not end with optimization: Week 11 anneals to *estimate* the very Z this module dodges (annealed importance sampling — a bridge of intermediate temperatures between a tractable distribution and the target), and Week 12's diffusion models are, at heart, an annealing schedule run as a generative process — noise added along a schedule and removed along the reverse one, with a learned denoiser standing where the Metropolis rule stands today. The annealing framework, introduced in a 1983 *Science* paper, remains active through the final lecture.

17 SGD as Langevin dynamics: the temperature of training

Recommended reading

Core (~1 h). Mandt, Hoffman, and Blei (2017), §3–§4 — the warm-up reading and the lecture's modern backbone: SGD near a minimum as an Ornstein–Uhlenbeck process, its stationary distribution, and the learning rate read as a temperature. Alongside it, the SGD/Langevin sections of Mehta et al. (2019), in the course's notation.
Optional background. Langevin (1908) — three pages of 1908 French, the original equation for a particle buffeted by thermal noise in a potential; the ancestor of everything below. Risken (1989), Ch. 4–5 — the Langevin → Fokker–Planck → stationary-measure machinery done rigorously, where our physical derivation is done carefully. Chaudhari and Soatto (2018) — the sober account of what real SGD noise does to the clean picture: anisotropy, no potential, limit cycles; read after the lecture, not before.
Prerequisite reminder. The Boltzmann distribution Equation 1.2 and the free-energy toolkit of Week 1; the Gaussian integral Equation 2.1 and the sums-of-many-terms reasoning behind it (the mini-batch gradient is such a sum); and Week 8 in full — the Metropolis sampler (Section 15.3), simulated annealing, and SGLD's temperature learning-rate bridge (Welling and Teh 2011), which today stops being an analogy. New content, built from scratch: the Langevin SDE for SGD, the Fokker–Planck equation and its probability current, the stationary Boltzmann solution, and the H-theorem that makes it the destination — the dynamics-side machinery this module is named for.

17.1 Training is sampling

Week 8 ended on a bridge built deliberately: SGLD adds Gaussian noise to a gradient step and thereby *samples* $e^{-L/T}$ at a temperature set by that injected noise (Welling and Teh 2011) — while the same lecture's closing scaling argument already hinted that plain SGD carries a temperature set by the step size itself. But SGLD injects its noise explicitly, and the bridge appeared to depend on that injection. It does not. Ordinary stochastic gradient descent — the algorithm training every network in this course, with no noise added by anyone — is *already* a noisy dynamics, because each step evaluates the gradient on a random mini-batch rather than the full dataset. A mini-batch gradient is the true gradient plus a zero-mean fluctuation, and a particle descending a potential under zero-mean kicks is not performing optimization as the textbooks draw it. It is performing **Brownian motion in a potential** — the problem Langevin (1908) wrote the equation for.

That reframe is the lecture's central idea, and it has a long history. In 1908 Langevin described a Brownian particle by Newton's equation plus two new forces, a friction and a random thermal kick, and showed the pair reproduces Einstein's diffusion. **Fokker (1914)** and **Planck (1917)** then wrote the complementary equation: not for the particle's trajectory but for the *evolution of its probability distribution*, whose long-time limit is the Boltzmann distribution. A century later, Mandt, Hoffman, and Blei (2017) observed that the SGD iterate satisfies Langevin's equation with the loss as the potential and the mini-batch fluctuations as the thermal kicks — so the Fokker–Planck equation governs the distribution of the parameters during training, and its stationary solution answers a question we did not know we were allowed to ask: *what distribution does training converge to?*

The stationary distribution turns out to be a Boltzmann distribution over the loss, $p_\infty \propto e^{-L/T}$, at a temperature $T \propto \eta/|B|$ — the learning rate over the batch size. The Boltzmann law of Week 1 returns a final time in this module, now with the parameters θ as the configuration and the loss L as the energy. Training *is* sampling; SGD thermalizes around a minimum rather than coming to rest at it; and the two most-tuned hyperparameters in deep learning are two dials on a single physical quantity. The derivation occupies the two engine sections; the consequences and caveats follow, and the tutorial measures the resulting prediction.

> Take-home 1
>
> SGD is not deterministic descent: the random mini-batch makes its gradient noisy, and noisy descent in a potential is Langevin (Brownian) dynamics — θ a particle, the loss a potential, the mini-batch fluctuations thermal kicks. SGD fluctuates around minima rather than resting at them. Training is sampling, and the learning rate sets the temperature.

17.2 Optimizer or sampler

The same algorithm admits two readings. Read as an *optimizer*, SGD's noise is a nuisance: it prevents convergence to the minimizer, and the practitioner anneals it away. Read as a *sampler*, the noise is the mechanism: SGD draws from a distribution concentrated on low loss, the noise strength decides how concentrated, and "convergence" means the *distribution* has stopped changing, not the iterate. Both readings are correct; the sampler reading is the one with predictive power, because it attaches a number — a temperature — to the vague notion of "how much SGD explores."

Figure 17.1 draws the sampler reading before any equation. The iterate descends into a well of the loss and then does not stop: it jitters, indefinitely, in a stationary cloud around the minimum. The cloud's width will come out as $\sqrt{T/\lambda}$ — temperature over local curvature — so the same temperature makes a narrow cloud in a steep minimum and a wide one in a flat minimum. Both knobs of practice collapse into that single T: raising the learning rate η heats the dynamics, and enlarging the batch $|B|$ — averaging away more of the gradient noise — cools it. The whole phenomenology of "large batches converge sharply, small batches explore" is one statement about one number.

A distinction about spaces is essential here, because the course's spine depends on it. Today's dynamics runs in *parameter* space: the variable is θ, the gradients are

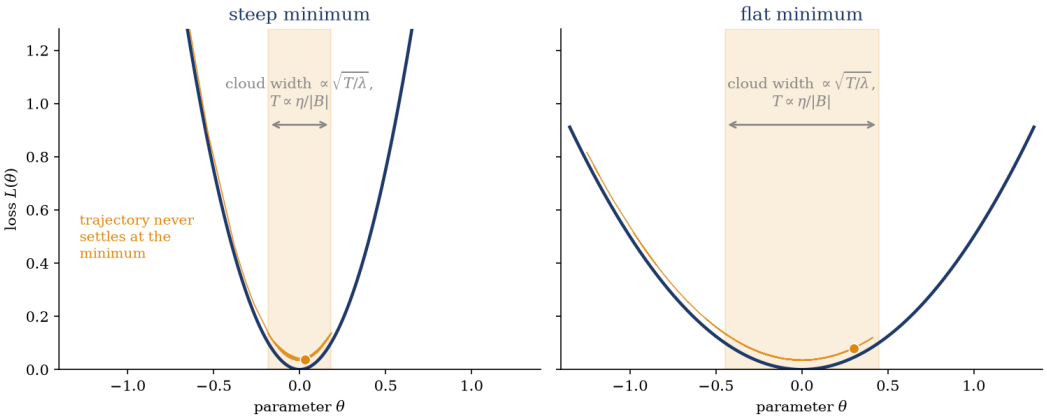

Figure 17.1: The central picture of the lecture: SGD as a Brownian particle in the loss landscape. In both panels the iterate (orange trajectory) falls into the well and then never settles — it thermalizes into a stationary cloud (shaded band) around the minimum, of width $\propto \sqrt{T/\lambda}$ for local curvature λ, with the temperature $T \propto \eta/|B|$ set by the learning rate over the batch size. Left: a steep minimum confines the cloud tightly. Right: at the same temperature, a flat minimum leaves a wide cloud — the geometry that becomes the flat-minima story of the next lecture.

∇_θ, and the energy is the loss — the same space where Week 3 met its training obstruction $\nabla_\theta \log Z$ (Equation 5.5). It is not the *configuration*-space dynamics over data x that Week 10 will need for score matching, where the object is ∇_x and the partition function conveniently drops out. Same machinery, two different spaces; we will run it in the other space soon enough. Two notational points: W below is the Wiener process, not a weight matrix; and T is a *derived* property of the algorithm, not an external thermostat — nobody set it, but it is there.

17.3 The engine, part I: the noise in the gradient

The empirical loss is an average over the dataset, $L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(\theta)$, and a full gradient costs N per-sample gradients. SGD instead draws a random mini-batch $B \subset \{1, \dots, N\}$ and uses

$$\nabla \hat{L}_B(\theta) \;=\; \frac{1}{|B|}\sum_{i\in B} \nabla \ell_i(\theta),$$

an average of $|B|$ randomly selected per-sample gradients. Averages of many random terms are Week 1's home ground: this estimator is *unbiased*, $\langle \nabla \hat{L}_B \rangle = \nabla L$, and its fluctuations shrink as $1/|B|$ in covariance. Writing $C(\theta) = \frac{1}{N}\sum_i (\nabla \ell_i - \nabla L)(\nabla \ell_i - \nabla L)^{\mathsf{T}}$ for the covariance of the per-sample gradients — the dataset's intrinsic gradient spread at θ —

$$\nabla \hat{L}_B \;=\; \nabla L + \xi, \qquad \langle \xi \rangle = 0, \qquad \mathrm{Cov}(\xi) \;=\; \frac{C(\theta)}{|B|}, \tag{17.1}$$

mini-batch gradient noise

and for $|B|$ reasonably large the central limit theorem makes ξ Gaussian. (For sampling without replacement a finite-population factor $1 - |B|/N$ multiplies the covariance; for the usual $|B| \ll N$ it is invisible and we drop it.) Figure 17.2 shows what Equation 17.1 looks like: a scatter of mini-batch gradients around the true one, tightening as the batch grows — and, in general, spread *unequally* across directions, a detail that returns in Section 17.5 with consequences.

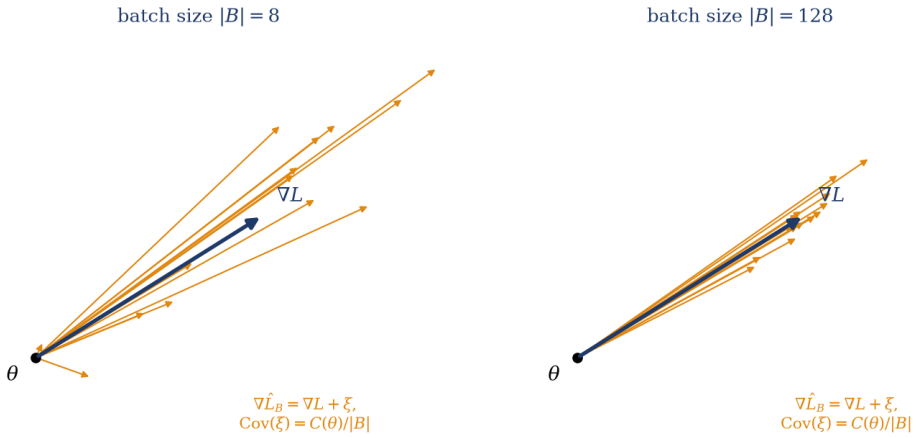

Figure 17.2: The engine's first picture: the mini-batch gradient is the full gradient plus zero-mean noise. From the same parameter point θ, fourteen random mini-batches give fourteen gradient estimates (orange) scattered around the full-batch gradient ∇L (navy), with covariance $C(\theta)/|B|$. Left: $|B| = 8$ — individual batches can point far off, some even uphill. Right: $|B| = 128$ — the scatter tightens as $1/\sqrt{|B|}$. This noise is the thermal kick that turns descent into Langevin dynamics, and its shape (the same in both panels, only its scale differs) decides whether SGD reaches a true equilibrium.

From step to stochastic differential equation. The SGD update with learning rate η is then drift plus kick,

$$\theta_{k+1} \;=\; \theta_k - \eta\,\nabla\hat{L}_B(\theta_k) \;=\; \theta_k \underbrace{-\,\eta\,\nabla L(\theta_k)}_{\text{drift}} \underbrace{-\,\eta\,\xi_k}_{\text{noise, } \mathrm{Cov}=\eta^2 C/|B|}\,,$$

and the continuous-time limit requires one bookkeeping decision: how much time does one step represent? Take $\Delta t = \eta$. This is the convention that makes the limit clean — the drift per unit time becomes $-\nabla L$, independent of η, so the entire η-dependence is pushed into the noise (Mandt, Hoffman, and Blei 2017). The noise contributes covariance $\eta^2 C/|B|$ per step, i.e. per time $\Delta t = \eta$; a diffusion with tensor D accumulates covariance $2D\Delta t$ over the same interval; matching the two gives $D = \eta C/(2|B|)$. For isotropic noise, $C = c\,I$, the diffusion tensor is a pure temperature, $D = T I$, and SGD is revealed as a discretization of the *Langevin equation*:

$$\mathrm{d}\theta \;=\; -\nabla L(\theta)\,\mathrm{d}t \;+\; \sqrt{2T}\,\mathrm{d}W, \qquad T \;=\; \frac{\eta\, c}{2|B|} \;\propto\; \frac{\eta}{|B|}, \tag{17.2}$$

SGD as Langevin dynamics

with W the Wiener process — the mathematical idealization of Brownian kicks. The temperature is the learning rate over the batch size, times the dataset's own gradient

variance c. Hot training means large steps or small batches; cold training means small steps or large batches; and the incidental noise has a thermodynamic name. One caveat rides along and is settled in the example: the SDE is the *small-η limit* of the discrete update — a fixed-small-η model, strictly, since the true $\eta \to 0$ limit sends the temperature $T \propto \eta$ to zero along with it — and at practical learning rates the discreteness leaves measurable corrections.

17.4 The engine, part II: Fokker–Planck, and the Boltzmann stationary state

The Langevin equation describes one noisy trajectory; the question that matters — what does training converge to? — is about the *distribution* $p(\theta, t)$ of the iterate, and needs the equation Fokker and Planck supplied. We derive it the physical way, from conservation of probability; the rigorous route (Itô calculus, the Kramers–Moyal expansion) reaches the same equation and lives in Risken (1989).

Probability is conserved — the iterate is always somewhere — so p obeys a continuity equation, $\partial_t p = -\nabla \cdot J$, and the physics enters through the **probability current** $J(\theta, t)$: how probability flows. The Langevin dynamics Equation 17.2 moves probability in two ways. The drift carries it downhill at velocity $-\nabla L$, contributing a current $-p\,\nabla L$; the noise spreads it from dense regions to empty ones, contributing Fick's diffusion current $-T\,\nabla p$. Their sum is the total current, and the continuity equation becomes the *Fokker–Planck equation*:

$$\boxed{\partial_t p \;=\; -\nabla \cdot J \;=\; \nabla \cdot (p\,\nabla L) + T\,\nabla^2 p, \qquad J \;=\; -p\,\nabla L - T\,\nabla p.} \tag{17.3}$$

Fokker–Planck equation

Every term is readable. The first transports the distribution toward low loss — pure optimization. The second is diffusion at temperature T — pure exploration. Training, in this equation, is literally a competition between descending and spreading, and the stationary state is where the two currents cancel.

The stationary solution. A stationary distribution has $\partial_t p = 0$, i.e. $\nabla \cdot J = 0$. The *equilibrium* solution does better and sets the current itself to zero everywhere — no flow at all, the detailed-balance condition. (The distinction between $J = 0$ and $\nabla \cdot J = 0$ is about to matter.) Setting $J = 0$ in Equation 17.3,

$$p\,\nabla L + T\,\nabla p = 0 \quad\Longrightarrow\quad \nabla \log p = -\frac{\nabla L}{T} \quad\Longrightarrow\quad \boxed{p_\infty(\theta) \;\propto\; e^{-L(\theta)/T}.} \tag{17.4}$$

stationary Boltzmann distribution over the loss

Run long at fixed η and $|B|$, SGD's iterate is distributed as a Boltzmann distribution over the loss at temperature $T \propto \eta/|B|$. The distribution that opened this course (Equation 1.2) closes the module's circle: the configuration is now the network's parameters, the energy is the loss, and the algorithm every practitioner runs is the sampler. Near a minimum θ^* with Hessian H, expanding L to second order makes Equation 17.4 a Gaussian with covariance $\Sigma = TH^{-1}$: a cloud of linear size $\sqrt{T/\lambda}$ along each Hessian eigendirection — the shaded bands of Figure 17.1, now with their exact widths.

From fixed point to destination: the H-theorem. Setting $J = 0$ verified that the Boltzmann distribution is *a* stationary state; it did not show the dynamics *goes*

there. The missing statement is a Lyapunov argument, classical enough to carry a nineteenth-century name, and it takes four lines (Welling, Lu, and Holdijk 2026, ch. 21 on the Fokker–Planck equation). Track the Kullback–Leibler divergence of the evolving distribution from the candidate equilibrium, $D(t) = D_{\mathrm{KL}}(p_t \,\|\, p_\infty)$ — by last week's identity Equation 16.2, the excess free energy in units of T. Two identities precede the computation. First, the current of Equation 17.3 collapses against the equilibrium into a single gradient: since $\nabla \log p_\infty = -\nabla L/T$,

$$J \;=\; -p\,(\nabla L + T\,\nabla \log p) \;=\; -T\,p\,\nabla \log \frac{p}{p_\infty}.$$

Second, $\int \mathrm{d}\theta\; \partial_t p = 0$, so the constant in $\log(p/p_\infty) + 1$ drops when we differentiate. Then, inserting the continuity equation and integrating by parts,

$$\frac{\mathrm{d}D}{\mathrm{d}t} \;=\; \int \mathrm{d}\theta\; \partial_t p\, \log \frac{p}{p_\infty} \;=\; \int \mathrm{d}\theta\; J \cdot \nabla \log \frac{p}{p_\infty} \;=\; -T \int \mathrm{d}\theta\; p \left\| \nabla \log \frac{p}{p_\infty} \right\|^2 \;\le\; 0,$$

the H-theorem: KL to equilibrium only decreases

with equality only when $\nabla \log(p/p_\infty)$ vanishes everywhere the iterate has probability — that is, only at $p = p_\infty$ itself. The divergence to the Boltzmann distribution decreases monotonically and stops decreasing nowhere else, so the fixed point is the destination: from any start, $p_t \to e^{-L/T}$ (granted the usual regularity — a confining loss, boundary terms that vanish). Read thermodynamically through Equation 16.2, the same display says the excess free energy drains at the explicit rate $T^2 \int p \,\|\nabla \log(p/p_\infty)\|^2$ — the continuous-time twin of the kernel-by-kernel drain Week 8 proved for the quench, and the integrand is a quantity Week 11 will meet under its own name, the *entropy-production rate*. A scope limitation: collapsing J into a single gradient used the *isotropic* current, and with an anisotropic diffusion tensor the manipulation survives only when $D^{-1}\nabla L$ is itself a gradient — exactly the condition of the next section. Where that fails, no functional of this form decreases, nothing plays the role of p_∞, and the persistent circulation of the non-equilibrium steady state is the visible symptom of the missing Lyapunov function.

Figure 17.3 integrates Equation 17.3 numerically in a double-well loss and shows the full relaxation process: probability pours down the walls, thermalizes within the first well it finds, leaks across the barrier by diffusion, and settles onto $e^{-L/T}$ — including the equal population of the second well that pure gradient descent, started on the right, would never touch.

> **Take-home 2**
>
> The iterate's distribution obeys the Fokker–Planck equation $\partial_t p = \nabla \cdot (p \nabla L) + T\nabla^2 p$ — descent transporting probability against diffusion spreading it — and its equilibrium, where the probability current vanishes, is the Boltzmann distribution $p_\infty \propto e^{-L/T}$ (Equation 17.4). SGD samples a Boltzmann distribution over the loss, thermalizing into a cloud of covariance TH^{-1} around a minimum rather than reaching it.

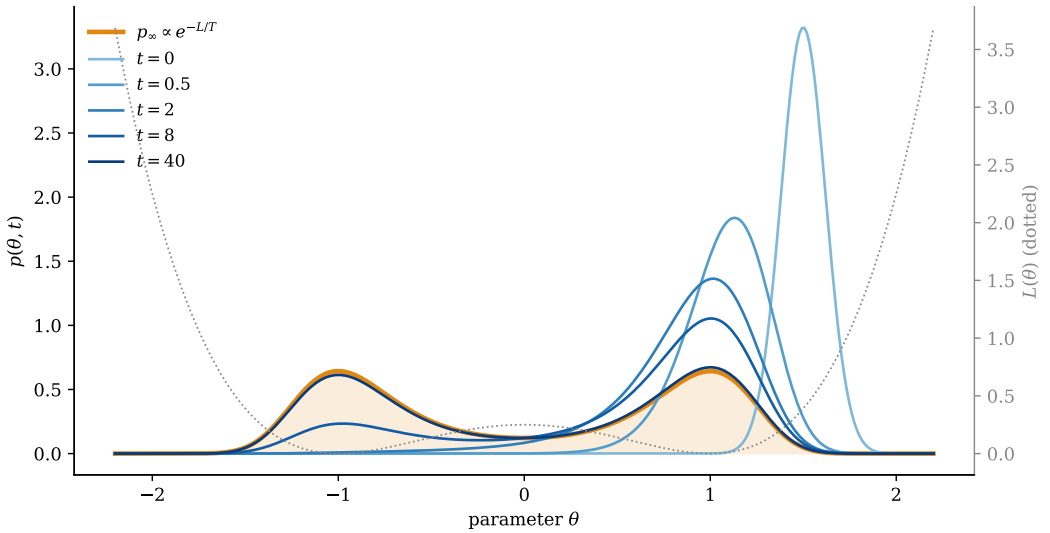

Figure 17.3: The Fokker–Planck equation Equation 17.3 solved numerically in the double-well loss $L(\theta) = (\theta^2-1)^2/4$ (gray dotted, right axis) at $T = 0.15$, starting from a narrow distribution on the right wall ($t = 0$). Drift pours the probability into the right well ($t = 0.5$), where it thermalizes to a local Boltzmann shape ($t = 2$); diffusion then carries mass over the barrier — height $0.25 \approx 1.7\,T$ — into the left well ($t = 8$), until the current vanishes everywhere and the density settles onto the stationary Boltzmann law $p_\infty \propto e^{-L/T}$ (orange): by $t = 40$ the two agree to within 5%, with 0.478 of the mass in the left well against the exact 0.5. Gradient descent started at the same point would sit at $\theta = 1$ forever; the temperature explores both wells.

17.5 The temperature of training, and where the picture bends

What $T \propto \eta/|B|$ buys. Three practical facts, previously empirical, are now one thermodynamic statement. First, the **linear scaling rule** of large-batch training — when you double the batch size, double the learning rate — is exactly the statement that only the ratio $\eta/|B|$ matters: it holds the temperature fixed. Second, batch size is not thermodynamically neutral: growing $|B|$ at fixed η *cools* the dynamics, which is why large-batch training converges sharply but explores little — a fact with generalization consequences that the next lecture takes up. Third, Week 8's annealing schedules translate verbatim: decaying the learning rate during training is *cooling this sampler*, walking $T \to 0$ so the stationary cloud contracts onto a minimum. The practitioner's learning-rate schedule and the physicist's cooling schedule are, by Equation 17.2, the same object — SGLD (Welling and Teh 2011) made that identification by design, adding calibrated isotropic noise; plain SGD arrives at it for free, with noise it already had.

Where it bends. The derivation of Equation 17.4 slipped in one assumption: *isotropic* noise, $C = cI$, so that the diffusion is a scalar temperature. Real gradient noise fails this, structurally. The covariance $C(\theta)$ is built from the data, varies across parameter space, and near minima aligns its large directions with the large-curvature directions of the loss — the noise kicks hardest exactly where the landscape is steepest. With an anisotropic diffusion tensor D, the equilibrium condition $J = 0$ requires $\nabla L = -D\,\nabla \log p$, which demands that $D^{-1}\nabla L$ be itself a gradient — and for generic D and L it is not. Then *no* distribution makes the current vanish: the best available stationary state has $\nabla \cdot J = 0$ with $J \neq 0$ — probability circulating forever, a *non-equilibrium* steady state, exactly the case anticipated by the distinction planted under Equation 17.3. Chaudhari and Soatto (2018) push this to its conclusion: for deep networks, SGD is out of equilibrium, its stationary distribution is not $e^{-L/T}$ for any effective loss, and its dynamics can orbit in limit cycles rather than rest. The temperature picture survives as the *leading, isotropic approximation* — genuinely predictive, as the example below shows, and genuinely incomplete.

The near-minimum truth. One regime stays exactly solvable and gives the tutorial its target. Near a quadratic minimum, $L \approx \frac{1}{2}\theta^{\mathsf{T}} H \theta$, the Langevin process is the *Ornstein–Uhlenbeck process*, and its stationary covariance Σ solves the **Lyapunov equation**

$$H\Sigma + \Sigma H \;=\; 2D, \qquad D = \frac{\eta\, C}{2|B|}, \tag{17.5}$$

stationary covariance, general noise

which collapses to the Boltzmann answer $\Sigma = TH^{-1}$ exactly when $C \propto I$ and not otherwise. This equation — not the clean Boltzmann law — is what a measurement should be compared against, and measuring its ingredients is the tutorial's assignment: estimate $C(\theta)$ on a real network, check its isotropy, and test the predicted stationary cloud.

Trap

Three ways to misuse a temperature. ① The clean $p_\infty \propto e^{-L/T}$ requires *isotropic* gradient noise. Real SGD noise is anisotropic and state-dependent, so

the true stationary state solves Equation 17.5 near minima and is in general non-equilibrium, with persistent probability currents and no potential (Chaudhari and Soatto 2018) — do not claim SGD exactly samples $e^{-L/T}$. ② Both knobs move $T \propto \eta/|B|$: changing the batch size at fixed learning rate changes the temperature (the linear scaling rule compensates precisely this). ③ Today's dynamics lives in *parameter* space — ∇_θ, loss as energy — not the configuration-space ∇_x of Weeks 10–12. The same Langevin/Fokker–Planck machinery runs in both spaces; the objects it moves are entirely different.

Take-home 3

$T \propto \eta/|B|$ makes the learning rate and batch size two dials on one temperature: the linear scaling rule holds T fixed, large batches cool, and annealing the learning rate is cooling the sampler. But the Boltzmann law assumed isotropic noise — real gradient noise is anisotropic and state-dependent, the stationary covariance solves the Lyapunov equation $H\Sigma + \Sigma H = 2D$ rather than TH^{-1}, and in general SGD is a non-equilibrium process with persistent currents. The clean picture is the leading approximation; the tutorial measures the correction.

17.6 Example: SGD on a quadratic loss

The exactly solvable case, simulated in full, tests every claim above against a number. Take $L(\theta) = \frac{1}{2}\theta^{\mathsf{T}} H\theta$ in two dimensions with $H = \mathrm{diag}(4, 1)$ — one steep direction, one flat — and run plain SGD with synthetic gradient noise of controllable covariance C (batch size absorbed into C; $T = \eta/2$ for unit isotropic noise). Figure 17.4 shows the two experiments.

The left panel is the thermalization claim. An ensemble of 4000 runs starts from a single point; the ensemble variance of θ_1 rises and then *plateaus* — the iterate has stopped converging and started sampling. The plateau sits where Equation 17.4 puts it, $\mathrm{Var}(\theta_1) = T/\lambda_1$, and halving η halves it: measured 0.00259 and 0.00128 against predicted 0.00250 and 0.00125 for $\eta = 0.02$ and 0.01. This is annealing in miniature: each cooler curve is the same algorithm with a smaller cloud. The hottest run shows the promised discreteness correction — measured 0.00540 against the SDE's 0.00500, an 8% excess that the exact discrete-time calculation ($\mathrm{Var} = \frac{T}{\lambda}\frac{2}{2-\eta\lambda} = 0.00543$) accounts for almost entirely: the Langevin picture is the small-η limit, and at $\eta\lambda = 0.16$ the limit is already visibly approximate.

The right panel is the caveat made visible. With isotropic noise (navy) the stationary cloud matches the Boltzmann ellipse TH^{-1} — elongated along the flat direction, as Figure 17.1 promised. With anisotropic noise of the same average strength, tilted 45° to the Hessian axes (orange), the cloud tilts too: its covariance matches the Lyapunov solution of Equation 17.5 (correlation 0.71 measured, 0.72 predicted) and no Boltzmann distribution — no function of L alone — has that shape, since any $e^{-L/T}$ inherits the Hessian's axes (gray dashed). The red arrows complete the diagnosis: the stationary current $J = p\cdot(D\Sigma^{-1} - H)\,\theta$ does not vanish but *circulates* (its matrix has purely imaginary eigenvalues $\pm 1.57i$) — probability orbiting the minimum forever, a non-equilibrium steady state in the smallest possible example. The effect visible in two dimensions persists in real networks; measuring it there is the tutorial's work.

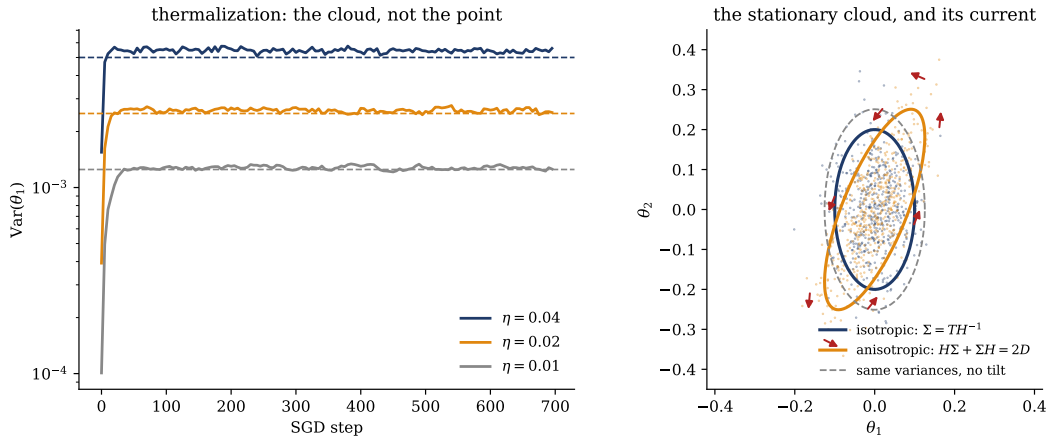

Figure 17.4: SGD on the quadratic loss $L = \frac{1}{2}\theta^{\mathsf{T}} H\theta$, $H = \mathrm{diag}(4, 1)$ — the exactly solvable (Ornstein–Uhlenbeck) case. Left: ensemble variance of θ_1 over 4000 runs for three learning rates (isotropic noise). Each run thermalizes into a stationary cloud whose variance (plateau) matches the Boltzmann prediction T/λ_1 with $T = \eta/2$ (dashed): 0.00259 vs. 0.00250 at $\eta = 0.02$, 0.00128 vs. 0.00125 at $\eta = 0.01$; halving the learning rate halves the cloud (annealing in miniature). At $\eta = 0.04$ the measured 0.00540 exceeds the SDE's 0.00500 by the discrete-step correction $2/(2-\eta\lambda_1)$, giving 0.00543 — the Langevin limit is a small-η statement. Right: stationary clouds at $\eta = 0.02$. Isotropic noise (navy points) fills the Boltzmann ellipse $\Sigma = TH^{-1}$, axis-aligned and wide along the flat direction. Anisotropic noise of equal average strength, tilted 45° (orange points), produces a tilted cloud matching the Lyapunov solution $H\Sigma + \Sigma H = 2D$ (correlation 0.71 measured vs. 0.72 predicted) — a shape no $e^{-L/T}$ can produce (gray dashed: the same variances without the tilt). Red arrows: the non-vanishing stationary current circulating around the minimum (eigenvalues $\pm 1.57i$) — non-equilibrium, in two dimensions.

17.7 Outlook: the measurement, the flat minima, and the reverse

The week continues from here. The warm-up card, due the evening before the next lecture, is unusually a recap: read the Ornstein–Uhlenbeck sections of Mandt, Hoffman, and Blei (2017) holding today's derivation — the paper is this lecture in the notation of the field that rediscovered it. The tutorial (16–18, Ph12 106) takes the theory to the laboratory: measure the mini-batch gradient-noise covariance $C(\theta)$ on a real network, test its isotropy, and compare the stationary fluctuations against Equation 17.5 — the prediction is on the table. The next lecture draws the consequence the whole picture has been pointing at: if training is sampling at a temperature, then the temperature decides *which* minima carry the stationary weight — wide, flat basins hold more Boltzmann mass than sharp ones of equal depth — and the flat-minima account of generalization, with entropy-SGD as its deliberate algorithmic version, follows.

The machinery extends well beyond this week. The Langevin equation and its Fokker–Planck companion are, in Week 10, transplanted from parameter space to *configuration* space — dynamics over data x rather than weights θ, where $\nabla_x \log Z = 0$ removes Module 1's obstruction outright — and in Week 12 they are run *backwards in time*: a diffusion that noises data into a Gaussian, inverted into a generative model. Every diffusion model is these equations, reversed — the road from sampling to generation runs through the same two boxed equations.

> *SGD is not descent — it is Langevin dynamics: training samples a Boltzmann distribution over the loss, and the learning rate is its temperature.*

18 Flat minima: generalization is a free energy

Recommended reading

Core (~1 h). Chaudhari et al. (2017) — entropy-SGD, the algorithm that turns this lecture's thesis into a training objective: read §3–§4 for the local free energy and its inner sampler. Alongside it, Keskar et al. (2017) — the empirical anchor: large-batch training finds sharp minima that generalize worse; read for the evidence, especially the sharpness–gap plots.
Optional background. Hochreiter and Schmidhuber (1997) — *Flat Minima*, the original argument (via minimum description length), made twenty years before deep learning could test it. Hinton and Camp (1993) — the MDL/Bayesian ancestor: free energy as a complexity penalty on weights. MacKay (2003), Ch. 28 — the Occam factor done cleanly from the Bayesian side. Dinh et al. (2017) — the essential caveat, *Sharp Minima Can Generalize for Deep Nets*; read last, so the caveat lands on a formed picture.
Prerequisite reminder. All of the previous lecture — SGD samples $e^{-L/T}$ (Equation 17.4) at temperature $T \propto \eta/|B|$ (Equation 17.2), with anisotropic noise (Section 17.5); Laplace's method Equation 2.2 and the Gaussian integral Equation 2.1 (Week 1); the free energy $F = \langle E \rangle - TS$ (Week 1) and its career as an objective (Weeks 5 and 7, Equation 14.6); SGLD as the deliberate sampler (Welling and Teh 2011). New content: flat versus sharp minima, the Occam factor as basin entropy, SGD's free-energy preference, and entropy-SGD.

18.1 Which minimum?

The previous lecture ended with training revealed as sampling: SGD draws from $e^{-L/T}$, thermalizing around minima rather than reaching them. A question follows immediately. A modern network's loss landscape holds astronomically many minima, and gradient training will find *one* of them — but the minima are not interchangeable. Two parameter vectors with the *same training loss* can differ substantially on test data; whatever separates them is invisible to the loss value itself. The empirical discriminator has been known, in outline, for a long time: **flat minima generalize, sharp minima do not.** A minimum sitting in a wide, low-curvature basin tolerates perturbation — including the perturbation that matters, the shift from the training loss to the test loss — while a needle-thin minimum of identical depth is brittle. Which basin the optimizer lands in therefore decides generalization, and the previous lecture's thermodynamics decides which basin the optimizer lands in.

The history assembles across three decades. Hochreiter and Schmidhuber (1997) argued that flat minima generalize because they can be described cheaply — a

minimum-description-length argument with Hinton and Camp (1993) as its ancestor, where the free energy of the weights is literally a complexity penalty; both are the Bayesian *Occam's razor* of MacKay (2003), translated to network weights. Twenty years later Keskar et al. (2017) supplied the modern evidence: large-batch training — which the previous lecture taught us to read as *low-temperature* training — systematically finds sharper minima with worse test error. And Chaudhari et al. (2017) closed the loop by building the preference into an algorithm, entropy-SGD, whose objective is not the loss but a local free energy. That free energy is where the physics enters, and it is the same object this course has carried since Week 1.

The question: *why do flat minima generalize, and why does SGD — untold about any of this — find them?* A finite-temperature sampler does not seek the lowest energy but the lowest *free energy*, energy minus temperature times entropy; a basin's entropy is the log of its volume; and a flat minimum is a high-entropy minimum. SGD is that sampler, by the previous lecture's derivation, and its temperature $\eta/|B|$ tunes how strongly volume competes with depth. Module 3 closes with the course's central object applied to generalization.

> Take-home 1
>
> Minima with equal training loss can generalize very differently, and the discriminator is flatness: a wide basin is robust to the train→test shift, a sharp one is brittle. Which minimum SGD finds is a thermodynamic question — a finite-temperature sampler weights basins by free energy, meaning loss *and* volume, not loss alone.

18.2 Sharp or flat, loss or free energy

The robustness argument needs one line and no new machinery. Model the passage from training to test as a small displacement of the loss landscape — the test loss is the training loss evaluated on slightly different data, so its minima sit slightly elsewhere. At a training minimum with Hessian H, a shift of effective size δ raises the loss by roughly $\frac{1}{2}\delta^{\mathsf{T}} H \delta$: the *generalization gap scales with the curvature.* A sharp minimum (H large) converts a small distribution shift into a large loss increase; a flat one barely notices. Figure 18.1 draws the whole argument: two minima of exactly equal training loss, one narrow and one wide, under a shifted test curve — the sharp minimum's loss jumps by $\lambda\delta^2/2 = 0.40$, the flat one's by 0.02, a factor of $\lambda_{\text{sharp}}/\lambda_{\text{flat}} = 25$ with nothing else different between them. (The modern rigorous version of this argument is a *PAC-Bayes* bound, where flatness controls the generalization gap through a KL term; we keep to the physics telling and name the rigorous one.)

The dichotomy is then *loss versus free energy.* Ranking minima by loss ignores geometry entirely; ranking them by free energy — loss minus temperature times basin entropy — is what a sampler does automatically, and it is the ranking that tracks generalization. Basin entropy is computable through Laplace's method (Equation 2.2), as the next section shows.

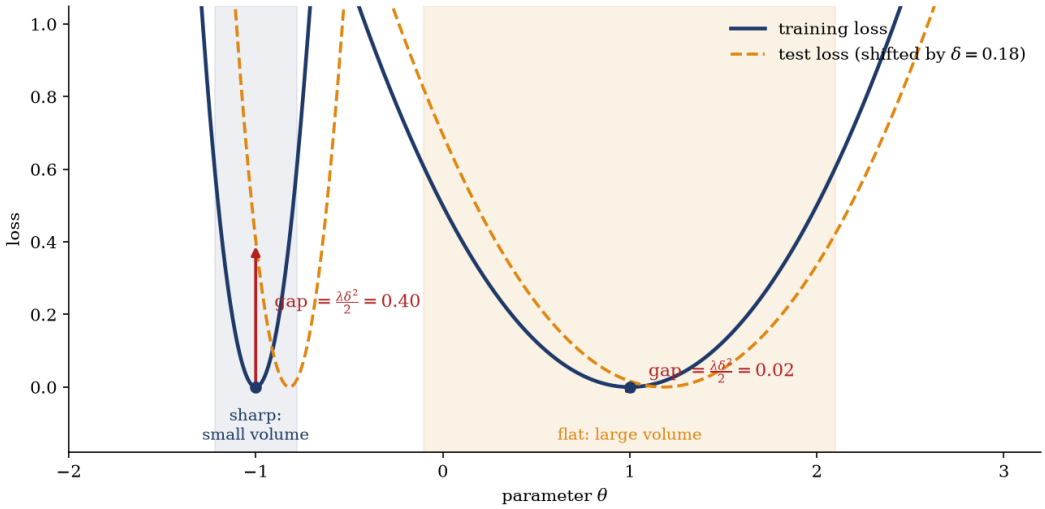

Figure 18.1: The central picture of the lecture: generalization is robustness to the train→test shift. Two minima of the training loss (navy) with exactly equal depth — sharp ($\lambda = 25$, left) and flat ($\lambda = 1$, right) — under the test loss (orange dashed), modeled as the same landscape shifted by $\delta = 0.18$. The gap at each minimum is $\frac{\lambda\delta^2}{2}$: the sharp minimum jumps by 0.40 while the flat one moves by 0.02 — a factor 25, the curvature ratio, at identical training loss. The shaded bands mark each basin's width: the volume that the next section converts into an entropy.

18.3 The engine, part I: flatness is low free energy

The statistical weight of an entire basin, not of its lowest point, is what matters, and Laplace's method (Equation 2.2) provides it. Consider first the Bayesian question — how much evidence does a basin carry? — at unit temperature. The evidence is the integral of e^{-L} over the basin (a flat prior over the basin is assumed here; MacKay's full Occam factor divides this posterior volume by the prior width); expanding L to second order about the minimum θ^* and doing the resulting d-dimensional Gaussian integral,

$$\int_{\text{basin}} \mathrm{d}\theta\, e^{-L(\theta)} \;\approx\; e^{-L(\theta^*)}\, \frac{(2\pi)^{d/2}}{\sqrt{\det H}}, \qquad H = \nabla^2 L(\theta^*), \tag{18.1}$$

Laplace evidence and the Occam factor

each Hessian eigendirection contributing its one-dimensional width $\sqrt{2\pi/\lambda_i}$. The prefactor $(2\pi)^{d/2}/\sqrt{\det H}$ is the **Occam factor** (MacKay 2003): the *volume* of the basin, large for flat minima and small for sharp ones. Two basins of equal depth do not carry equal evidence — the flat one carries more, in proportion to how many parameter settings inside it fit the data well (Figure 18.2, left). The name is apt: the factor penalizes needlessly precise solutions, exactly Occam's preference for the less fine-tuned explanation.

The thermodynamic reading follows by attaching the log-volume to the familiar identity. Define the entropy of a basin as the log of its volume, $S = -\frac{1}{2}\log\det H + \text{const}$; then $F = \langle E\rangle - TS$ becomes, for a basin explored at temperature T,

$$F \;\approx\; L(\theta^*) + \frac{T}{2}\log\det H + \text{const}, \tag{18.2}$$

free energy of a minimum

the loss at the bottom plus a curvature penalty that grows with temperature. (The same result comes from repeating Equation 18.1 with $e^{-L/T}$: the basin's Boltzmann weight is $e^{-L(\theta^*)/T}(2\pi T)^{d/2}/\sqrt{\det H} \equiv e^{-F/T}$, with the T-dependent constant common to all basins and dropped.) The consequence, drawn in Figure 18.2 (right): a *deeper but narrower* minimum can have *higher* free energy than a shallower, wider one. At $T = 0.1$, a sharp basin with $\lambda = 25$ and perfect training loss $L^* = 0$ has $F = 0.161$; a flat basin with $\lambda = 1$ and *worse* training loss $L^* = 0.08$ has $F = 0.080$ — the flat minimum wins on free energy, 0.080 against 0.161, a margin the loss column cannot see. Whether that ranking governs anything depends on the optimizer actually being a finite-temperature sampler. By the previous lecture's derivation, it is.

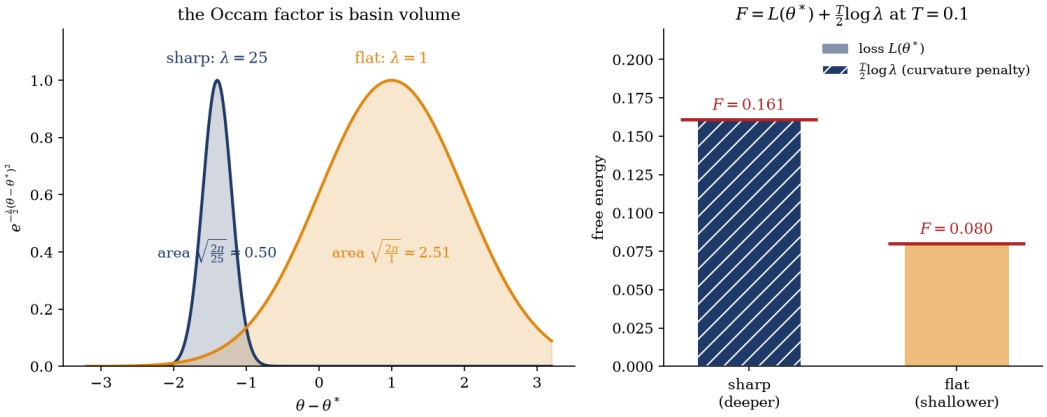

Figure 18.2: Laplace turns basin geometry into statistical weight. Left: the Gaussian factor $e^{-\frac{\lambda}{2}(\theta-\theta^*)^2}$ at a sharp ($\lambda = 25$) and a flat ($\lambda = 1$) minimum; the enclosed area — the one-dimensional Occam factor $\sqrt{2\pi/\lambda}$ — is 0.50 against 2.51: the flat basin holds five times the volume. Right: the free energy Equation 18.2 at $T = 0.1$ for a sharp-but-deeper minimum ($L^* = 0$, hatched curvature penalty $\frac{T}{2}\log 25 = 0.161$) against a flat-but-shallower one ($L^* = 0.08$, penalty zero): the flat minimum has the lower free energy, 0.080 against 0.161, despite the higher loss.

18.4 The engine, part II: why SGD prefers flat minima

The previous lecture proved that SGD run at fixed η and $|B|$ samples the Boltzmann distribution $p_\infty \propto e^{-L/T}$ over parameter space (Equation 17.4). The probability of finding the iterate in a given basin is that distribution integrated over the basin — which is precisely the Laplace computation above:

$$P(\text{basin}) \;\propto\; e^{-F/T} \;\sim\; \frac{e^{-L(\theta^*)/T}}{\sqrt{\det H}}, \tag{18.3}$$

SGD's basin weight

energy and volume together, with the temperature $T \propto \eta/|B|$ arbitrating between them. Hot training — small batches, large learning rate — weights volume heavily:

the sampler rarely visits narrow wells, however deep, and settles in wide basins. Cold training — large batches — chases depth and finds the sharp minima that Keskar's experiments caught generalizing poorly (Keskar et al. 2017). The large-batch generalization gap is not a mystery of optimization; it is Equation 18.3 read at low temperature. And the previous lecture's anisotropy caveat compounds rather than undermines the effect: the gradient noise is *anisotropic*, largest along the steep directions of the landscape, so SGD is kicked out of sharp wells even faster than a uniform thermostat would manage — a Kramers escape-rate argument we state and do not derive, and exactly the covariance structure the tutorial measures.

Entropy-SGD: the preference as an objective. If flat minima are what we want, we need not rely on SGD's implicit thermodynamics — optimize for flatness directly. Chaudhari et al. (2017) replace the loss by the **local free energy**

$$F_\gamma(\theta) \;=\; -\log \int \mathrm{d}\theta' \; e^{-L(\theta') - \frac{\gamma}{2}\|\theta-\theta'\|^2}, \tag{18.4}$$

entropy-SGD objective (local free energy)

a log-partition-function over a Gaussian neighborhood of *scope* $1/\sqrt{\gamma}$ around θ (conventions in the literature differ by harmless prefactors). Small F_γ requires not a low loss *at* θ but a large volume of low loss *near* θ: narrow wells contribute almost nothing to the integral and are effectively erased, while wide basins survive. The gradient has a form that makes the objective trainable:

$$\nabla F_\gamma(\theta) \;=\; \gamma\,(\theta - \langle\theta'\rangle_\rho), \qquad \rho(\theta') \propto e^{-L(\theta') - \frac{\gamma}{2}\|\theta-\theta'\|^2},$$

the displacement of θ from the *mean of the local Gibbs distribution* — estimated, in the algorithm, by an inner SGLD chain (Welling and Teh 2011): Week 8's sampler running as a subroutine inside every outer step, with the scope γ annealed as training proceeds. The free energy entered as a bound (Week 5), became a training loss by amortization (Week 7, Equation 14.6), and now appears as an objective computed *by sampling* — a partition function estimated with the very MCMC machinery of this module.

Figure 18.3 shows the objective in action on a landscape designed to test it: a narrow global minimum (lower loss) against a wide valley (higher loss), plus small-scale ruggedness. The smoothed F_γ erases the wiggles and the spike together, and its global minimum relocates to the wide valley — the optimizer that follows F_γ walks past the sharpest, deepest hole in the landscape, by design.

Take-home 2

Because SGD samples $e^{-L/T}$, it weights basins by $e^{-F/T} \sim e^{-L(\theta^*)/T}/\sqrt{\det H}$ (Equation 18.3): flat minima are favored, with the strength of the preference set by $T \propto \eta/|B|$ — large-batch (cold) training finds sharp minima that generalize worse (Keskar et al. 2017), and the anisotropic noise accelerates the escape from sharp wells. Entropy-SGD makes the preference explicit: its objective F_γ (Equation 18.4) is a local free energy whose gradient is computed by an inner SGLD sampler.

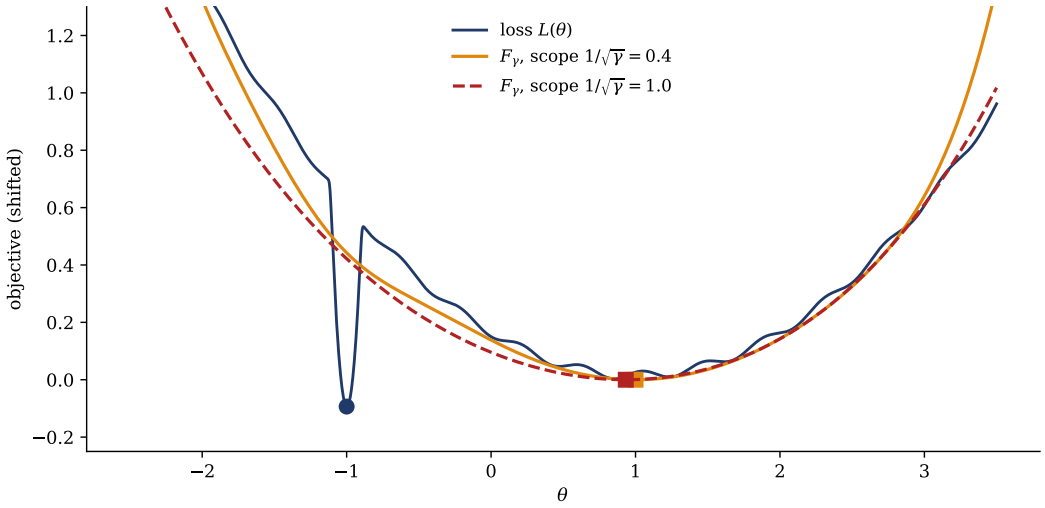

Figure 18.3: The entropy-SGD objective Equation 18.4 on a landscape built to tempt a loss minimizer: the loss $L(\theta)$ (navy) has its global minimum in a needle at $\theta = -1$ (depth -0.09, curvature ~ 120) next to a wide valley at $\theta = +1$ (minimum 0, curvature 0.3), with small wiggles throughout. The local free energy F_γ (each curve shifted so its minimum sits at zero) smooths over a Gaussian neighborhood: at scope $1/\sqrt{\gamma} = 0.4$ (orange) the wiggles and the needle are erased and the global minimum (square) relocates to the wide valley at $\theta \approx 1.0$; a wider scope (red dashed) does the same more aggressively. An optimizer descending F_γ walks past the deepest point of the landscape — deliberately.

18.5 The caveat: flatness is not invariant

The picture above is powerful, and one qualification bounds the claim. Flatness, measured by raw Hessian curvature, is **not a property of the function the network computes** — it depends on the parameterization. Dinh et al. (2017) make this sharp with a one-line construction: in a ReLU network, scaling one layer's weights by c and the next layer's by $1/c$ leaves the input–output function — and therefore the test error — *exactly unchanged*, while rescaling the Hessian's eigenvalues arbitrarily. Any minimum can be made to look as sharp as desired without touching what the network does; hence the paper's title, *sharp minima can generalize*. The free-energy account inherits the same subtlety through its measure: the basin "volume" in Equation 18.1 is a volume in $\mathrm{d}\theta$, and a reparameterization that reshapes $\mathrm{d}\theta$ reshapes the entropy with it.

The repair is to measure curvature in a metric the reparameterization cannot fool — the Fisher-information (natural-gradient) geometry, normalized flatness measures, or the PAC-Bayes route, where the generalization bound is stated invariantly from the start; sharpness-aware minimization (SAM, Foret et al. (2021)), the practical descendant of entropy-SGD, minimizes the worst loss in a perturbation ball — though its sharpness too can be fooled by reparameterization until repaired (the scale-invariant ASAM of Kwon et al. 2021). We name these and leave them to the literature. The standing of the lecture's thesis, then: *flat minima generalize* is a robust empirical regularity with an illuminating physical mechanism, and it is a leading picture rather than a theorem — the same standing this course gave mean field before its correction (Week 5) and contrastive divergence's bias (Week

3). Knowing a tool's domain of validity is part of mastering it.

Trap

Flatness is reparameterization-dependent. Rescaling adjacent ReLU layers by c and $1/c$ changes the Hessian arbitrarily without changing the function (Dinh et al. 2017) — raw curvature, and the basin volume behind Equation 18.2, are coordinate quantities. Sharp minima can generalize. The heuristic becomes reliable only in an invariant formulation (Fisher metric, PAC-Bayes, relative sharpness); quote raw $\det H$ comparisons only between minima of the *same* parameterization, as done throughout this lecture.

Take-home 3

Raw Hessian flatness can be created or destroyed by reparameterization at fixed network function (Dinh et al. 2017), and the free-energy reading is metric-dependent through the volume measure. The fix is an invariant notion of flatness (Fisher / PAC-Bayes / relative sharpness). Flat-minima-generalize is a powerful leading picture, not a theorem.

18.6 Example: two minima, one temperature dial

We now test the lecture's claims on a landscape small enough to integrate exactly. Build a one-dimensional loss with a sharp minimum ($\lambda_s = 25$) and a flat one ($\lambda_f = 1$), and run the two experiments of Figure 18.4.

The left panel tests the robustness claim at *equal depth*: shifting the landscape by $\delta = 0.1$ raises the loss at the sharp minimum by $\lambda_s \delta^2/2 = 0.125$ and at the flat one by 0.005 — the generalization gap is twenty-five times larger at the sharp minimum, the curvature ratio exactly, with the training losses identical. The right panel then makes the competition unfair: the sharp minimum is made *deeper* by $\Delta L = 0.08$, so a zero-temperature loss minimizer would choose it every time. The exact Boltzmann occupation of the flat basin, $P_{\text{flat}}(T)$, computed by quadrature, tells a different story: cold, the sampler sits in the sharp well ($P_{\text{flat}} = 0.08$ at $T = 0.02$); hot, it lives in the flat one (0.77 at $T = 0.2$); and the crossover sits where Equation 18.2 says the two free energies cross,

$$T^* = \frac{2\,\Delta L}{\log(\lambda_s/\lambda_f)} = \frac{0.16}{\log 25} \approx 0.050,$$

with the Laplace prediction $P_{\text{flat}} = [1 + e^{\Delta F/T}]^{-1}$ lying on top of the exact curve across the whole range. Translated through $T \propto \eta/|B|$, this is Keskar's experiment in one dimension: batch size and learning rate move the training run along this curve, and which minimum "SGD converged to" is a point on it.

18.7 Outlook: Module 3 closes

The tutorial (16–18, Ph12 106) belongs to the previous lecture's crux — measuring the mini-batch noise covariance against the Fokker–Planck prediction — and today's

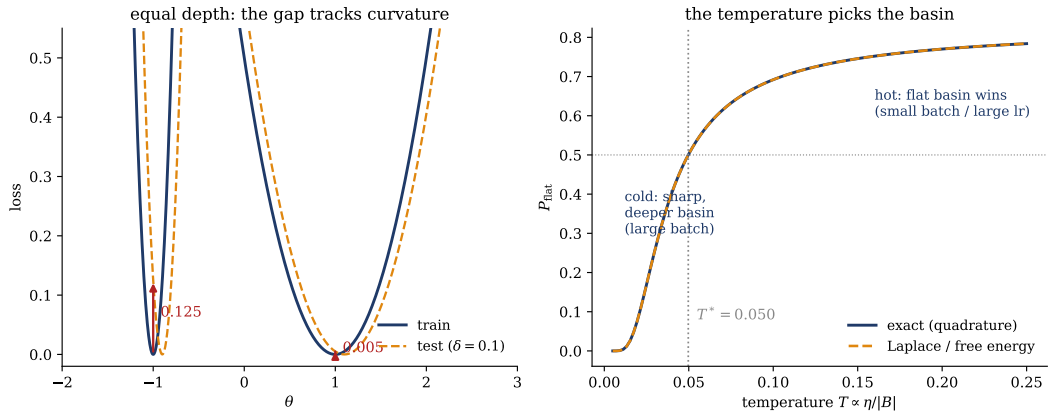

Figure 18.4: The worked example: a double-well loss with a sharp ($\lambda_s = 25$) and a flat ($\lambda_f = 1$) minimum. Left: at equal depth, the train→test shift $\delta = 0.1$ opens a gap $\lambda\delta^2/2$ at each minimum — 0.125 (sharp) against 0.005 (flat), a factor 25 at identical training loss. Right: with the sharp minimum now deeper by $\Delta L = 0.08$, the equilibrium probability that a sampler of $e^{-L/T}$ occupies the flat basin, computed exactly by quadrature (navy) and by the Laplace/free-energy formula $P = [1 + e^{\Delta F/T}]^{-1}$ with $\Delta F = \Delta L - \frac{T}{2}\log(\lambda_s/\lambda_f)$ (orange dashed, indistinguishable): cold ($T = 0.02$) the sampler sits in the sharp-but-deep well ($P_{\text{flat}} = 0.08$); hot ($T = 0.2$) in the flat-but-shallow one (0.77); the crossover $T^* = 2\Delta L/\log(\lambda_s/\lambda_f) \approx 0.050$ (gray dotted) is where the two free energies cross. Through $T \propto \eta/|B|$, batch size and learning rate select a point on this curve.

lecture has told you what that measurement is *for*: the anisotropy you will find in $C(\theta)$ is the mechanism that hurries SGD out of sharp wells and into the flat basins that generalize. You will measure it yourselves — a computation-led week, one machine to a group, opened after the room has predicted what the covariance will look like. The two lectures of the week are one argument: training is Langevin dynamics, and its temperature buys generalization.

Module 3's arc: Week 8 established that the intractable Z need not be computed to be used — Metropolis made it cancel (Section 15.3), and annealing turned the sampler into an optimizer by cooling (Chapter 16). Week 9 revealed that the optimizer everyone already runs is such a sampler, with temperature $\eta/|B|$ (Equation 17.2), equilibrating toward $e^{-L/T}$ (Equation 17.4). And today that temperature acquired its purpose: it weights basins by free energy (Equation 18.3), preferring the flat, high-entropy minima that survive the train→test shift — with entropy-SGD (Equation 18.4) turning the preference into an explicit objective. Training is sampling; the learning rate is a temperature; generalization is a free energy.

Module 4 takes this machinery to its destination. Week 10 removes the partition function outright — score matching works with $\nabla_x \log p$, where Z differentiates to zero (Hyvärinen 2005) — moving the course's dynamics from parameter space to configuration space at last. Week 11 turns nonequilibrium physics into an estimator: the Jarzynski and Crooks fluctuation theorems (Jarzynski 1997; Crooks 1999) extract free-energy differences from driven, out-of-equilibrium trajectories. And Week 12 runs the previous lecture's Langevin/Fokker–Planck equations backwards in time to *generate*: the diffusion model, the course title's destination. The free energy that

has organized every module now builds the samplers of the frontier.

> *SGD does not chase the lowest loss — it chases the lowest free energy, and a flat minimum is a low-free-energy minimum.*

Part IV

Module 4 — Nonequilibrium thermodynamics of generative modeling

19 Score matching: differentiating past the partition function

Recommended reading

Core (~1 h). Hyvärinen (2005) — the founding score-matching paper, and the resolution of this course's central obstruction in fifteen pages: the objective, the integration-by-parts theorem, and the consistency proof. This is the warm-up reading (§1–2); today's engine sections follow §2–§3 closely. Alongside it, the energy-based-model sections of Mehta et al. (2019), for the physicist's framing of unnormalized models.
Optional background. Vincent (2011) — denoising score matching: the practical variant, and the essential bridge to Week 12. Song and Ermon (2019) — the modern revival (NCSN): scores at many noise scales, generation by annealed Langevin; the direct ancestor of diffusion models. Song et al. (2020) — sliced score matching, the scalable estimator for the Jacobian-trace term. And your own Week 3 notes (Section 5.5) — the obstruction this lecture removes; worth re-reading for the contrast.
Prerequisite reminder. The energy-based model $p = e^{-E}/Z$ and the Boltzmann distribution Equation 1.2 (Week 1); the negative phase $\nabla_\theta \log Z$ and contrastive divergence (Section 5.5, Section 6.4 — Week 3); Langevin dynamics and its stationary law (Section 17.4 — Week 9, here transplanted to configuration space); integration by parts. New content: the score $\nabla_x \log p$, the $\nabla_x \log Z = 0$ removal, and Hyvärinen's objective.

19.1 The partition function, revisited

Every module of this course has been shaped by the same object. Week 1 introduced the partition function $Z = \sum_x e^{-E(x)}$ as the quantity that turns an energy into a probability, and flagged it as the recurring obstruction; Week 3 met the obstruction in earnest — the likelihood gradient of a Boltzmann machine needs $\nabla_\theta \log Z$, the negative phase (Equation 5.5), an average over the model that no tractable computation supplies, and contrastive divergence survived only by dodging it and paying in bias (Section 6.4). Module 2 stopped fighting for $\log Z$ exactly and *bounded* it, building the variational free energy (Equation 9.1) and everything through the VAE on that bound. Module 3 arranged for Z to *cancel*: the Metropolis rule needs only ratios (Section 15.3), and training itself turned out to be such a sampler. Each module found a different workaround — bounding, cancelling, dodging — but none eliminated Z from the equations.

The present lecture removes it. The partition function is an integral over *all* configurations: a number. It depends on the parameters θ — hence Week 3's obstruction

— but it does not depend on which configuration x you happen to be looking at. To the *configuration* variable, Z is a constant, and constants have zero gradient:

$$\nabla_x \log Z \;=\; 0.$$

So the moment we differentiate $\log p$ with respect to x rather than θ, the object that blocked energy-based learning disappears from the equations — eliminated — not approximately, but identically — by the derivative. Hyvärinen (2005) built an estimation method on exactly this observation, *score matching*, which fits an unnormalized model to data with no Z anywhere; Vincent (2011) made it cheap (denoising); Song and Ermon (2019) turned it into state-of-the-art generation and set it on the road to diffusion. The idea is, in retrospect, straightforward: differentiate past the normalizer.

The question: *how do you fit $p = e^{-E}/Z$ to data when you cannot compute Z?* First, fit the *score* $\nabla_x \log p$ instead of the density — then Z drops out. Second, the remaining obstacle — that we do not know the score of the *data* either — falls to an integration by parts. What comes out is an exact, sampling-free objective for unnormalized models, and the object it estimates, the score, is precisely the drift a Langevin equation needs to generate.

> Take-home 1
>
> Z is a constant in x, so $\nabla_x \log Z = 0$: the score of an energy-based model is $\nabla_x \log p = -\nabla_x E$, with no partition function anywhere. The obstruction that Week 1 named and Week 3 collided with is invisible to the configuration gradient — this is the resolution announced in the first lecture.

19.2 Two gradients of one constant

We state the escape as a dichotomy before using it, because the entire method is the choice between two derivatives of the same object. The *parameter* gradient $\nabla_\theta \log Z = -\langle \nabla_\theta E \rangle_{\text{model}}$ is Week 3's obstruction: not zero, not tractable, the negative phase that forced contrastive divergence. The *configuration* gradient $\nabla_x \log Z$ is zero, identically, trivially. Same Z; one derivative is the obstruction and the other does not exist. This is why every gradient in the course carries a subscript — the choice between ∇_θ and ∇_x is what makes the removal visible. (The distinction appeared in Section 17.2 from the parameter-space side; the configuration-space counterpart is the present section.)

What does one gain by fitting $\nabla_x \log p$ instead of p? Figure 19.1 shows the two descriptions of one and the same distribution. The density is a scalar field whose values must integrate to one — normalization is constitutive, and Z is its price. The **score** $s(x) = \nabla_x \log p(x)$ is a *vector field*: at every point, an arrow pointing in the direction of increasing log-probability — toward the data, everywhere. Arrows carry no normalization; scaling p by any constant leaves every arrow untouched. And the arrows are not a lossy summary: up to the overall constant, the score determines the distribution (integrate it back up), and — the module's coming theme — a field of arrows toward the data is already a generator, because it is exactly the drift that Week 9's Langevin equation needs to sample p.

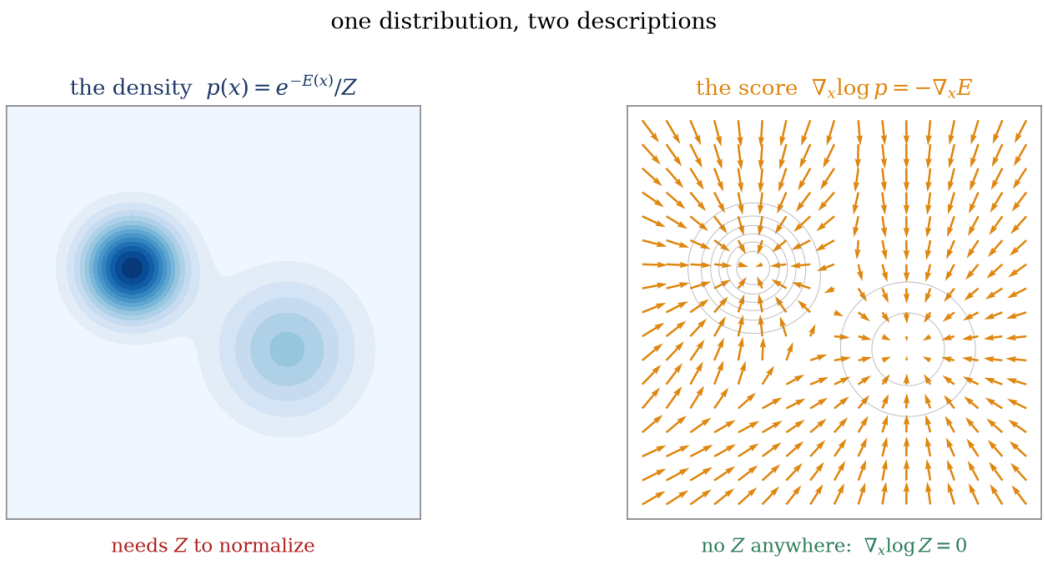

Figure 19.1: The central picture of the lecture: one distribution, two descriptions. Left: the density $p(x) = e^{-E(x)}/Z$ of a two-mode distribution — a scalar field that must integrate to one, which is what the partition function is for. Right: the score $\nabla_x \log p = -\nabla_x E$ of the same distribution — a vector field of arrows pointing toward the data (gray contours: the density), invariant under any rescaling of p, hence free of Z. Score matching learns the right panel from samples; Langevin dynamics turns the right panel back into the left one.

A notational change: after serving as an Ising spin for six weeks, s **now denotes the score** — a gradient field over configuration space, ML's standard symbol for it — and σ below is a noise scale, not a spin. The energy $E(x)$ and its configuration gradient $\nabla_x E$ are the objects of Weeks 1–3, unchanged.

19.3 The engine, part I: the score is Z-free

The first result is central, and the derivation is short. For an energy-based model $p_\theta(x) = e^{-E_\theta(x)}/Z_\theta$,

$$s_\theta(x) \;=\; \nabla_x \log p_\theta(x) \;=\; \nabla_x\big(-E_\theta(x) - \log Z_\theta\big) \;=\; -\nabla_x E_\theta(x), \qquad \text{since } \nabla_x \log Z_\theta = 0. \tag{19.1}$$

the score is Z-free

The score of an unnormalized model is the negative gradient of its energy — computable, for any energy you can write down, by a single differentiation, with no knowledge of Z whatsoever. Contrast this with what Week 3 needed. Maximum likelihood differentiates $\log p_\theta$ in θ, and there $\log Z_\theta$ does *not* drop out: $\nabla_\theta \log Z_\theta = -\langle \nabla_\theta E_\theta \rangle_{\text{model}}$, the negative phase, an expectation under the very distribution we cannot sample cheaply (Equation 5.5). The two statements sit side by side:

$$\underbrace{\nabla_\theta \log Z_\theta = -\big\langle \nabla_\theta E_\theta \big\rangle_{\text{model}} \;\neq\; 0}_{\text{Week 3: the obstruction}} \qquad \underbrace{\nabla_x \log Z_\theta = 0}_{\text{today: the escape}}.$$

To be precise about what has and has not been achieved: Z_θ is not made zero — it still exists, still depends on θ, and still cannot be computed. A *different derivative of it* is zero, and score matching is the estimation method that only ever needs that derivative. The remaining question is what to match the model score *against*. The natural target is the score of the data — which we do not know either, since we have samples from p_{data}, not its formula. Removing that second unknown is Hyvärinen's theorem.

19.4 The engine, part II: Hyvärinen's theorem

The natural objective is to make the model's score field match the data's, in mean square under the data distribution — the *Fisher divergence*:

$$J(\theta) \;=\; \frac{1}{2}\left\langle \left\| s_\theta(x) - \nabla_x \log p_{\text{data}}(x) \right\|^2 \right\rangle_{p_{\text{data}}},$$

which vanishes exactly when s_θ equals the data score (almost everywhere), so its minimizer recovers the true model when the family contains it (Hyvärinen 2005). As written it is useless: it contains $\nabla_x \log p_{\text{data}}$, the score of an unknown density. Expand the square and note where the unknown sits,

$$J(\theta) \;=\; \left\langle \tfrac{1}{2}\|s_\theta\|^2 \right\rangle_{p_{\text{data}}} - \left\langle s_\theta \cdot \nabla_x \log p_{\text{data}} \right\rangle_{p_{\text{data}}} + \underbrace{\left\langle \tfrac{1}{2}\|\nabla_x \log p_{\text{data}}\|^2 \right\rangle_{p_{\text{data}}}}_{\text{independent of } \theta} :$$

the first term is pure model, the last is a constant we may discard, and only the cross term couples to the unknown. That term is the target of the next manipulation. Write the expectation as an integral and use $p_{\text{data}} \nabla_x \log p_{\text{data}} = \nabla_x p_{\text{data}}$:

$$-\left\langle s_\theta \cdot \nabla_x \log p_{\text{data}} \right\rangle_{p_{\text{data}}} \;=\; -\int \mathrm{d}x\; s_\theta(x) \cdot \nabla_x p_{\text{data}}(x) \;=\; +\int \mathrm{d}x\; p_{\text{data}}(x)\, \nabla_x \cdot s_\theta(x),$$

integrating by parts in each coordinate, with the boundary terms vanishing provided $p_{\text{data}}(x)\, s_\theta(x) \to 0$ at infinity (true for any decaying density and reasonable model; the regularity assumptions are stated in full in Hyvärinen (2005), and they can genuinely fail on bounded domains or heavy tails — this is the theorem's fine print). The unknown data score is gone: differentiation has been thrown from the data onto the model, where we can afford it. Reassembling,

$$\boxed{J(\theta) \;=\; \left\langle \tfrac{1}{2}\,\|s_\theta(x)\|^2 \;+\; \operatorname{tr} \nabla_x s_\theta(x) \right\rangle_{p_{\text{data}}} \;+\; \text{const},} \tag{19.2}$$

Hyvärinen's score-matching objective

an expectation over the *data* of two quantities computable at any sample point: the squared norm of the model score, and its divergence $\operatorname{tr} \nabla_x s_\theta = \sum_i \partial^2 \log p_\theta / \partial x_i^2$. For an energy-based model, using Equation 19.1, the objective reads

$$J(\theta) \;=\; \left\langle \tfrac{1}{2}\,\|\nabla_x E_\theta\|^2 \;-\; \nabla_x^2 E_\theta \right\rangle_{p_{\text{data}}} + \text{const} : \tag{19.3}$$

score matching for an EBM

the mean squared force minus the mean Laplacian of the energy, evaluated on data points. Note what has happened to the intractability of energy-based learning: the negative phase — an average over the *model*, requiring sampling — has been replaced entirely by averages over the *data*, requiring nothing but derivatives of the energy at points you already have. There is no Z, no Monte Carlo, and no bias. The two removals compose: $\nabla_x \log Z = 0$ deleted the normalizer, and integration by parts deleted the data score. What maximum likelihood could not do exactly for a single unnormalized model, score matching does exactly for all of them.

The price is a change of divergence: score matching minimizes the Fisher divergence, not the KL divergence of maximum likelihood. The estimator is *consistent* — with a realizable model and enough data it recovers the same truth — but at finite samples the two weight errors differently (Fisher divergence is comparatively insensitive to getting the mixture *weights* of well-separated modes right, precisely because so little data sits between the modes). It is a different estimator, not a reparameterized likelihood.

> Take-home 2
>
> Integration by parts moves the derivative off the unknown data score and onto the model: the Fisher divergence becomes Hyvärinen's objective $J = \langle \frac{1}{2}\|s_\theta\|^2 + \operatorname{tr} \nabla_x s_\theta \rangle_{p_{\text{data}}} + \text{const}$ (Equation 19.2), computable from samples alone under mild boundary assumptions. Together with $\nabla_x \log Z = 0$, an unnormalized model is fitted exactly with no partition function and no sampling — at the price of minimizing the Fisher divergence rather than the KL.

19.5 What score matching delivers, and the road to diffusion

The deliverable is a score, not a density. Score matching estimates $-\nabla_x E$, equivalently the energy up to an additive constant — and that constant is exactly $\log Z$, which the method neither recovers nor needs. You cannot evaluate $p_\theta(x)$; you cannot compare likelihoods. What you *can* do is generate: the score is the drift of the configuration-space Langevin equation

$$\mathrm{d}x \;=\; s_\theta(x)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W,$$

whose stationary distribution is p_θ — Week 9's machinery (Equation 17.3) transplanted from parameters to data, with the learned arrows of Figure 19.1 doing the steering. A fitted score field is a working sampler of an unnormalizable model. This structural fact is the module's foundation: generation needs the score, not the density, and the score carries no normalizer.

The cost, and the denoising fix. The divergence term in Equation 19.2 is the objection in high dimension: $\operatorname{tr} \nabla_x s_\theta$ is a Jacobian trace, costing on the order of d backpropagations per data point. Vincent (2011) removed it with a move that looks like a compromise and proved decisive. Perturb each data point with Gaussian noise, $\tilde{x} = x + \sigma\varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$, and match the score of the *noised* distribution q_σ. The conditional score is closed-form, $\nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x) = -(\tilde{x} - x)/\sigma^2$, and Vincent's

identity states that regressing on it is equivalent (up to a θ-independent constant) to score-matching the noised marginal:

$$J_{\mathrm{DSM}}(\theta) \;=\; \frac{1}{2}\left\langle \left\| s_\theta(\tilde{x}) + \frac{\tilde{x}-x}{\sigma^2} \right\|^2 \right\rangle_{x\sim p_{\mathrm{data}},\, \tilde{x}\sim q_\sigma(\cdot|x)}, \tag{19.4}$$

denoising score matching

with minimizer $s_\theta = \nabla \log q_\sigma$, the score of the data smoothed at scale σ. (The identity is two lines: expanding the square, the cross term $\langle s_\theta \cdot \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x)\rangle$ collapses by $\int \mathrm{d}x\, p_{\mathrm{data}}(x)\, q_\sigma(\tilde{x} \mid x) \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x) = \nabla_{\tilde{x}} q_\sigma(\tilde{x})$ into the same cross term with the marginal's score — no integration by parts, no Jacobian.) Read the objective as a task: the model is handed a noised point and trained to predict the direction back to the clean one — **denoising**. Matching scores and removing noise are the same problem (Figure 19.2, left). Sliced score matching (Song et al. 2020), which estimates the Jacobian trace with random projections, is the other scalable route; denoising is the one history chose.

The bridge to Week 12. A score learned at one small noise scale inherits a weakness of Langevin sampling itself: in the low-density regions between modes there is almost no data, the learned arrows are unreliable there (the example below catches this happening), and a sampler must cross exactly those regions to mix. The repair of Song and Ermon (2019) is to learn the score at *many* noise scales — heavily smoothed distributions have no empty regions — and generate by *annealed Langevin*: start from pure noise where the coarse score is trustworthy, and step down the noise ladder as the samples approach the data (Figure 19.2, right). Forward noising, a learned family of denoising scores, a reverse pass from noise to data: this is a **diffusion model** in everything but name, and the per-scale training loss is Equation 19.4. The next lecture supplies the missing mathematics — Anderson's reverse-time SDE, which makes "run the noising backwards" an exact equation with the score as its only unknown — and Week 12 assembles the full machine. The object everything runs on is the Z-free score built today.

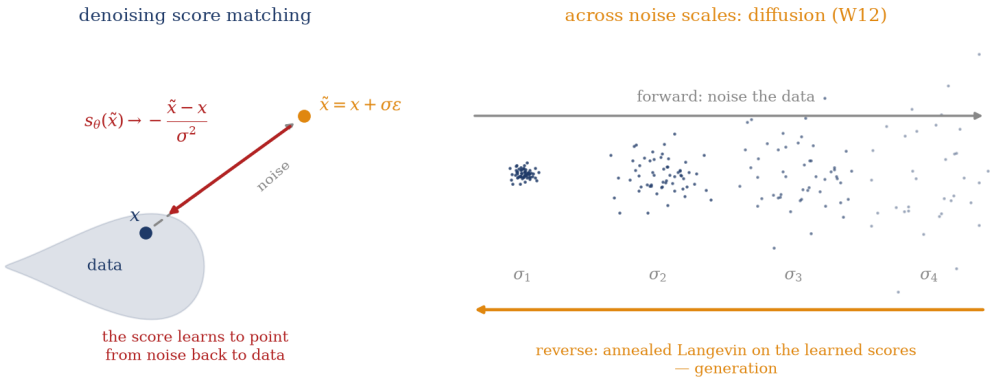

Figure 19.2: Denoising score matching, and where it leads. Left: a data point x is perturbed to $\tilde{x} = x + \sigma\varepsilon$ (gray dashed), and the score model is trained to point back, $s_\theta(\tilde{x}) \to -(\tilde{x}-x)/\sigma^2$ (red): matching the score of noised data is denoising — no Jacobian trace, no partition function. Right: the same idea across a ladder of noise scales $\sigma_1 < \cdots < \sigma_4$: the forward direction noises data into a featureless cloud; generation runs the ladder in reverse by annealed Langevin on the learned scores (Song and Ermon 2019). This is a diffusion model in embryo, trained per scale by Equation 19.4 — Week 12's subject.

Trap

Three standard errors. ① The escape is $\nabla_x \log Z = 0$ — the *configuration* gradient. Week 3's obstruction $\nabla_\theta \log Z$, the parameter gradient, is not zero and never became zero; a different derivative of the same object vanishes. Conflating the two subscripts erases the entire logic of the course's spine. ② Score matching delivers an *unnormalized* model: the energy up to an additive constant ($= \log Z$), a sampler (via Langevin), but never $p_\theta(x)$ or a likelihood. ③ Hyvärinen's integration by parts has fine print — boundary terms must vanish and the Fisher divergence is not the KL — so the exactness claim is "exact under stated assumptions, for a different divergence," not "maximum likelihood for free."

Take-home 3

Score matching yields the score (the energy up to the constant $\log Z$): you can sample it by Langevin — the score is the drift, Week 9's machinery in configuration space — but not evaluate the density. The Jacobian-trace cost is removed by denoising score matching (Equation 19.4): match the score of noise-perturbed data, which is a denoising task, and exactly a diffusion model's per-scale training loss. Learned across noise scales with annealed Langevin (Song and Ermon 2019), the Z-free score becomes a generator — Week 12 in embryo.

19.6 Example: a model fitted with no Z

We now make the removal operational on the smallest nontrivial case, with an additional simplification: for a score model *linear in its parameters*, $s_\theta(x) = \sum_k w_k \varphi_k(x)$ with fixed basis functions φ_k, Hyvärinen's objective Equation 19.2 is *quadratic* in w, so exact score matching reduces to solving a linear system — no gradient descent and no neural network, the estimator in its purest form. Take 5000 samples from a two-mode Gaussian mixture, a basis of 25 Gaussian bumps, and fit three ways of looking at the same estimate (Figure 19.3).

The left panel is the estimator at work: the fitted score (navy) tracks the true mixture score (gray) closely wherever data lives, and denoising score matching at $\sigma = 0.2$ (orange dashed) recovers essentially the same field — from noised samples, with no divergence term. Both fits wobble beyond $|x| \gtrsim 2.5$, and the wobble is expected: score matching weights its objective by p_{data}, so the score is simply unconstrained where the data does not go — the low-density blind spot that multi-scale noise (Week 12) exists to fix. The middle panel illustrates the central point: integrating the fitted score gives the energy, and the recovered curve runs parallel to the true $-\log p$ — the vertical gap between them is constant to within 0.11 across the data region. An additive constant is exactly what a gradient cannot see, and for $p = e^{-E}/Z$ the normalization $\log Z$ lives in that constant: the one object score matching cannot deliver is the one it was built to avoid. The right panel closes the loop: Langevin sampling with the learned score as drift regenerates the data density, placing 49.1% of its samples in the left mode against the true 50% — generation from a model whose normalizer was never computed.

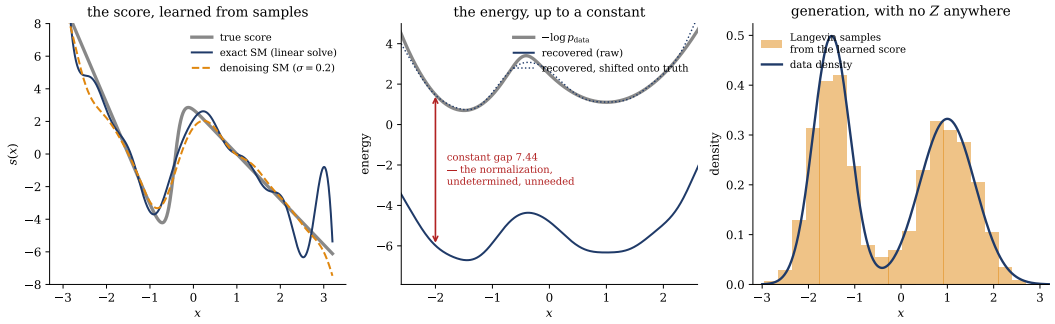

Figure 19.3: Score matching with no partition function, on 5000 samples from the mixture $\frac{1}{2}\mathcal{N}(-1.5, 0.4^2) + \frac{1}{2}\mathcal{N}(1.0, 0.6^2)$, using a score model linear in 25 Gaussian basis functions (Hyvärinen's objective is then quadratic — solved exactly by linear algebra). Left: the fitted score (navy) against the true mixture score (gray); denoising score matching at $\sigma = 0.2$ (orange dashed) recovers the same field from noised samples with no divergence term. Both wobble outside the data region ($|x| \gtrsim 2.5$) — score matching is data-weighted, and the score is unconstrained where the density vanishes: the blind spot that motivates multiple noise scales (W12). Middle: integrating the fitted score recovers the energy up to an additive constant — the raw recovered curve (navy) runs parallel to the true $-\log p$ (gray), their gap constant to within 0.11 across the data region; that gap (the integration constant, where $\log Z$ hides) is exactly the information score matching discards. Right: Langevin sampling with the learned score as drift regenerates the data density, with 49.1% of samples in the left mode against the true 50% — generation from a never-normalized model.

19.7 Outlook: the spine, resolved

The next lecture takes the reverse-time SDE up in full generality (Anderson's 1982 result): a learned score is not merely a Langevin drift but the exact ingredient that inverts *any* diffusion — the mathematical heart of Week 12's generative models. The tutorial (Ph12 106) is derivation-led: you will fit a two-dimensional score model with Hyvärinen's objective, and derive by hand — for the one-dimensional Ornstein–Uhlenbeck process you now know well — the *reverse-time SDE* that turns a forward noising process into a generator. The tutorial connects Equation 19.2 to Week 3's RBM fitting: the object that blocked training there has a specific relationship to today's score, and the session's concluding exercise makes the connection explicit.

The wider map of the module: Week 11 turns nonequilibrium thermodynamics itself into an estimator — the Jarzynski and Crooks fluctuation theorems (Jarzynski 1997; Crooks 1999) extract equilibrium free-energy differences from driven trajectories, annealed importance sampling being their ML incarnation — and Week 12 assembles diffusion models (DDPM, score-SDE (Ho, Jain, and Abbeel 2020; Song, Sohl-Dickstein, et al. 2021)) from the parts now on the table: a forward noising process (Week 9's Fokker–Planck), a score learned by denoising (Equation 19.4), and a reverse-time equation (the next lecture). Every ingredient of a diffusion model is now in place.

Figure 19.4 summarizes the course's four encounters with Z. The partition function was met four ways across four modules: *dodged* by contrastive divergence, which traded exactness for tractability; *bounded* by the variational free energy, which traded tightness for a guarantee; *cancelled* by Metropolis ratios, which traded closed forms for samples; and now *removed* by the configuration gradient, which traded the density for the score — and found that generation does not need the density in the first place.

> *The partition function is a constant to the score — and constants have zero gradient.*

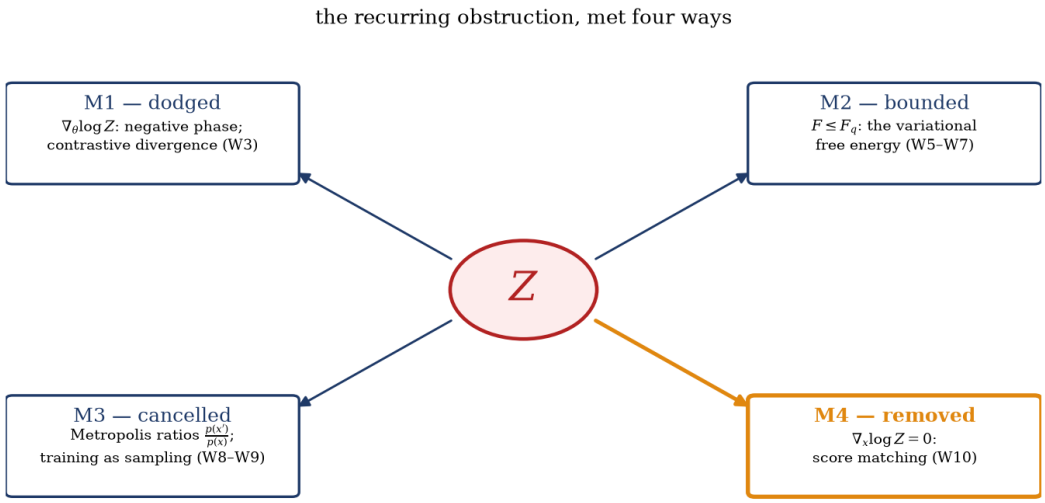

Figure 19.4: The course's spine, closed. The partition function Z — the recurring obstruction named in Week 1 — was met four ways across four modules: dodged (Module 1: the negative phase $\nabla_\theta \log Z$ forced contrastive divergence), bounded (Module 2: the variational free energy $F \le F_q$), cancelled (Module 3: Metropolis ratios, training as sampling), and finally removed (Module 4: $\nabla_x \log Z = 0$, score matching). Each escape paid a different price; the last one pays with the normalization itself, which generative modeling turns out not to need.

20 The reverse-time SDE: generation is time reversal

Recommended reading

Core (~1 h). Anderson (1982) — the crux theorem, proved in control theory four decades before generative modeling needed it: a diffusion run backwards in time is again a diffusion, with the score of the marginals correcting the drift. Read for the statement and the object it demands; the Fokker–Planck derivation below is our route to it. Alongside it, Song and Ermon (2019) — scores learned at many noise levels, generation by annealed Langevin: the modern rediscovery that a score field generates, and the stepping stone between the previous lecture and today's theorem.
Optional background. Song, Sohl-Dickstein, et al. (2021) — the full continuous framework (forward SDE, reverse SDE, probability-flow ODE); this is *Week 12's* backbone, so skim the framing sections only and stop before the ODE and likelihood machinery; Vincent (2011) — the previous lecture's denoising score matching, re-read with today's eyes as "the score of p_t."
Prerequisite reminder. All of the previous lecture (Chapter 19) — the score $s(x) = \nabla_x \log p(x)$, the removal $\nabla_x \log Z = 0$ (Equation 19.1), Hyvärinen's objective (Equation 19.2), and denoising score matching (Equation 19.4); the Langevin/Fokker–Planck machinery of Chapter 17 (especially Equation 17.3, today's engine, now run in configuration space); and the mixing analysis of Section 15.4, whose central obstacle today's construction circumvents. New content: forward noising processes and their marginals, Anderson's reverse-time SDE with two derivations (the flipped current, and a discrete-time proof by Bayes' rule), transport versus equilibration, and the time-dependent score as continuous denoising score matching.

20.1 Almost a generator

The previous lecture left one step between the learned score and a full generative model; this lecture supplies it. The score $\nabla_x \log p = -\nabla_x E$ is free of the partition function, and Hyvärinen's objective — in its denoising form especially — fits it from data alone. And Week 9 supplied the reason a score field should be worth fitting: the score is a *drift*. Langevin dynamics $\mathrm{d}x = \nabla_x \log p(x)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W$ has p as its stationary law (the configuration-space twin of Section 17.4's parameter-space result), so a learned score plus a Langevin integrator is, in principle, a machine that turns noise into samples — *almost* a generator. (The warm-up card asked you to reverse an Ornstein–Uhlenbeck process, and it separates two questions: what the *marginals* of the reversed process must be, which Week 9 answers, and what *dynamics* achieves them, which is another matter entirely. That missing dynamics

is today's boxed theorem, and in the tutorial you will build the special case by hand.)

The qualification matters. Langevin sampling *equilibrates*: started anywhere, it must run until it forgets its initialization — and that is Module 3's mixing problem again, in a new setting. On a multimodal target, the walker crosses between modes at exponentially rare intervals (Section 15.4; the trapped walker of Section 15.2), and *knowing the score does not lower a single barrier* — the drift field is precisely what builds the barrier. Worse, a second failure is specific to the learned setting: score matching trains s_θ where the data live; in the low-density region *between* modes — exactly the terrain a mixing walker must cross — the learned score is unconstrained by any training signal, and the walker is steered by noise of our own manufacture. The worked example below makes the first failure quantitative in its purest form: with the *exact* score in hand and a generous step budget, Langevin on a two-mode target simply keeps whatever mode weights its initialization happened to have. Equilibrium sampling did not get easier because the previous lecture removed Z.

The fix is the founding idea of this module: **do not equilibrate — transport.** Take the one distribution in the problem that requires no mixing whatsoever — pure Gaussian noise — and *manufacture a path* from the data to it: a fixed, parameter-free forward diffusion that corrupts data to noise. Then reverse time. That the reversal exists — that a diffusion run backwards is again a diffusion, with a drift correction given exactly by the score of the time-marginals — is a theorem, proved by Anderson (1982) in the control-theory literature, forty years before generative modeling needed it (a recurring pattern in this course: Onsager 1936 waited for TAP, Bethe 1935 for belief propagation, Anderson 1982 for diffusion models). Generation becomes the corruption run in reverse, steered by the previous lecture's Z-free score (Figure 20.1) — and the mixing problem is not solved but *never encountered*, because the reverse path starts at the only distribution that is already mixed and travels only through blurred intermediates whose barriers have melted.

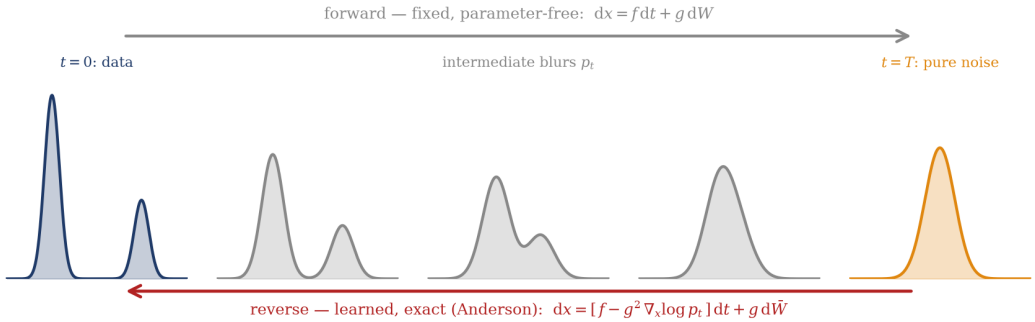

Figure 20.1: The central picture of the lecture — a diffusion model as a film run twice. Top row (forward, left to right): a fixed noising SDE corrupts samples of the data density (navy, two unequal modes) through blurred intermediates p_t into a unit Gaussian — parameter-free, trivial to simulate, and the endpoint requires no mixing whatsoever. Bottom row (reverse, right to left): Anderson's reverse-time SDE, steered by the score of each intermediate blur, carries fresh noise back through the *same* marginals to the data. Every panel of the bottom row is a distribution that the previous lecture's denoising score matching was trained on; the barrier of the leftmost panel is never crossed, because the reverse path visits it only in its melted forms.

Take-home 1

A learned score plus Langevin is *almost* a generator — defeated by mixing (barriers between modes, unpriced by knowing the score) and coverage (the score is untrained exactly where a mixing walker must travel). The fix is nonequilibrium: corrupt data to noise with a fixed forward diffusion — noise needs no mixing — and reverse time, which Anderson's theorem makes exact.

20.2 The forward process: corruption is easy

Everything today runs on one humble SDE. Take the Ornstein–Uhlenbeck (variance-preserving) forward process

$$\mathrm{d}x \;=\; -x\,\mathrm{d}t \;+\; \sqrt{2}\,\mathrm{d}W \tag{20.1}$$

the forward (noising) process

— Week 9's workhorse, now applied to *data*: start $x_0 \sim p_0$ (the data distribution) and let every sample decay toward the origin while being kicked by noise. Its transition law is one line of Week 9 algebra (the script keeps it): $x_t \mid x_0 \sim \mathcal{N}(x_0\, e^{-t},\; 1 - e^{-2t})$. Three facts about the resulting *time-marginals* p_t follow:

The endpoint is free. As t grows, the memory of x_0 decays as e^{-t} and the variance saturates: $p_T \to \mathcal{N}(0,1)$ for T of a few units, *whatever the data were*. The far end of the path is a unit Gaussian — exactly sampleable, no chain, no mixing, no Z. Corruption is the one direction in which everything is easy.

The intermediates are blurs, and the barrier vanishes. For each t, p_t is the data smeared with a known Gaussian (a mixture of Gaussians stays a mixture: means shrink as e^{-t}, widths grow toward one). Figure 20.2 computes the whole family exactly for a two-mode target: the sharp bimodal density at $t = 0$ merges, hump by hump, into the unit Gaussian. Compare this picture with Langevin's failure — the deep valley between the modes, the barrier that equilibration must cross, simply *fills in* along the path. A process that travels through the p_t in sequence never faces the $t = 0$ barrier at all.

The intermediates are where the score is trained. The previous lecture's denoising score matching (Equation 19.4) fits s_θ on *noise-perturbed* data — which is to say, on samples of precisely these p_t. The regions a reverse-time path will travel through are, by construction, the regions where the learned score has seen data. Both of Langevin's failures — barriers and coverage — are dissolved by the same corruption.

20.3 The engine: the reverse-time SDE

Corrupting is trivial; the question is whether the corruption can be reversed. Precisely: does there exist a stochastic process, run from $t = T$ backwards to $t = 0$, whose time-t marginals are the *same* p_t as the forward corruption — so that starting it from cheap noise delivers, at $t = 0$, genuine samples of the data? We derive the answer with machinery the course already owns: Week 9's Fokker–Planck equation, read with fresh eyes.

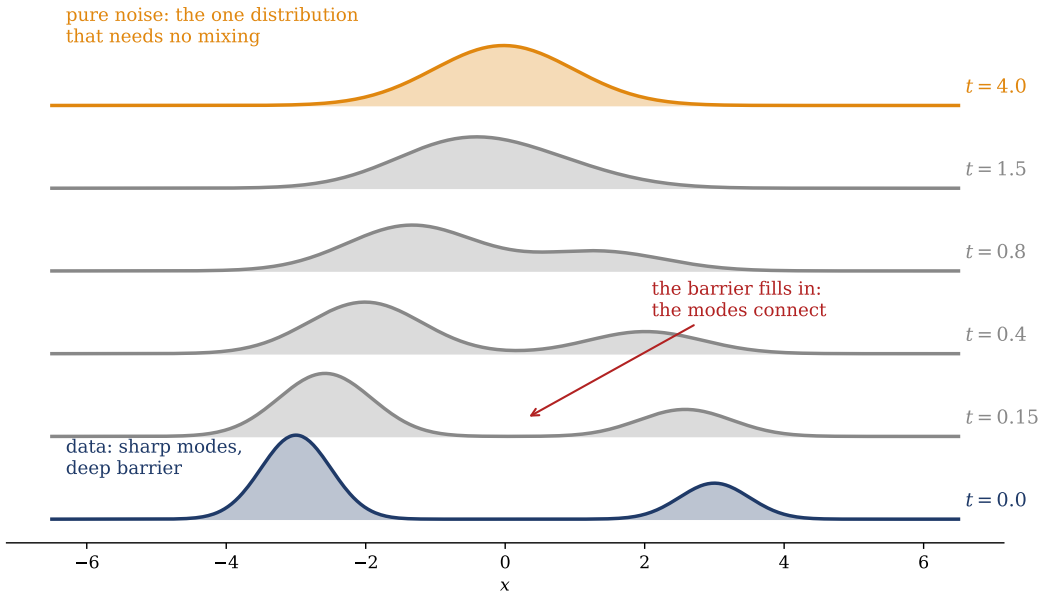

Figure 20.2: The barrier melts. Exact time-marginals p_t of the running two-mode data density $(0.7\,\mathcal{N}(-3, 0.5^2) + 0.3\,\mathcal{N}(3, 0.5^2))$ under the forward process Equation 20.1, drawn bottom to top — a Gaussian mixture stays a Gaussian mixture, with means shrinking as e^{-t} and widths saturating, so every curve is closed-form. The deep valley between the modes — the barrier that blocks equilibrium sampling at $t = 0$ — fills in within half a time unit, and by $t = 4$ the marginal is a unit Gaussian, indistinguishable from noise. A process that travels through these blurs in sequence never confronts the $t = 0$ barrier; that process is the subject of the next section.

Step 1 — the marginal flow, recalled. The forward SDE $\mathrm{d}x = f(x,t)\,\mathrm{d}t + g(t)\,\mathrm{d}W$ (our case: $f = -x$, $g = \sqrt{2}$) transports its marginals according to the Fokker–Planck equation of Equation 17.3:

$$\partial_t\, p_t \;=\; -\nabla_x \cdot (f\, p_t) \;+\; \tfrac{g^2}{2}\, \nabla_x^2\, p_t$$

— a drift current plus diffusive spreading.

Step 2 — the central rewrite. The diffusion term is secretly a drift term. Since $\nabla_x p_t = p_t\, \nabla_x \log p_t$ — the score, converting a density gradient into a transport velocity — we can write $\frac{g^2}{2}\nabla_x^2 p_t = \nabla_x \cdot (\frac{g^2}{2}(\nabla_x \log p_t)\, p_t)$, and the whole Fokker–Planck equation collapses into a *single probability current*:

$$\partial_t\, p_t \;=\; -\nabla_x \cdot \Big[\Big(\underbrace{f - \tfrac{g^2}{2}\nabla_x \log p_t}_{\text{effective velocity}} \Big)\, p_t \Big].$$

Read it physically: the noisy dynamics moves probability *as if* it were a deterministic flow with velocity $f - \frac{g^2}{2}\nabla_x \log p_t$ — the drift, plus the diffusion's net tendency to slide mass down the density gradient, made explicit by the score.

Step 3 — reverse the current. A movie of a flow, played backwards, is the flow with its velocity negated: the time-reversed marginals satisfy the same equation with the current's sign flipped. If we want the *backward* process to be a diffusion too — drift plus fresh noise $\bar{W}$, so that we can simulate it — we must re-add a diffusive term $\frac{g^2}{2}\nabla_x^2 p_t$ and compensate its current the same way we just learned to: absorb another $-\frac{g^2}{2}\nabla_x \log p_t$ into the backward drift. Reversed current plus compensated re-noising gives the backward drift $f - g^2 \nabla_x \log p_t$ — the two half-scores combining into one whole — and the result is **Anderson's theorem** (Anderson 1982):

$$\boxed{\mathrm{d}x \;=\; \Big[f(x,t) \;-\; g(t)^2\, \nabla_x \log p_t(x) \Big]\, \mathrm{d}t \;+\; g(t)\, \mathrm{d}\bar{W},} \tag{20.2}$$

the reverse-time SDE (Anderson 1982)

integrated from $t = T$ *down* to $t = 0$, starting from $p_T \approx \mathcal{N}(0, I)$: its marginal at every intermediate time is exactly p_t, and its output at $t = 0$ is exactly the data distribution. (What $\bar{W}$ and "backward $\mathrm{d}t$" mean as stochastic calculus, and Anderson's original martingale proof, stay at statement level; but the theorem does not have to rest on the flipped-current picture. The discrete-time argument below proves it exactly, using nothing beyond Bayes' rule.)

Physically: *the reversal is the forward drift minus g^2 times the score of the current blur.* At every instant, the score of p_t — a field of arrows pointing toward where the probability mass of the current blur level sits — steers the noise back into data. Generation is time reversal, and the steering field is the previous lecture's Z-free object, evaluated along the corruption path.

The same theorem in discrete time. Anderson proved the reversal with martingale machinery; the argument above flips a probability current. A third route is the most elementary of the three — a statement about Markov chains, carried entirely by Bayes' rule (Welling, Lu, and Holdijk 2026, ch. 9 on Markov processes and time reversal). Chop the forward process into K steps of size Δt with the Euler–Maruyama kernel of Week 9, $F_k(x_k \mid x_{k-1}) = \mathcal{N}(x_{k-1} + f(x_{k-1})\,\Delta t,\; g^2 \Delta t)$, so that the probability of an entire corruption path factorizes in the forward direction,

$$p(x_0, \dots, x_K) \;=\; p_0(x_0) \prod_{k=1}^{K} F_k(x_k \mid x_{k-1}).$$

Now define the *backward kernel* by applying Bayes' rule to a single step,

$$B_k(x_{k-1} \mid x_k) \;=\; \frac{F_k(x_k \mid x_{k-1})\, p_{k-1}(x_{k-1})}{p_k(x_k)}, \tag{20.3}$$

the backward kernel: Bayes' rule on one step

where p_k denotes the time-marginal after k forward steps. Two facts follow by inspection. First, B_k is a genuine transition kernel: integrating over x_{k-1} and using the Chapman–Kolmogorov relation $p_k(x_k) = \int \mathrm{d}x_{k-1}\, F_k(x_k \mid x_{k-1})\, p_{k-1}(x_{k-1})$ gives $\int \mathrm{d}x_{k-1}\, B_k(x_{k-1} \mid x_k) = 1$. Second, growing a chain *backward* from the noise end with these kernels reproduces the forward path probability exactly — substitute Equation 20.3 and telescope:

$$p_K(x_K) \prod_{k=1}^{K} B_k(x_{k-1} \mid x_k) \;=\; \underbrace{p_K(x_K) \prod_{k=1}^{K} \frac{p_{k-1}(x_{k-1})}{p_k(x_k)}}_{= p_0(x_0)} \prod_{k=1}^{K} F_k(x_k \mid x_{k-1}) \;=\; p(x_0, \dots, x_K),$$

every marginal except p_0 cancelling between numerator and denominator. The backward chain matches the forward one not only marginal by marginal but in its full joint law across all times; that the corruption can be reversed is, in discrete time, a two-line identity. The score emerges when the step size shrinks. Write $x_{k-1} = x_k + \delta$ with δ of order $\sqrt{\Delta t}$ and expand the logarithm of Equation 20.3: the Gaussian kernel contributes $-(\delta + f\Delta t)^2/(2g^2\Delta t)$, the marginal contributes $\log p_{k-1}(x_k + \delta) \approx \log p_k(x_k) + \delta \cdot \nabla_x \log p_k(x_k)$ to the order that survives, and completing the square in δ leaves

$$B_k(x_{k-1} \mid x_k) \;\approx\; \mathcal{N}\Big(x_k - \big[\, f(x_k) - g^2\, \nabla_x \log p_k(x_k) \,\big]\, \Delta t,\;\; g^2 \Delta t\Big)$$

— precisely one Euler step of Equation 20.2, run downward in time. Anderson's drift is the small-step limit of Bayes' rule.

One structural feature of Equation 20.3 carries the content of the trap below, and it connects back to Week 8. The backward kernel is built from the *time-marginals* p_{k-1}, p_k — the nonequilibrium blurs of a corruption in progress — not from any stationary distribution. Set it beside the adjoint kernel of Section 15.3 (Equation 15.3): the identical algebraic construction, with the stationary law in place of the marginals. The adjoint reverses a chain *at* stationarity and remains a diffusion; the backward kernel reverses a chain *out of* equilibrium and denoises, and the two coincide only once the marginals stop moving — at which point there is nothing left to generate. A generative model lives precisely in the gap between the two, which is why the score it consumes is that of every intermediate p_t and not of any single equilibrium law. Week 12 meets the conditional cousin of Equation 20.3 — the same Bayes step with the clean data point x_0 pinned — which is closed-form Gaussian and becomes the regression target of the DDPM.

One remark organizes everything a diffusion model does or fails to do: **the theorem is exact.** No approximation was made in reversing time. When Week 12 builds

industrial diffusion models, their *only* approximations will be two: the learned score $s_\theta(x,t) \approx \nabla_x \log p_t(x)$, and the discretization of Equation 20.2 into finite steps. Everything else — the existence of the reversal, the correctness of its marginals — is mathematics. A diffusion model is a theorem plus a regression.

Trap

Locate the approximation correctly — and feed the theorem the right score. Two errors cluster here. ① The reverse-time SDE is exact; diffusion models approximate the *score* and the *discretization*, never the reversal itself — misplacing the approximation misdiagnoses every failure mode. ② The score the theorem consumes is that of the **time-marginals** p_t — the whole ladder of blurred densities — not the data score $\nabla_x \log p_0$ alone. A score known only at $t = 0$ cannot steer a sample out of pure noise; this is *why* multi-scale denoising score matching is not an optimization trick but the load-bearing requirement. (And a bookkeeping cousin: the reverse SDE reproduces *marginals*, not rewound individual trajectories — it is a new stochastic process, not a tape played backwards.)

Take-home 2

Anderson's theorem: the reversal of $\mathrm{d}x = f\,\mathrm{d}t + g\,\mathrm{d}W$ is $\mathrm{d}x = [f - g^2\nabla_x \log p_t]\,\mathrm{d}t + g\,\mathrm{d}\bar{W}$, run from noise at $t = T$ back to data at $t = 0$, with every intermediate marginal reproduced exactly — derived by writing the Fokker–Planck diffusion as a probability current and flipping it. The construction is exact; the only learned ingredient is the score of the blurred marginals. A score is not almost a generator — with a forward process wrapped around it, it *is* one.

20.4 The score it needs is the score we have

A theorem that demands $\nabla_x \log p_t$ for *every* t might seem to have traded one impossible object for a continuum of them. It has traded up, and the previous lecture explains why. Under the forward process Equation 20.1, p_t is the data convolved with a *known* Gaussian — x_t is just $x_0 e^{-t}$ plus Gaussian noise of known width. So fitting $s_\theta(x,t)$ to $\nabla_x \log p_t$ is exactly the previous lecture's **denoising score matching** (Equation 19.4) at noise scale $\sigma(t) = \sqrt{1 - e^{-2t}}$: shrink a data point to $x\,e^{-t}$, add Gaussian noise of width $\sigma(t)$, and ask the network for the direction back, with the closed-form target $-(\tilde{x} - x\,e^{-t})/\sigma(t)^2$ that Vincent's identity provides (Vincent 2011) — the clean point enters *scaled*, exactly as Week 12's discrete $-(x_t - \sqrt{\bar{\alpha}_t}\,x_0)/(1 - \bar{\alpha}_t)$ will. One network, conditioned on t, trained by drawing random data points and random times and regressing on the denoising target — no Jacobian traces, no Z, no sampling loops. The theorem consumes precisely the object the previous lecture's practical variant produces; the two results align, four decades apart.

Historically the fit came first as a ladder rather than a continuum: Song and Ermon (2019) trained scores at a discrete sequence of noise scales and generated by *annealed Langevin* — equilibrate briefly at high noise, step down a scale, repeat — a discrete shadow of the reverse SDE that already produced state-of-the-art samples and pointed at the continuous theorem. What remains for Week 12 is assembly and

industrial detail: the discrete-time DDPM parameterization and its ε-prediction form of the same loss (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015), the probability-flow ODE (a deterministic sibling of Equation 20.2 with the same marginals — named now, derived then), and the schedules and likelihood bookkeeping of Song, Sohl-Dickstein, et al. (2021). The pipeline itself — *data corrupted to noise by a fixed SDE; a t-conditioned score learned by denoising; generation by the reverse-time SDE* — is complete as of this lecture. That pipeline **is** a diffusion model.

> Take-home 3
>
> The reverse SDE needs $\nabla_x \log p_t$ at every t — and for a Gaussian forward process that is exactly denoising score matching at scale $\sigma(t)$: one time-conditioned network, trained on the previous lecture's closed-form target. Corrupt with a fixed SDE, learn the time-score by denoising, generate by Equation 20.2: that pipeline is a diffusion model in all but industrial detail, which is Week 12's.

20.5 Example: transport beats equilibration

The claim can now be tested numerically. To separate the *dynamics* question from the *learning* question, use a target whose scores we know exactly at every t: the running two-mode density $p_0 = 0.7\,\mathcal{N}(-3, 0.5^2) + 0.3\,\mathcal{N}(3, 0.5^2)$, whose forward marginals stay closed-form Gaussian mixtures. Two samplers, both given the **exact score** and an **identical budget** (800 steps, 20{,}000 particles), both started from the same unit Gaussian: ① *equilibration* — plain Langevin at $t = 0$, the previous lecture's "almost a generator"; ② *transport* — the reverse SDE Equation 20.2, integrated from $T = 4$ down to 0.

The result (Figure 20.3) is unambiguous. Langevin, with the true score in hand, reproduces two correctly shaped modes — carrying the *wrong weights*: 0.49 in the right-hand mode against a true 0.30. Its initialization split the particles roughly evenly between the two basins, and in 800 steps essentially no walker crossed the barrier; the sampler is locally perfect and globally frozen, exactly as Section 15.4 taught (and the histogram alone gives no warning). The reverse SDE, at the same cost, lands the mode weight at 0.301 — the truth to three decimals — because its particles never had to cross anything: they *descended the filmstrip*, splitting into the two modes in the correct proportions back when the modes were still connected blurs (the trajectory fan in the right panel shows the split happening around $t \approx 1$, where Figure 20.2 shows the barrier only beginning to form). Mixing was not accelerated; it was made unnecessary.

> *Generation is time reversal: corrupt the data to noise, learn the score of every intermediate blur, and run the film backwards — no Z, and no waiting for mixing.*

The tutorial (16–18, Ph12 106) is derivation-led, with two board problems. First, the reversal the warm-up card attempted — the one-dimensional Ornstein–Uhlenbeck process — done properly and by hand: with the general theorem now available, the exercise is to earn its simplest instance from scratch. Second, fit a two-dimensional score model by denoising score matching and generate from it. The tutorial also

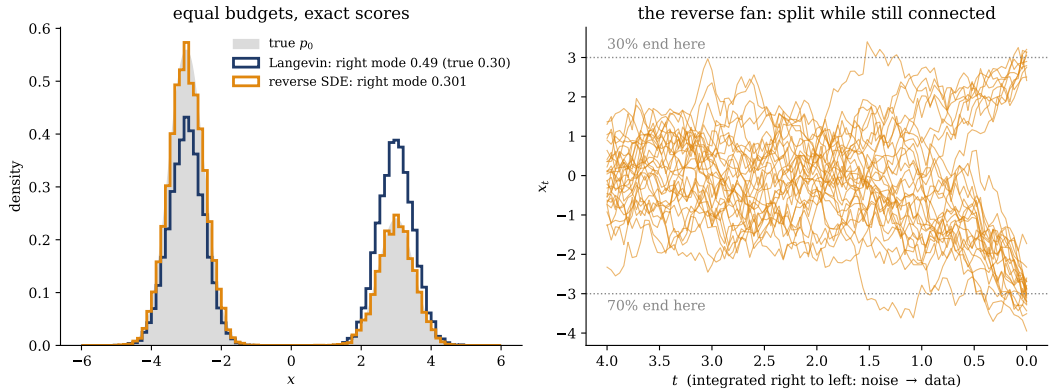

Figure 20.3: Transport versus equilibration, with the exact score and identical budgets (800 steps, 20,000 particles, both started from $\mathcal{N}(0, 1)$). Left: final samples against the true data density (gray). Plain Langevin at $t = 0$ (navy) produces perfectly shaped modes with frozen initialization weights — 0.49 in the right mode against a true 0.30: no barrier crossings in budget, and nothing in the histogram warns you. The reverse-time SDE Equation 20.2 (orange) lands the weight at 0.301. Right: a fan of reverse-SDE trajectories — read right to left, from noise at $t = 4$ to data at $t = 0$ — splitting into the two modes in the correct proportions while the marginals are still connected blurs (compare Figure 20.2). The barrier is never crossed because it is never encountered.

closes a connection left open since Week 3. The tutorial uses Equation 20.2 and Hyvärinen's objective; no devices.

20.6 Outlook: the cost of running time backwards

Week 10 closes with the arc from statistical physics to generative modeling complete at the level of mathematics. The previous lecture removed the partition function from *training*: the score is blind to Z, and Hyvärinen's integration by parts made it learnable. Today removed equilibration from *generation*: Anderson's theorem turns a fixed corruption process into an exact generative one, steered by the learned score, starting from the only distribution that requires no mixing. Between the two lectures, every obstruction this course has fought — the normalizer that blocked the Boltzmann machine, the mixing that priced every sampler — has been either dissolved or bypassed, and the machinery reduces to three components: a forward SDE, a regression loss, and a reverse SDE. The arc from Boltzmann to diffusion is, at this point, a theorem with a training procedure.

What remains of the module is physics — and it is the physics the module is named for. Running a distribution along a nonequilibrium path is not free: it dissipates, and next week the course meets the accounting identities that govern the cost. The **Jarzynski equality** and **Crooks' theorem** relate the *irreversible work* of finite-time transports to *equilibrium free-energy differences* — nonequilibrium trajectories computing equilibrium quantities — and their machine-learning shadow, **annealed importance sampling**, resolves the open question from Module 3: extracting Z itself from chains that never equilibrate (the gap Section 15.4 conceded, now closed).

And Week 12 assembles today's pipeline into the industrial object — DDPM's discrete steps and ε-parameterization, the probability-flow ODE, and the schedules.

21 Fluctuation theorems: buying $\log Z$ with irreversible work

Recommended reading

Core (~1 h). Jarzynski (1997) — four pages, and the warm-up reading for the next lecture: an *equality* where 170 years of thermodynamics had only the inequality $\langle W \rangle \geq \Delta F$. Drive a system from equilibrium at A to B at any finite speed, measure the work trajectory by trajectory, and $\langle e^{-\beta W} \rangle = e^{-\beta \Delta F}$ exactly. The paper derives it from Hamiltonian dynamics; today we take the discrete Markov route instead — read it for the statement and the estimator reading. Alongside it, Crooks (1999) — the sharper parent: not one average but the whole forward and reverse *distributions* of work, locked together and crossing at ΔF. Our engine follows its detailed-balance logic.
Optional background. Jarzynski (2011) — the retrospective review, the cleanest survey of what the work relations do and do not say; Liphardt et al. (2002) and Collin et al. (2005) — the theorems verified by pulling single RNA molecules, equilibrium folding free energies read off irreversible pulls (the experimental hook); Sohl-Dickstein et al. (2015) — read the title and introduction only: the founding diffusion paper, built on exactly this literature; Neal (2001) — the next lecture's subject, the estimator engineered, skim §1 if curious.
Prerequisite reminder. The canonical distribution $p = e^{-\beta E}/Z$ and $F = -T \log Z$ (Week 1); detailed balance and Markov kernels with Boltzmann stationary laws (Chapter 15, Equation 15.1); the forward/reverse process picture of Chapter 20. New content: driving protocols and trajectory work, the Crooks and Jarzynski relations, the abstract fluctuation-theorem identity from which both fall out, dissipation as a forward/reverse KL divergence, and free-energy estimation from nonequilibrium trajectories.

21.1 The remaining problem

Week 10 completed what the course began in Week 3. Its first lecture removed the partition function from *training* — the score $\nabla_x \log p = -\nabla_x E$ is blind to Z (Equation 19.1) — and its second removed it from *generation*, with Anderson's reverse-time SDE (Equation 20.2) turning a fixed corruption process into an exact generator. Between the two, every obstruction the course had fought was dissolved or bypassed — except one, which has been outstanding since Module 3.

Recall the escape from Z that opened Week 8 (Chapter 15): to *sample* $p \propto e^{-\beta E}$ you never need Z, because it cancels in every ratio $p(x')/p(x) = e^{-\beta \Delta E}$. That cancellation is exactly why Markov chain Monte Carlo works — and exactly why it can never hand you Z itself. A sampler reports averages; it does not report the normalizer, and so it cannot compare two models by likelihood, cannot report a

free energy, cannot tell you *how probable* a configuration is, only how probable it is *relative to another*. Module 3 noted this limitation and deferred the resolution.

The problem is tractable once it is restated. We almost never want Z alone; we want *comparisons* — and a comparison of two partition functions is a free-energy difference,

$$\Delta F \;=\; F_B - F_A \;=\; -T\,\log\frac{Z_B}{Z_A}, \tag{21.1}$$

free energy as a ratio of partition functions

a **ratio** of the two forbidden numbers. And here is the lever: if A is a tractable reference — independent spins, a Gaussian, anything whose Z_A is known in closed form — then an estimator of ΔF *is* an estimator of $\log Z_B$. The absolute value of $\log Z_B$ is one subtraction from a ratio that can be estimated.

What nonequilibrium statistical mechanics discovered, in a burst between 1997 and 1999, is that this ratio is cheap to estimate. Jarzynski (1997) proved that if you drive a system from equilibrium at A to the control setting of B in *finite time* — however fast, however far from equilibrium — and record the work W expended along each stochastic trajectory, then

$$\langle e^{-\beta W}\rangle \;=\; e^{-\beta\Delta F},$$

an exact equality at any driving speed, where thermodynamics since Clausius had offered only $\langle W\rangle \geq \Delta F$. Two years later Crooks (1999) found the sharper statement from which Jarzynski's falls out in one line: the entire *distributions* of work in the forward and time-reversed protocols are locked together, and cross at exactly $W = \Delta F$. These are exact identities, not near-equilibrium approximations, and they turn driven, irreversible trajectories into an unbiased estimator of $\log Z$ — the quantity that has been inaccessible throughout the course. Module 3 sampled without Z; this week estimates it.

The identities were not idle theory for long. Liphardt et al. (2002) and then Collin et al. (2005) pulled single RNA hairpins open with optical tweezers — a strongly irreversible operation — and recovered the molecules' equilibrium folding free energies from the work histograms, reading ΔF off the point where the forward and reverse distributions cross. And downstream, the same machinery became an algorithm: Neal (2001) turned Jarzynski's identity into *annealed importance sampling* (the next lecture), and Sohl-Dickstein et al. (2015) built the first diffusion model on this literature — the paper is titled *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. A familiar historical pattern: a physics result waiting decades for machine learning to need it (Onsager 1936 for TAP, Bethe 1935 for belief propagation, Anderson 1982 for the reverse-time SDE, and now Jarzynski and Crooks for the diffusion training bound).

Figure 21.1 summarizes the setup. A control parameter $\lambda(t)$ is driven from A to B in finite time — a trap stiffened, a piston pushed, a field ramped. Drive it slowly and the system stays in the instantaneous equilibrium $\pi_{\lambda(t)}$ throughout; drive it fast and the actual distribution $p(x,t)$ *lags* behind, and the shaded gap between them is paid as dissipated work. The reverse protocol, run from equilibrium at B back to A, generates a *different* sequence of lagging distributions — it is not the forward film played backwards. Crooks' theorem is the precise statement that the dissipated work measures exactly how different those two films are.

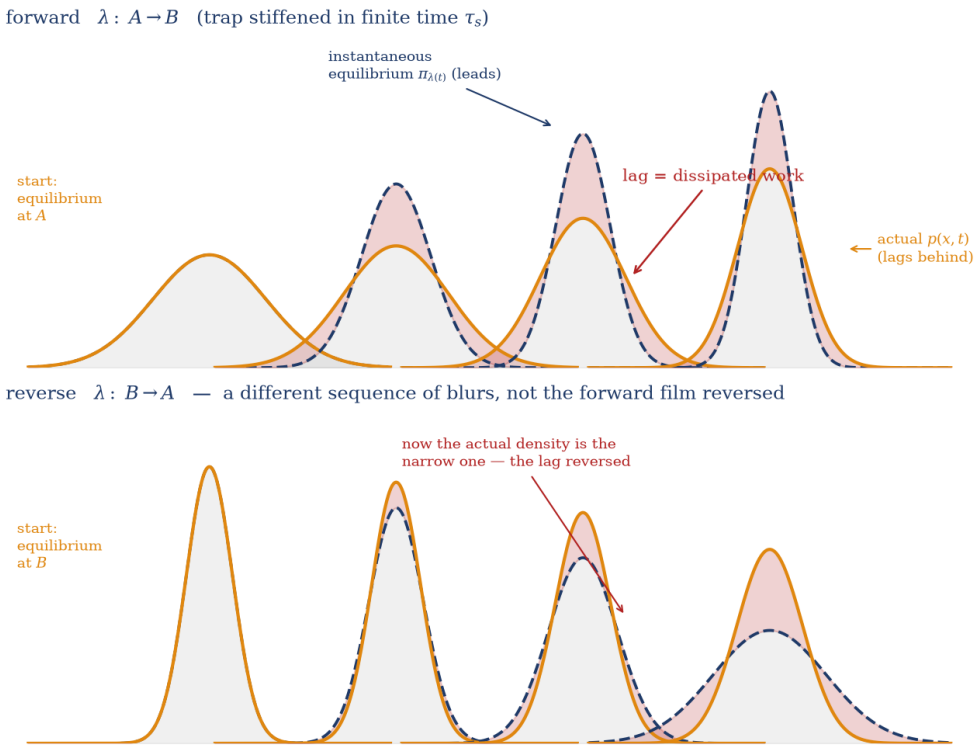

Figure 21.1: The central picture. A control parameter $\lambda(t)$ is driven in finite time, here a harmonic trap $E_\lambda(x) = \frac{1}{2}k(\lambda)\,x^2$ stiffened from A to B (top) and softened from B back to A (bottom); the curves are the exact time-marginals of the overdamped dynamics. Top: the instantaneous equilibrium $\pi_{\lambda(t)}$ (navy dashed) contracts as the trap stiffens, but the actual density $p(x,t)$ (orange) lags behind, still too wide — the red gap is the lag that the dissipated work pays for (it bounds the dissipation, Equation 21.5). Bottom: run in reverse the lag flips sign — the actual density is now the *narrow* one, still remembering the stiff trap it started in. The forward and reverse sequences of blurs are genuinely different, and (Crooks) their difference is exactly what the second law charges for finite-time driving. This is Week 10's filmstrip with a price tag.

Classical thermodynamics answers the finite-time question with a bound and two useless limits. The bound is $\langle W \rangle \geq \Delta F$, with equality only in the quasistatic limit. So to extract ΔF classically you have two options, and both fail. Drive infinitely slowly (thermodynamic integration): exact, but it costs infinite time. Or switch instantly and reweight (free-energy perturbation, Zwanzig (1954)): fast, but its variance is hopeless. The question: *what does a finite-time, irreversible protocol know about* ΔF*?* ΔF is fully determined by the work fluctuations, provided one averages the right function.

> Take-home 1
>
> Free-energy differences are ratios of partition functions, $\Delta F = -T \log(Z_B/Z_A)$ (Equation 21.1); with a tractable reference A, estimating ΔF estimates $\log Z_B$ — the number MCMC (Module 3) could never deliver, because Z cancels in the ratios sampling relies on. The fluctuation theorems extract that number from *finite-time, irreversible* driving: the information the second law's inequality discards survives in the fluctuations of the work.

21.2 The discrete protocol: work and heat on the course's own machinery

To track a driven system exactly we discretize the protocol into two elementary moves and account for each on the vocabulary Week 8 already built. Introduce an interpolating family of energies

$$E_k(x) = E(x; \lambda_k), \qquad k = 0, 1, \dots, N, \qquad E_0 = E_A,\ E_N = E_B,$$

with the corresponding equilibrium distributions $p_k(x) = e^{-\beta E_k(x)}/Z_k$. This is the ladder annealed importance sampling will engineer in the next lecture and the ladder diffusion makes continuous in Week 12; for now it is just a fine subdivision of the drive from A to B. A single step of the protocol is the composition of two moves:

① **Work step.** Switch the energy $E_{k-1} \to E_k$ at *frozen* configuration x. The system does not move, so no heat is exchanged; the change in energy is charged to the driver as work,

$$w_k = E_k(x) - E_{k-1}(x).$$

② **Heat step.** Move the configuration by a Markov kernel $T_k(x \to x')$ that is detailed-balanced with respect to p_k — Week 8's boxed condition (Equation 15.1), $T_k(x \to x')\, p_k(x) = T_k(x' \to x)\, p_k(x')$, applied at fixed λ_k. The energy changes because x moves, and that change is heat exchanged with the bath; no work is done.

A forward trajectory draws $x_0 \sim p_0$ from equilibrium at A and then alternates: work step to λ_1, heat step at λ_1 producing x_1, work step to λ_2, heat step producing x_2, and so on to x_N. The total work along the trajectory is the sum of the work steps only,

$$W[x] \;=\; \sum_{k=1}^{N} \big[\, E_k(x_{k-1}) - E_{k-1}(x_{k-1}) \,\big], \tag{21.2}$$

each term evaluated at the configuration *before* the heat step that follows. This is the trajectory analogue of the first law: the energy change from moving λ at frozen x is work, the energy change from moving x at fixed λ is heat, and W collects the former. Conflating W with the naive endpoint difference $E_B(x_N) - E_A(x_0)$ — which also folds in all the heat — breaks every identity below; the work is the driver's ledger, not the system's energy budget.

The *reverse* process starts from equilibrium at B, $x_N \sim p_N$, and runs the protocol backwards, $\lambda_N \to \lambda_0$, applying the same kernels in reversed order. One point matters now because it is persistently misread: **only the initial state of each process need be in equilibrium.** The forward process starts from p_A and the reverse from p_B; nothing is assumed to equilibrate along the way, and nothing need reach equilibrium at the far end. The theorems hold regardless of how far from equilibrium the driving pushes the system. Every object on the board so far is Week 8's — Boltzmann weights and detailed-balanced kernels — and the only new element is the bookkeeping of a moving λ. With that in place the ratio of the forward and reverse path probabilities is a three-line computation.

21.3 The engine: the path ratio, Crooks, Jarzynski

Compare the probability of a forward trajectory with the probability of the *same* configurations visited in reverse order under the reverse protocol.

Step 1 — path probabilities. The forward trajectory $x = (x_0, x_1, \dots, x_N)$ has probability

$$P_F[x] \;=\; p_0(x_0) \prod_{k=1}^{N} T_k(x_{k-1} \to x_k),$$

the equilibrium start times the product of the heat-step kernels. The reversed trajectory $\tilde{x} = (x_N, \dots, x_0)$ under the reverse protocol has probability

$$P_R[\tilde{x}] \;=\; p_N(x_N) \prod_{k=1}^{N} T_k(x_k \to x_{k-1}),$$

with the same kernels traversed in the opposite direction.

Step 2 — detailed balance flips each kernel. The one identity we use, once per step, is Equation 15.1:

$$T_k(x_k \to x_{k-1}) \;=\; T_k(x_{k-1} \to x_k)\, \frac{p_k(x_{k-1})}{p_k(x_k)}.$$

Every forward kernel factor cancels against its reverse partner; what survives is a ratio of Boltzmann weights.

Step 3 — the telescoping. Substituting and cancelling the kernels,

$$\frac{P_R[\tilde{x}]}{P_F[x]} \;=\; \frac{p_N(x_N)}{p_0(x_0)} \prod_{k=1}^{N} \frac{p_k(x_{k-1})}{p_k(x_k)} \;=\; \prod_{k=1}^{N} \frac{p_k(x_{k-1})}{p_{k-1}(x_{k-1})} \;=\; \prod_{k=1}^{N} \frac{Z_{k-1}}{Z_k}\, e^{-\beta[E_k(x_{k-1}) - E_{k-1}(x_{k-1})]}.$$

The middle equality is the telescoping: the boundary factors $p_N(x_N)$ and $1/p_0(x_0)$ are absorbed as the $k = N$ and $k = 1$ ends of the shifted product $\prod_k p_k(x_{k-1})/p_{k-1}(x_{k-1})$. Now the two surviving pieces are exactly the

quantities we named. The product of partition-function ratios telescopes to $Z_0/Z_N = Z_A/Z_B = e^{\beta\Delta F}$ (Equation 21.1), and the exponent is precisely the total work (Equation 21.2). Both collapse:

$$\boxed{\frac{P_R[\tilde{x}]}{P_F[x]} \;=\; e^{-\beta\,(W[x]-\Delta F)}} \tag{21.3}$$

Crooks fluctuation theorem (path form)

How much less probable the reversed movie is than the forward one is set by exactly the work you wasted: dissipation *is* time-reversal asymmetry, measured trajectory by trajectory. Since the work is odd under reversal, $W[\tilde{x}] = -W[x]$, binning trajectories by their work value converts Equation 21.3 into the form the experiments use,

$$\frac{P_F(W)}{P_R(-W)} \;=\; e^{\beta(W-\Delta F)},$$

and the two work distributions **cross at** $W = \Delta F$, where the exponent vanishes. Free energy is read off an intersection — which is exactly what Collin et al. (2005) did with RNA.

Jarzynski falls out by summation. Multiply Equation 21.3 through by $P_F[x]$ and sum over all trajectories:

$$\langle e^{-\beta W}\rangle_F \;=\; \sum_x P_F[x]\, e^{-\beta W[x]} \;=\; e^{-\beta\Delta F}\sum_x P_R[\tilde{x}] \;=\; e^{-\beta\Delta F},$$

since the reverse path measure is normalized. This is

$$\boxed{\langle e^{-\beta W}\rangle \;=\; e^{-\beta\Delta F}, \quad \text{at any driving speed}} \tag{21.4}$$

Jarzynski equality

— the reweighted forward ensemble *is* the reverse ensemble, and normalization of the latter is the equality. **The second law is its convexity shadow.** The exponential is convex, so Jensen's inequality gives $\langle e^{-\beta W}\rangle \geq e^{-\beta\langle W\rangle}$, hence $e^{-\beta\Delta F} \geq e^{-\beta\langle W\rangle}$, i.e. $\langle W\rangle \geq \Delta F$: Clausius recovered, not as a separate law but as the average's shadow of a trajectory-level equality. An equality that survives any driving speed, derived from nothing but detailed balance and bookkeeping — the same two ingredients as Week 8. In the next lecture $e^{-\beta W}$ becomes an importance weight and this derivation becomes an algorithm.

The abstract form. Before the consequences, it is worth seeing how little of that derivation was physics, because the stripped-down version explains why both theorems have the shape they have — and why the course keeps meeting them (Welling, Lu, and Holdijk 2026, ch. 17 on fluctuation theorems). Let ρ and ω be *any* two normalized path measures with common support, and define on paths drawn from ρ the log-ratio $\Sigma = \log(\rho/\omega)$ — in the general theory this object is called the *entropy production*, the name promised twice in Module 3. Then

$$\langle e^{-\Sigma}\rangle_\rho \;=\; \int \mathrm{d}x_{0:N}\;\rho\cdot\frac{\omega}{\rho} \;=\; \int \mathrm{d}x_{0:N}\;\omega \;=\; 1$$

the integral fluctuation theorem: one line, no physics

— an *integral fluctuation theorem*, in one line and with no physical input, and Jensen applied to it gives $\langle\Sigma\rangle_\rho = D_{\mathrm{KL}}(\rho\|\omega) \geq 0$: a second law for any pair of measures, which is nothing but the positivity of a KL divergence. The distribution-level statement costs one more line. If $P(\chi)$ is the law of Σ under ρ, the display above says

the nonnegative function $P(\chi)\,e^{-\chi}$ integrates to one, so $Q(\chi) \equiv P(-\chi)\,e^{\chi}$ is itself a normalized distribution and

$$\frac{P(\chi)}{Q(-\chi)} \;=\; e^{\chi}$$

— a *detailed fluctuation theorem*, and integrating it back recovers the integral one: the two statements are equivalent. Everything thermodynamic enters through a single choice, *which* ω. Choose ω to be the reverse-protocol path measure with equilibrium endpoints and Σ becomes $\beta(W - \Delta F)$ — Step 3's computation — while Q becomes the law of the reverse-protocol work: Crooks and Jarzynski, verbatim. (For Q to describe a process one can actually run, the map $\rho \mapsto \omega$ must be an *involution* — applying it twice returns ρ — which time reversal is; that is why the forward and reverse work distributions pair off so symmetrically.) The module instantiates the same identity at other choices of ω: The next lecture chooses the reverse *annealing* path measure, turning the one-line normalization into the unbiasedness of an estimator and $\langle\Sigma\rangle$ into its Jensen gap; and Week 10's backward kernel is the unique choice that makes $\Sigma \equiv 0$ on every path — a reversal with nothing to dissipate. One identity yields a family of theorems, each selected by a choice of measure.

Trap

The equality does not beat the second law — and its guarantee is only about the average. Three errors cluster here. ① Reading Equation 21.4 as "free energy from work, for free" inverts it: the *exponential* average equals $e^{-\beta\Delta F}$, and Jensen then forces $\langle W\rangle \geq \Delta F$ on the ordinary average. Individual trajectories may have $W < \Delta F$ — and *must*, for the exponential average to come out as low as it does — but no average of the work ever undercuts ΔF. ② The work is Equation 21.2, the ledger of moving λ at frozen x; it is not the endpoint energy difference, which also contains the heat. ③ Detailed balance is used per rung, at fixed λ_k, as the *identity* in Step 2 — the kernels need not converge to p_k, and typically do not. It is precisely because no equilibration along the path is required that the result holds at arbitrary driving speed.

Take-home 2

For a driven protocol with detailed-balanced kernels, one detailed-balance flip per step telescopes the forward/reverse path ratio to $P_R/P_F = e^{-\beta(W-\Delta F)}$ (Crooks, Equation 21.3): dissipated work measures time-reversal asymmetry exactly. Summing against P_F gives $\langle e^{-\beta W}\rangle = e^{-\beta\Delta F}$ (Jarzynski, Equation 21.4) at any driving speed, and Jensen's inequality recovers $\langle W\rangle \geq \Delta F$. The second law is the convexity shadow of an exact trajectory-level equality.

21.4 Consequences: the price of reversal, and an estimator for Z

Two things fall out of the engine — a conceptual identity that prices Week 10's time reversal, and the estimator that resolves Week 8's open question, together with the practical limitation.

Dissipation is a KL divergence. Take the logarithm of Equation 21.3 and average over the forward ensemble. The left side is $\langle \log(P_F/P_R) \rangle_F$, which is by definition the Kullback–Leibler divergence $D_{\mathrm{KL}}(P_F \,\|\, P_R)$ between the forward and reverse path measures; the right side is $\beta(\langle W \rangle_F - \Delta F)$. Hence

$$\beta \langle W_{\mathrm{diss}} \rangle \;=\; \beta\,(\langle W \rangle_F - \Delta F) \;=\; D_{\mathrm{KL}}(P_F \,\|\, P_R) \;\geq\; 0, \tag{21.5}$$

dissipation as forward/reverse KL

the second law written in the course's own D_{KL} notation. A process is reversible if and only if its forward and reverse path measures coincide, and the dissipated work is precisely their divergence. This is the exact price tag on the reversal Week 10 built: Anderson's theorem made the reversal exist as mathematics, and Equation 21.5 says what finite-time driving costs — the asymmetry between the forward and reverse path ensembles, measured in nats of relative entropy.

The estimator. Rearrange Equation 21.4. Run M driven trajectories from equilibrium at a *tractable* reference A, record each work W_m, and form

$$\widehat{\Delta F} \;=\; -T \,\log\!\Big(\tfrac{1}{M} \textstyle\sum_{m=1}^{M} e^{-\beta W_m} \Big). \tag{21.6}$$

Because $\frac{1}{M}\sum_m e^{-\beta W_m}$ is an unbiased estimator of $\langle e^{-\beta W} \rangle = e^{-\beta \Delta F}$, this is a consistent estimator of ΔF — and hence of $\log Z_B$ — *with no quasistatic requirement whatsoever.* Week 8's open question is formally answered: nonequilibrium trajectories deliver the number that equilibrium sampling could not.

The catch. Formally paid is not practically paid, and Figure 21.4 makes the gap concrete. The average $\langle e^{-\beta W} \rangle$ is dominated by trajectories in the far *left* tail of $P_F(W)$ — the rare, "second-law-beating" ones with $W < \Delta F$ — because the weight $e^{-\beta W}$ is largest exactly where W is smallest. By Crooks, the reweighted integrand $e^{-\beta W} P_F(W)$ is proportional to $P_R(-W)$: the average lives where the *reverse* process lives, which under strong forward driving is a region the forward process almost never visits. Drive hard and the trajectories that carry the answer are ones you essentially never sample; the estimator stays unbiased in $e^{-\beta \Delta F}$, but its variance grows like $e^{\beta \langle W_{\mathrm{diss}} \rangle}$ (a rule of thumb, not derived here), and the finite-sample estimate of ΔF in Equation 21.6 is biased *high* — a heavy-tailed sample mean systematically undershoots $\langle e^{-\beta W} \rangle$, and $-T \log$ of too small a number is too large. The direction is not incidental: since $-\log$ is convex, Jensen gives $\mathbb{E}[\widehat{\Delta F}] \geq \Delta F$ for any sample size, so the estimator is unbiased in Z and biased in F. The cure is to keep the dissipation per stage small — many interpolating distributions, a heat step between each, so no single step lags far. That is the ladder of Section 21.2, engineered, and it is *annealed importance sampling* (Neal (2001)), whose importance weights are precisely $e^{-\beta W}$. The next lecture builds it; the tutorial runs it.

> **Take-home 3**
>
> Averaging the log of the Crooks ratio gives $\beta \langle W_{\mathrm{diss}} \rangle = D_{\mathrm{KL}}(P_F \| P_R) \geq 0$ (Equation 21.5): dissipation is the KL divergence between forward and reverse path ensembles, so reversibility is measure-equality and the second law is a KL positivity. The estimator $\widehat{\Delta F} = -T \log \overline{e^{-\beta W}}$ (Equation 21.6) is unbiased at any driving speed but carried by rare low-work trajectories; its variance explodes with dissipation. Managing that variance is annealed importance sampling (the next lecture).

21.5 Example: the switched harmonic oscillator

One concrete driven system makes the identities visible. Take a particle in a harmonic trap $E_\lambda(x) = \frac{1}{2}k(\lambda)\,x^2$ at temperature $T = 1$, and ramp the stiffness linearly from $k_A = 1$ to $k_B = 16$ over N increments; between increments the particle relaxes by overdamped Langevin dynamics, which for a harmonic trap has an exact Ornstein–Uhlenbeck update, so the heat-step kernels are detailed-balanced exactly. The free-energy difference is known in closed form: $Z_\lambda \propto k(\lambda)^{-1/2}$, so

$$\Delta F = \tfrac{T}{2}\log\frac{k_B}{k_A} = \tfrac{1}{2}\log 16 \approx 1.386.$$

This is the number the trajectories must reproduce. The numerical example below shows the *physics* — the Crooks crossing, and the bias-free-but-not-cost-free character of the estimator; the estimator *engineering* (annealed importance sampling versus one-shot switching, how many rungs) is the next lecture's, and the students' own numerical verification is the tutorial's.

Figure 21.2 shows the Crooks theorem directly: at a moderate driving speed the forward work distribution $P_F(W)$ and the reversed reverse-protocol distribution $P_R(-W)$ intersect precisely at ΔF, and the log-ratio (inset) is a straight line of slope $\beta = 1$, exactly as Equation 21.3 demands.

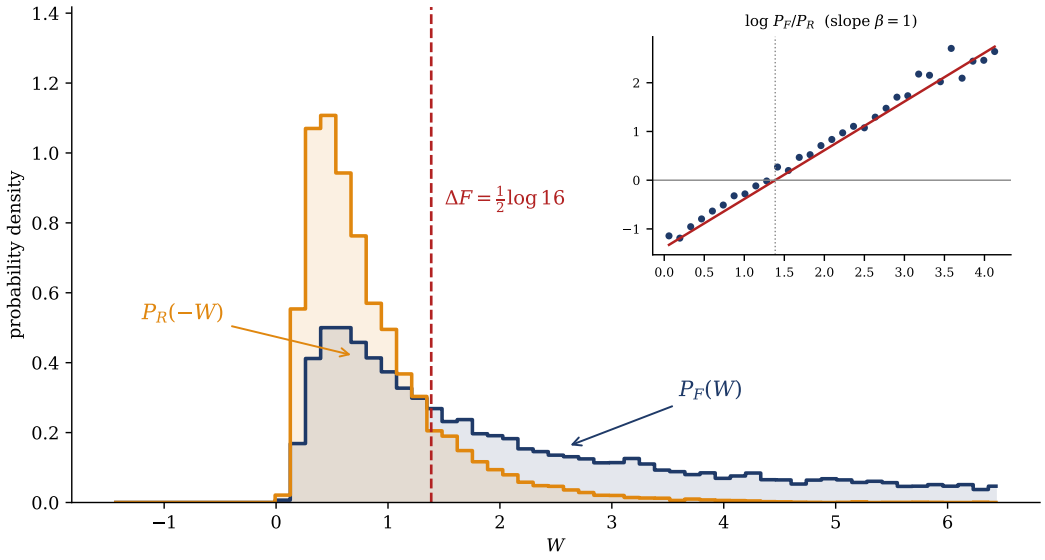

Figure 21.2: The Crooks crossing on the switched oscillator (stiffness $1 \to 16$, $\tau_s = 0.4$, $N = 40$ increments, $M = 15{,}000$ trajectories per direction). The forward work distribution $P_F(W)$ (navy) dissipates and sits to the right; the reversed distribution $P_R(-W)$ (orange) sits to the left; they intersect at the exact $\Delta F = \frac{1}{2}\log 16 \approx 1.386$ (red line), where the Crooks exponent $\beta(W - \Delta F)$ vanishes. Inset: the log-ratio $\log[P_F(W)/P_R(-W)]$ over the well-sampled region falls on a straight line of slope $\beta = 1$ (red), the signature of Equation 21.3. The single-molecule experiments recover folding free energies from exactly this intersection.

Figure 21.3 isolates the cost. As the protocol is driven faster, the mean work $\langle W \rangle$ climbs well above ΔF — that is the dissipation of Equation 21.5 growing — while

the Jarzynski estimate Equation 21.6 sits on ΔF at every speed. The equality is unbiased no matter how hard the drive. What speed costs, at this sample size, is variance: the bootstrap error bars widen as the driving accelerates, while the estimate itself holds on ΔF. There is a finite-sample bias too, and it is worth getting the sign right — it runs *upward*. The average $\langle e^{-\beta W}\rangle$ is carried by rare low-work trajectories; a finite sample typically misses them, so the sample mean of $e^{-\beta W}$ comes out too small and $\widehat{\Delta F} = -\frac{1}{\beta}\log \overline{e^{-\beta W}}$ too *large* — Jensen's inequality makes this a theorem, $\mathbb{E}[\widehat{\Delta F}] \geq \Delta F$ (the Gore–Ritort–Bustamante bias). At $M = 4000$ on this near-Gaussian oscillator that bias sits below the noise floor; it is what becomes severe under heavier-tailed driving, and exactly the failure annealed importance sampling exists to cure.

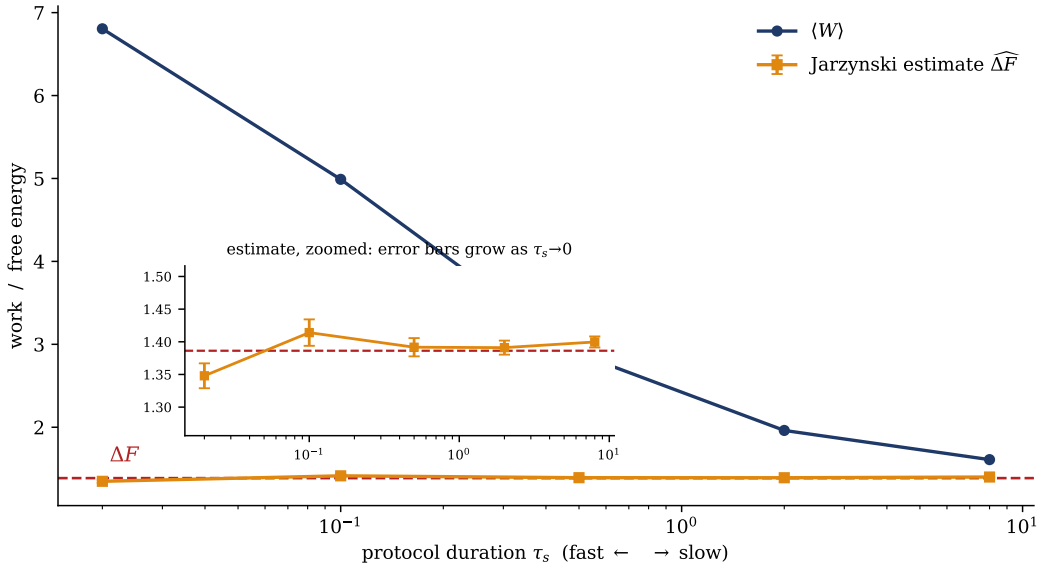

Figure 21.3: Bias-free is not cost-free. Mean work $\langle W\rangle$ (navy) and the Jarzynski free-energy estimate Equation 21.6 (orange, with bootstrap error bars) versus protocol duration τ_s, for the same $1 \to 16$ switch ($N = 40$, $M = 4000$ trajectories, 300 bootstrap resamples). Slow driving (right) sits near equilibrium and $\langle W\rangle \to \Delta F$; fast driving (left) dissipates, and $\langle W\rangle$ rises far above the red reference $\Delta F = \frac{1}{2}\log 16$. The Jarzynski estimate holds on ΔF across three decades of speed — the equality is exact at any τ_s — while its error bars grow toward fast driving; the finite-sample bias runs *upward* (Jensen: $\mathbb{E}[\widehat{\Delta F}] \geq \Delta F$), from the ever-rarer low-work trajectories that carry the average, and stays below the noise floor at this M.

Finally, Figure 21.4 shows *why* strong driving makes the estimate fragile. It overlays the forward work distribution with the reweighted integrand $e^{-\beta W}P_F(W)$ that the Jarzynski average actually integrates. Under fast driving the forward trajectories dissipate to a mean $\langle W\rangle \approx 5$, far to the right of ΔF, but the reweighted integrand collapses onto the *low-work* trajectories near $W \approx 0.5$: the estimator effectively discards the dissipated majority and leans on the low-work minority. For this near-Gaussian oscillator the imbalance is still manageable — which is why Figure 21.3 stays close to ΔF — but the same mechanism, under a heavier-tailed protocol, is what makes the variance explode and forces the move to annealed importance sampling.

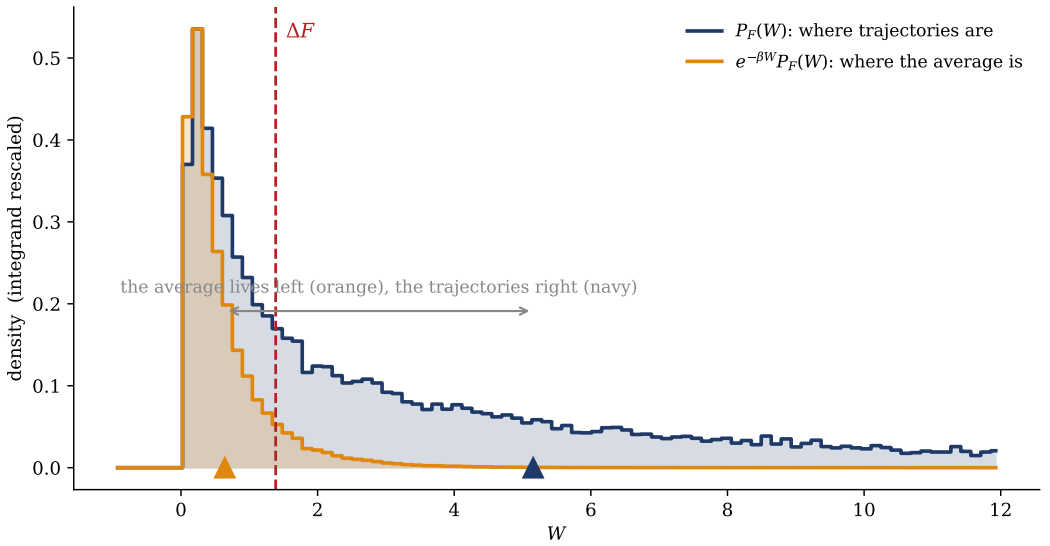

Figure 21.4: Where the average lives. For a fast switch ($\tau_s = 0.1$, $M = 20{,}000$), the forward work distribution $P_F(W)$ (navy) dissipates: its mass stretches far to the right of ΔF (red line), with mean $\langle W \rangle$ (navy tick) well beyond it. But the Jarzynski average integrates the reweighted density $e^{-\beta W} P_F(W)$ (orange, rescaled to unit peak for display), which by Crooks is proportional to $P_R(-W)$ and collapses onto the low-work end, with reweighted mean (orange tick) near and below ΔF. The estimator is carried by the low-work minority and all but ignores the dissipated bulk it actually sampled — the origin of the variance that Section 21.4 flagged. For this near-Gaussian oscillator the mismatch is still mild; under stronger dissipation it is what annealed importance sampling exists to manage.

$\langle e^{-\beta W} \rangle = e^{-\beta \Delta F}$ at any driving speed — equilibrium free energies, ratios of partition functions, bought with irreversible work; the second law is the Jensen shadow of an equality.

The tutorial (Ph12 106) is computation-led. You will verify the Jarzynski equality numerically on a system of your own and compute a free-energy difference by annealed importance sampling — the next lecture builds that algorithm from today's ratio — and make explicit a connection this lecture has only named. The tutorial uses Equation 21.3, the work definition Equation 21.2, and the estimator Equation 21.6.

21.6 Outlook: from identity to algorithm

Today's lecture converted ΔF into a computable estimator. The partition function the course could not evaluate becomes, through Equation 21.1, a free-energy difference; the fluctuation theorems (Equation 21.3, Equation 21.4) compute that difference from finite-time, irreversible driving; and with a tractable reference the estimator Equation 21.6 delivers $\log Z$ itself. To summarize the course's treatment of Z: Z was dodged in Module 1 (contrastive divergence), bounded in Module 2 (the variational free energy), cancelled in Module 3 (sampling), removed from training and generation in Week 10 (score matching and the reverse-time SDE), and now, as a ratio, **measured**.

What remains is engineering and assembly. The next lecture takes the estimator's one weakness — variance that explodes with dissipation (Section 21.4) — and cures it by engineering the ladder of Section 21.2: many interpolating distributions with a detailed-balanced move on each rung, so that no single step dissipates much. That is *annealed importance sampling* (Neal (2001)), and its importance weights are exactly the $e^{-\beta W}$ of Equation 21.6; it also closes the loop Week 8 opened, where annealing *optimized* and here the same construction *measures*. And Week 12 assembles the module's machinery into the industrial object: diffusion models, whose training bound is a nonequilibrium free-energy estimator in precisely the sense derived today, and whose lineage runs straight back through Sohl-Dickstein et al. (2015) to the identities on this page. The reversal Week 10 proved exact has, as of today, a price — Equation 21.5 — and the whole of Module 4 is the account of paying it.

22 Annealed importance sampling: the equality becomes an estimator

Recommended reading

Core (~1 h). Neal (2001) — the algorithm and the extended-state-space proof; §1–2 are today's engine, and Neal notes the correspondence to Jarzynski's equality explicitly. Read alongside it Jarzynski (1997) — the previous lecture's warm-up reading, worth a second pass after this lecture with the dictionary in hand: it is the same proof.
Optional background. Crooks (1999) — the detailed fluctuation theorem behind the previous lecture's work-distribution ratio; today it becomes an error analysis. Sohl-Dickstein et al. (2015) — the founding diffusion-model paper; read the abstract and §1 only, as a pre-echo of Week 12: its title names today's subject. Welling, Lu, and Holdijk (2026) — the companion text's fluctuation-theorem and free-energy-estimation chapters treat exactly this material.
Prerequisite reminder. All of the previous lecture (Chapter 21): trajectory work along a driven protocol (Equation 21.2), the Jarzynski equality (Equation 21.4), the Crooks theorem behind it (Equation 21.3), the second law $\langle W\rangle \geq \Delta F$ as its Jensen corollary, and the dominance of rare low-work trajectories in the exponential average. From Module 3: Markov kernels, detailed balance, and the Z-cancellation in acceptance ratios (Section 15.3); the annealing ladder of Chapter 16. From Module 1: the evaluation gap of Section 6.4 — a trained RBM whose log-likelihood nobody could compute. New content: importance sampling and its variance, the AIS algorithm, the extended-state-space proof of its unbiasedness, the dissipation–variance–bias accounting that prices the estimator, and the variance-optimal backward kernel that bridges to Week 12.

22.1 From equality to estimator

The previous lecture proved an equality; this lecture turns it into an estimator. For any driven protocol, however fast and however irreversible, the work along stochastic trajectories satisfies $\langle e^{-W}\rangle = e^{-\Delta F}$ — an exact statement where classical thermodynamics offered only the inequality $\langle W\rangle \geq \Delta F$, with the Crooks theorem (Equation 21.3) standing behind it and the second law recovered as its Jensen shadow. (Throughout this lecture we absorb β into the energies, so the previous lecture's $\langle e^{-\beta W}\rangle = e^{-\beta\Delta F}$ (Equation 21.4) reads $\langle e^{-W}\rangle = e^{-\Delta F}$; nothing is lost, and the symbols stay light.) The warm-up card sent you into Jarzynski's four pages with the derivation already on the board, and the natural question to carry out of that reading is the one that organizes today: if every single run dissipates — if $W > \Delta F$ on essentially every trajectory you will ever simulate — where does an exact *equality* live? The previous lecture's answer was that it lives in the rare trajectories with $W < \Delta F$, which the exponential average amplifies. That answer matters twice: it

explains why the equality is useful, and it identifies the estimator's practical limitation.

The equality addresses a problem the course has carried since Module 1, with three specific instances. Week 3 trained a restricted Boltzmann machine around its partition function — contrastive divergence dodges the negative phase — but flagged that the trained model's log-likelihood was out of reach, because $\log p(v) = \log f(v) - \log Z$ needs the one number training never produced (Section 6.4). Week 8 sharpened the complaint into a principle: MCMC samples $p \propto e^{-E}$ precisely because Z cancels in every acceptance ratio, and for exactly that reason a chain can never *deliver* Z — samples are not normalizers (Section 15.4). Week 10 removed Z from training and from generation (Equation 19.1, Equation 20.2) but pointedly did not produce a number for it: a score-matched model has no likelihood to report. Three modules of machinery, and the course still cannot compare two energy-based models, report a free energy, or evaluate the RBM it trained nine weeks ago. All three require the same object the previous lecture's theorem computes: a ratio of partition functions, Z_K/Z_0, with the reference Z_0 known in closed form.

The history has an unusual shape here. This course has repeatedly met results that waited decades for their application — Onsager's 1936 reaction field for TAP, Bethe's 1935 lattice approximation for belief propagation, Anderson's 1982 reversal theorem for diffusion models. This week the pattern compresses to a year or so: Jarzynski (1997) proved the equality for driven Hamiltonian systems, and Neal (2001, circulated as a preprint in 1998) constructed the same identity inside Markov-chain Monte Carlo, as a variance repair of importance sampling, noting the correspondence to Jarzynski himself. Two fields arrived at one identity nearly simultaneously and met only afterwards. Neal's construction is **annealed importance sampling** (AIS), and the task is to build the dictionary between the previous lecture's physics and Neal's algorithm, prove the algorithm exact by the same argument, and use Crooks to characterize its variance. Week 10 removed Z where it could be removed; Week 11 estimates it where it must be estimated.

22.2 One jump: importance sampling and its collapse

We start with the obvious attempt, because its failure dictates everything that follows. Suppose we can sample a tractable reference $\pi_0 = f_0/Z_0$ (a Gaussian, independent spins — anything with Z_0 in closed form) and we want the normalizer of a target $\pi_K = f_K/Z_K$ of which we hold only the unnormalized f_K. *Importance sampling* answers in one line: draw $x \sim \pi_0$ and average the *importance weight* $w = f_K(x)/f_0(x)$,

$$\langle w \rangle_{\pi_0} = \int \mathrm{d}x\, \pi_0(x)\, \frac{f_K(x)}{f_0(x)} = \frac{1}{Z_0} \int \mathrm{d}x\, f_K(x) = \frac{Z_K}{Z_0}.$$

The estimator is unbiased for any π_0 whatsoever, and it already has the previous lecture's structure: for the instantaneous protocol that switches $E_0 \to E_K$ in a single step, $\log w = -[E_K(x) - E_0(x)] = -W$, and the display above *is* the Jarzynski equality (Equation 21.4) for the fastest possible driving. So where is the catch? Compute the second moment the same way:

$$\langle w^2 \rangle_{\pi_0} = \int \mathrm{d}x\, \pi_0(x)\, \frac{f_K(x)^2}{f_0(x)^2} = \frac{Z_K^2}{Z_0^2} \int \mathrm{d}x\, \frac{\pi_K(x)^2}{\pi_0(x)},$$

so that the *relative* variance is

$$\frac{\mathrm{Var}[w]}{\langle w \rangle^2} = \int \mathrm{d}x\, \frac{\pi_K(x)^2}{\pi_0(x)} - 1 \;\equiv\; \chi^2(\pi_K \| \pi_0), \tag{22.1}$$

relative variance of one-jump importance sampling

the *chi-squared divergence* between target and reference. The formula is exact, but its variance is severe: the integrand puts π_K^2 over π_0, so any region where the target has mass and the reference has almost none contributes enormously. For two unit Gaussians a distance μ apart the integral closes — completing the square in $\int \mathrm{d}x\, \mathcal{N}(x;\mu,1)^2/\mathcal{N}(x;0,1)$ leaves a factor e^{μ^2} — giving $\chi^2 = e^{\mu^2} - 1$. At $\mu = 6$, a six-standard-deviation gap that any interesting target exceeds without trying, $\chi^2 \approx 4 \times 10^{15}$.

What that number costs in samples is made precise by the *effective sample size*: for self-normalized importance sampling, N draws from π_0 carry the statistical weight of only $\mathrm{ESS} = N/(1+\chi^2)$ independent draws from the target, estimated in practice by $\widehat{\mathrm{ESS}} = (\sum_i w_i)^2 / \sum_i w_i^2$. At $\mu = 6$ and $N = 10^4$, the effective sample size is of order 10^{-12} — a trillionth of one sample. The estimator fails quantitatively, not conceptually: it is unbiased, yet unusable across any real gap.

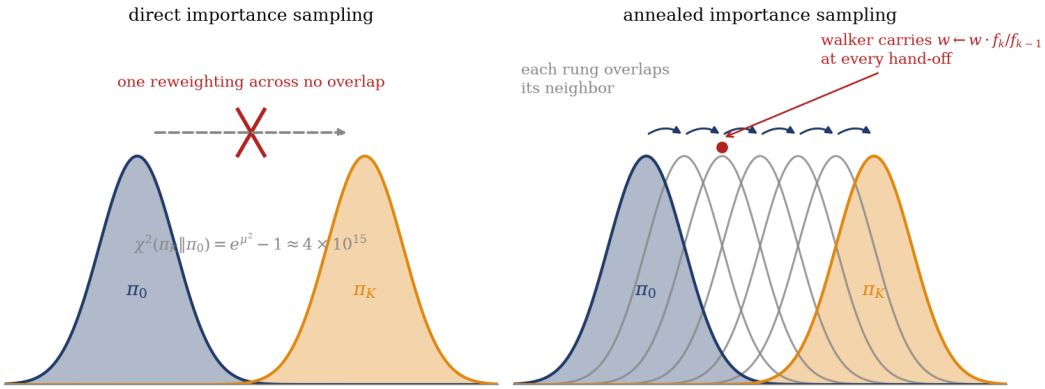

Figure 22.1: The central picture of the lecture. Left: direct importance sampling attempts a single reweighting from the reference π_0 to a target π_K six standard deviations away; the relative variance is the chi-squared divergence Equation 22.1, here $e^{36} - 1 \approx 4 \times 10^{15}$, and the estimator collapses. Right: annealed importance sampling threads the same gap with a ladder of intermediate distributions, each overlapping its neighbor; a walker alternates weight updates (at the hand-offs between rungs, where $w \leftarrow w \cdot f_k/f_{k-1}$) with MCMC moves that need no normalizers. The mean of the accumulated weight is exact for any ladder; what the ladder buys is variance.

The repair is the previous lecture's physics read backwards, and Figure 22.1 shows it before any formula does. A single switch dissipates severely; many gentle switches, each relaxed by a little dynamics, dissipate almost nothing. Thread the gap with a ladder of intermediate distributions, each overlapping its neighbor, and replace the one impossible reweighting by K easy ones. The claim that needs proof — and it

is less obvious than the picture suggests — is that interleaving reweightings with MCMC moves leaves the estimator *exactly* unbiased, for any ladder and any number of rungs.

> Take-home 1
>
> Direct importance sampling estimates Z_K/Z_0 without bias, with relative variance $\chi^2(\pi_K\|\pi_0)$ — exponentially large in any real gap between reference and target ($e^{\mu^2}-1$ for unit Gaussians a distance μ apart). Its logarithm is the previous lecture's work for an instantaneous switch, and its failure is maximal dissipation seen from the estimator's side. The repair is a ladder of intermediates: many small reweightings in place of one impossible one.

22.3 The ladder: switch, move, repeat

The intermediates come first. Between the reference f_0 and the target f_K place the *geometric path*

$$f_k(x) = f_0(x)^{1-\beta_k}\, f_K(x)^{\beta_k}, \qquad 0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1,$$

with $\pi_k = f_k/Z_k$. The *annealing schedule* β_k is a design choice, not a temperature — flag the clash once: β was an inverse temperature all course, and here it interpolates energies, $E_k = (1-\beta_k)E_0 + \beta_k E_K$. For a Boltzmann target with a flat reference the two readings coincide and the geometric path *is* the temperature ladder of Chapter 16 (Equation 16.4) — the same schedule that cooled to optimize in Week 8, now redeployed as a measuring instrument. Each π_k overlaps its neighbors by construction; the gap of Figure 22.1 has become K small steps.

The algorithm of Neal (2001) walks the ladder, alternating two elementary moves. Initialize $x_0 \sim \pi_0$ (exact, since the reference is tractable) and $w = 1$; then for $k = 1, \ldots, K$:

① **Switch.** Multiply the weight by the ratio of adjacent rungs at the *current* configuration,

$$w \;\leftarrow\; w \cdot \frac{f_k(x_{k-1})}{f_{k-1}(x_{k-1})}. \tag{22.2}$$

the AIS weight accumulates only at switches

② **Move.** Draw x_k from any Markov kernel T_k that leaves π_k *invariant* — a few Metropolis steps at rung k, say. By Equation 15.2 the kernel needs only energy differences; every Z_k cancels, exactly as in Week 8.

Return w after the last switch. Two structural remarks before the theorem. First, *where the record lives*: the kernel steps contribute nothing to w — the entire history accumulates at the switches, and in the previous lecture's dictionary the switch increment is precisely the work of moving the Hamiltonian under a frozen configuration, $\log w = -\sum_{k=1}^{K}[E_k(x_{k-1}) - E_{k-1}(x_{k-1})] = -W$. The protocol has become a schedule, the trajectory a sequence of MCMC states, and the work an accumulated log-weight; the dictionary is complete. Second, *what is never asked*: the kernels are required to leave each π_k invariant, not to converge to it. A single Metropolis step per rung is legitimate, and no stage of the algorithm equilibrates. AIS is a transport

with exact bookkeeping of its own irreversibility. The same structure appeared in Chapter 20 as the reverse-time SDE; here it takes discrete, weighted form. The next section proves the weight is unbiased.

22.4 The engine: importance sampling on whole paths

We prove that the accumulated weight is unbiased, $\langle w \rangle = Z_K/Z_0$, for any schedule and any invariant kernels. The idea, due to Neal, is to apply ordinary importance sampling not on configurations but on the *extended state space* of whole annealing paths $(x_0, x_1, \dots, x_{K-1})$ — and it is Jarzynski's trajectory-ensemble argument in discrete time.

The forward path density. The algorithm itself defines a distribution over paths: draw from the reference, then apply the kernels in order,

$$q_F(x_0, \dots, x_{K-1}) = \pi_0(x_0)\, T_1(x_0 \to x_1)\, T_2(x_1 \to x_2) \cdots T_{K-1}(x_{K-2} \to x_{K-1}).$$

The reverse path density. Define a second distribution on the *same* space by running the ladder from the far end: draw $x_{K-1} \sim \pi_K$, then walk down with the *reversed kernels*,

$$q_R(x_0, \dots, x_{K-1}) = \pi_K(x_{K-1})\, \tilde{T}_{K-1}(x_{K-1} \to x_{K-2}) \cdots \tilde{T}_1(x_1 \to x_0),$$

where $\tilde{T}_k$ is the *reversal* of T_k with respect to its invariant distribution,

$$\pi_k(x)\, T_k(x \to x') = \pi_k(x')\, \tilde{T}_k(x' \to x).$$

The reversal is exactly the balance bookkeeping of Week 8: for a detailed-balance kernel (Equation 15.1) the two sides already match with $\tilde{T}_k = T_k$, and Metropolis kernels are of this kind — but the proof needs only invariance, so we carry the general $\tilde{T}_k$ (summing the definition over x confirms that $\tilde{T}_k$ is a proper kernel precisely because $\pi_k T_k = \pi_k$).

The ratio telescopes. Now compute the importance-sampling ratio of the two path densities. Each substitution of the reversal definition trades a $\tilde{T}_k$ for a T_k at the cost of a ratio of π_k's:

$$\frac{q_R(x_0, \dots, x_{K-1})}{q_F(x_0, \dots, x_{K-1})} = \frac{\pi_K(x_{K-1})}{\pi_0(x_0)} \prod_{k=1}^{K-1} \frac{\tilde{T}_k(x_k \to x_{k-1})}{T_k(x_{k-1} \to x_k)} = \frac{\pi_K(x_{K-1})}{\pi_0(x_0)} \prod_{k=1}^{K-1} \frac{\pi_k(x_{k-1})}{\pi_k(x_k)}$$

$$= \prod_{k=1}^{K} \frac{\pi_k(x_{k-1})}{\pi_{k-1}(x_{k-1})} = \prod_{k=1}^{K} \underbrace{\frac{Z_{k-1}}{Z_k}}_{\text{telescopes to } Z_0/Z_K} \frac{f_k(x_{k-1})}{f_{k-1}(x_{k-1})} = \frac{Z_0}{Z_K}\, w,$$

where the second line regroups the same factors by configuration rather than by kernel — every $\pi_k(x_k)$ in a denominator meets the π_k evaluated one step earlier in a numerator — and the normalizers of the interior rungs cancel in adjacent pairs.

The ratio of two path measures is, pointwise, the AIS weight times a constant. Unbiasedness follows: since q_R is normalized on the extended space,

$$\langle w \rangle_{q_F} = \int \mathrm{d}x_0 \cdots \mathrm{d}x_{K-1}\; q_F \cdot \frac{Z_K}{Z_0}\frac{q_R}{q_F} = \frac{Z_K}{Z_0} \underbrace{\int \mathrm{d}x_0 \cdots \mathrm{d}x_{K-1}\; q_R}_{=1},$$

and we box the master result:

$$\boxed{\left\langle \prod_{k=1}^{K} \frac{f_k(x_{k-1})}{f_{k-1}(x_{k-1})} \right\rangle_{q_F} = \frac{Z_K}{Z_0} \quad \text{for any schedule and any invariant kernels.}} \tag{22.3}$$

annealed importance sampling is exact in expectation (Neal 2001)

In words: the AIS weight is the density ratio of the reverse path measure to the forward one, so the estimator is ordinary importance sampling promoted from states to whole annealing histories — and the χ^2 that controls its variance is now a divergence between *path* measures, which the schedule can make small even when the endpoint divergence is astronomical. One further line closes the dictionary. Substitute $f_k = e^{-E_k}$: the weight becomes $\log w = -W$, the boxed equation reads $\langle e^{-W} \rangle = e^{-\Delta F}$ (Equation 21.4, with β absorbed), and the extended state space — the previous lecture wrote its measures P_F and P_R — is Jarzynski's trajectory ensemble under a different name. The previous lecture's proof and Neal's are one proof; a physicist met it as a statement about driven systems, a statistician as a variance repair, within a year of each other.

> **Take-home 2**
>
> AIS alternates weight switches and invariant MCMC moves along a ladder from reference to target; the accumulated weight satisfies $\langle w \rangle = Z_K/Z_0$ exactly, for any schedule and any number of rungs (Equation 22.3) — proved by importance sampling on the space of whole annealing paths, which is Jarzynski's argument in discrete time. The kernels never see a normalizer, and nothing equilibrates; the identity alone carries the result.

22.5 The price: dissipation is the error bar

The expectation is exact regardless of schedule; what varies with the schedule is the variance, and the previous lecture's Crooks theorem governs it. **Jensen, and which side you are on.** The logarithm is concave, so $\langle \log w \rangle \le \log \langle w \rangle = \log(Z_K/Z_0)$: log-weights *underestimate* the log-normalizer on average, and a forward AIS run delivers a *stochastic lower bound* on $\log Z_K$. The direction matters more than it seems. An energy-based model's log-likelihood is $\log p = \log f - \log Z$, so an underestimated $\log Z$ *overestimates* the log-likelihood — AIS-evaluated models look systematically better than they are, and the bias shrinks only as the estimator's variance does.

The gap is the dissipation, and the dissipation is a KL divergence. Averaging the log of the telescoped ratio under q_F gives the size of the Jensen gap exactly: $\langle \log(q_F/q_R) \rangle_{q_F} = D_{\mathrm{KL}}(q_F \| q_R)$ on one side, and $\log(Z_K/Z_0) - \langle \log w \rangle$ on the other. With $\log w = -W$ and $\Delta F = -\log(Z_K/Z_0)$ this is

$$\boxed{\log \frac{Z_K}{Z_0} - \langle \log w \rangle = \langle W \rangle - \Delta F = \langle W_{\text{diss}} \rangle = D_{\text{KL}}(q_F \,\|\, q_R)} \tag{22.4}$$

bias in the log, dissipated work, and path-measure asymmetry are one number

— the previous lecture's dissipation identity (Equation 21.5), rederived on the ladder. The estimator's difficulty is a thermodynamic quantity: how irreversibly the ladder is traversed, measured as the divergence between the forward transport and its reversal. A reversible schedule ($q_F = q_R$) has zero gap and zero excess variance; everything the estimator suffers, it suffers in proportion to its own dissipation.

Variance, and the rare-trajectory warning made operational. The previous lecture located the exponential average in the rare low-work tail (Section 21.4); for the estimator this is not a curiosity but the failure mode. The variance of w grows exponentially with the dissipation (the one-jump limit Equation 22.1 is the extreme case), and a finite sample that has not yet visited the tail does not merely converge slowly — it produces a *stable-looking, wrong* answer, because the running mean sits quietly below the truth until the next tail event jerks it upward. Monitor the effective sample size $(\sum w_i)^2 / \sum w_i^2$, never the running mean; and note that at severe dissipation even $\widehat{\text{ESS}}$ flatters, since the weights that would expose the degeneracy are exactly the ones not yet sampled. The worked example below shows both pathologies on a target whose answer is known.

How fast does slowing down help? For a smooth schedule the dissipation obeys the near-equilibrium law $\langle W_{\text{diss}} \rangle \propto 1/K$: halving the per-rung jump quarters the per-rung dissipation (χ^2 per rung is quadratic in the jump) while doubling the rung count, for a net factor of one half. On the Gaussian ladder of the example this is elementary to verify — with well-mixed kernels the increment at rung k is $\Delta\beta\,\mu(x - \mu/2)$ with $x \sim \pi_{k-1}$, so each rung contributes variance $\mu^2\Delta\beta^2 = \mu^2/K^2$ and the total is $\text{Var}[\log w] \approx \mu^2/K$. And near equilibrium the work distribution is close to Gaussian, for which the Jarzynski equality itself fixes the relation between the two moments: $e^{-\Delta F} = \langle e^{-W} \rangle = e^{-\langle W \rangle + \text{Var}[W]/2}$, hence $\langle W_{\text{diss}} \rangle = \text{Var}[W]/2$ — the fluctuation–dissipation relation for work, and a self-consistency check the example will display. Rungs are cheap insurance: the six-sigma gap that in one jump would need $\sim 10^{15}$ raw draws to buy a single effective sample needs only $K \gtrsim \mu^2$ rungs to bring $\text{Var}[\log w]$ to order one.

Further refinements belong in the record, each at a paragraph. *The deterministic limit:* as $K \to \infty$ with perfect kernels, the sum of switch increments becomes $\log(Z_K/Z_0) = \int_0^1 \mathrm{d}\beta \, \langle \partial_\beta \log f_\beta \rangle_{\pi_\beta}$ — *thermodynamic integration*, the quasistatic method Week 8 named alongside AIS (Section 15.4); AIS interpolates between free-energy perturbation ($K = 1$) and this limit, spending variance to save equilibration. *Bidirectional estimators:* running the ladder downward from the target gives a reverse weight whose Jensen bias has the opposite sign, so the pair brackets $\log Z_K$ in a stochastic sandwich; combining both directions optimally is Bennett's acceptance-ratio method (Bennett 1976), and reading ΔF off the crossing of the forward and reverse work histograms is the previous lecture's Crooks crossing, now as an estimator. When the answer matters, run both directions. *Populations:* running many walkers in parallel and occasionally resampling them by weight — killing the light, cloning the heavy — is *sequential Monte Carlo*, the population variant of AIS; we name it and move on.

The open problem of Week 3 is resolved in practice, not only in principle. Salakhutdinov and Murray (2008) ran exactly this machinery — a temperature ladder from

the independent-unit RBM to the trained one, Gibbs transitions as the kernels — to put numbers on RBM partition functions and hence on the log-likelihoods that Section 6.4 declared unreachable; AIS has been the standard instrument for evaluating energy-based models since. Their caution is the Jensen paragraph above in the field's own experience: the reported log-likelihoods are optimistic in expectation, and the direction of the bound belongs alongside the reported number.

Trap

The standard misreadings of an AIS run. ① *Unbiased in Z is biased low in $\log Z$.* The mean of w is exact; the mean of $\log w$ undershoots by exactly the dissipation (Equation 22.4) — and for model evaluation the sign flips into flattery, since a low $\log Z$ inflates the log-likelihood. Report the direction of the bound with the number. ② *A stable running mean is not convergence.* The exponential average is carried by rare low-work paths; before the tail is sampled the estimate looks settled and is wrong, and at severe dissipation the ESS estimate is fooled with it. ③ *The weight accumulates only at switches, and invariance is the only requirement.* Kernel steps are weight-neutral (their Z_k cancels, Equation 15.2), and no rung equilibrates — accounting work into the MCMC moves, or demanding convergence at each rung, are the standard derivation errors. AIS is transport with exact bookkeeping, not a sequence of equilibrations.

Take-home 3

The Jensen gap $\log(Z_K/Z_0) - \langle \log w \rangle$ equals the mean dissipated work, equals $D_{\mathrm{KL}}(q_F \| q_R)$: bias in the log, variance, and thermodynamic irreversibility are one number (Equation 22.4). Forward AIS lower-bounds $\log Z$ stochastically; slowing the schedule shrinks the gap as $1/K$; near equilibrium $\langle W_{\mathrm{diss}} \rangle = \mathrm{Var}[W]/2$. Budget rungs by dissipation, monitor the ESS rather than the running mean, and when the answer matters, run both directions.

22.6 Optimizing the reversal

We now look back at the engine's proof with a designer's eye. Unbiasedness needed very little from the reverse path density: q_R had to start at the target and be normalized — nothing more. The reversed kernels $\tilde{T}_k$, built from the rungs' invariant distributions, were Neal's choice, not a requirement of the argument, and it is worth asking what the best choice would have been. The answer turns out to be Bayes' rule, the proof is a one-line application of Week 10's backward kernel, and the consequence — one week early — is the training objective of a diffusion model (Welling, Lu, and Holdijk 2026, ch. 12 on sequential importance sampling).

Unbiasedness survives any reversal. Replace the $\tilde{T}_k$ by arbitrary backward kernels K_k, normalized in their output ($\int \mathrm{d}x_{k-1}\, K_k(x_k \to x_{k-1}) = 1$), and define

$$q_R^{(K)}(x_0, \dots, x_{K-1}) = \pi_K(x_{K-1}) \prod_{k=1}^{K-1} K_k(x_k \to x_{k-1}), \qquad w = \frac{f_K(x_{K-1})}{f_0(x_0)} \prod_{k=1}^{K-1} \frac{K_k(x_k \to x_{k-1})}{T_k(x_{k-1} \to x_k)},$$

still computable on the fly — unnormalized densities and kernel evaluations only. Then $w = (Z_K/Z_0)\, q_R^{(K)}/q_F$ pointwise, so the one-line argument of Section 22.4 gives $\langle w \rangle_{q_F} = Z_K/Z_0$ unchanged, and the Jensen bookkeeping of Section 22.5 gives $\langle \log w \rangle = \log(Z_K/Z_0) - D_{\mathrm{KL}}(q_F \| q_R^{(K)})$: the dissipation identity Equation 22.4 holds verbatim for the general reversal. (Substituting $K_k = \tilde{T}_k$ collapses the kernel ratios through the reversal definition and recovers Equation 22.2, as it must.) Exactness is indifferent to how we reverse; the variance is not, and the reversal has become a design variable.

The optimal reversal is Bayes' rule. Week 10 established that a forward Markov chain factorizes exactly through its own Bayes kernels (Equation 20.3). Applied to the annealing walker: write q_k for the marginal of the forward process after rung k — with $q_0 = \pi_0$, and $q_k \neq \pi_k$ in general, since the unequilibrated walker *lags* the ladder — and let $B_k(x_k \to x_{k-1}) = T_k(x_{k-1} \to x_k)\, q_{k-1}(x_{k-1})/q_k(x_k)$ be the true inverse of the transport actually run. Then $q_F = q_{K-1}(x_{K-1}) \prod_k B_k(x_k \to x_{k-1})$, by the same two-line telescoping as in Week 10, and dividing this factorization by $q_R^{(K)}$ splits the dissipation rung by rung:

$$D_{\mathrm{KL}}(q_F \,\|\, q_R^{(K)}) = D_{\mathrm{KL}}(q_{K-1} \,\|\, \pi_K) + \sum_{k=1}^{K-1} \Big\langle D_{\mathrm{KL}}(B_k(x_k \to \cdot) \,\|\, K_k(x_k \to \cdot)) \Big\rangle_{q_k} . \tag{22.5}$$

the dissipation, resolved rung by rung

The first term is the *lag* — how far the transported reference q_{K-1} ends from the target — and does not involve the K_k at all. The second is a sum of nonnegative per-rung mismatches between the chosen reversal and the true inverse, so the minimum is immediate: set $K_k = B_k$ and every summand vanishes. The variance-optimal reversal is Bayes' rule applied to the *nonequilibrium marginals* of the transport, not the adjoint built on the rungs' equilibria. At the optimum, moreover, the weight telescopes through the q_k and the entire interior of the path cancels, $w = f_K(x_{K-1})/(Z_0\, q_{K-1}(x_{K-1}))$: what remains is one-jump importance sampling (Equation 22.1) from q_{K-1} — the reference *after* it has been dragged across the gap — rather than from π_0 across all of it. The exact denoiser repairs every rung's dissipation in hindsight, and only the lag is left standing.

Why Neal's algorithm does not use it. The kernel B_k requires the marginals q_k — the density of a driven, unequilibrated walker, an integral over its entire history, and precisely as intractable as the problem being solved. The adjoint $\tilde{T}_k$ needs only ratios of the f_k. AIS is therefore a deliberate trade — an evaluable reversal in place of the optimal one — and Equation 22.5 prices the trade exactly, at the per-rung mismatch between B_k and $\tilde{T}_k$. Week 10 also says when that price vanishes: the backward kernel coincides with the adjoint precisely at stationarity, so $B_k \to \tilde{T}_k$ as $q_k \to \pi_k$ — the adiabatic regime, in which the kernels keep the walker equilibrated on every rung. Neal's choice is the optimal reversal's equilibrium approximation, and it improves exactly as fast as the schedule slows.

The third option is Week 12's. Between the evaluable adjoint and the intractable Bayes inverse runs a middle road: *learn* the reversal. Parameterize the backward kernels and adjust them to minimize $D_{\mathrm{KL}}(q_F \| q_R)$ — by Equation 22.5 a regression of each K_k toward the true denoiser B_k, and by the generalized dissipation identity the same operation as maximizing the stochastic lower bound $\langle \log w \rangle$ on $\log(Z_K/Z_0)$: an evidence lower bound in the sense of Chapter 14, promoted from configurations to paths. That is a complete specification of the diffusion-model training problem,

stated a week before the course builds one — fix a forward corruption, parameterize its reversal, minimize a forward–reverse path divergence — and it explains in advance why the object Week 12's network regresses toward is a conditional Bayes kernel. What Week 12 adds is the Gaussian machinery that makes the regression tractable.

22.7 Example: a six-sigma gap, one jump versus sixty-four rungs

We now put numbers on the mechanism, on a target where every claim is checkable. Take $\pi_0 = \mathcal{N}(0, 1)$ and $\pi_K = \mathcal{N}(6, 1)$ — both endpoints normalized Gaussians, so the true answer is $\log(Z_K/Z_0) = 0$ by construction and every error is visible. The geometric path between them is the mean-shift ladder $\mathcal{N}(6\beta_k, 1)$ (up to a known factor absorbed in the f_k), the schedule is uniform, $\beta_k = k/K$, and the kernels are twenty Metropolis steps of width one per rung. We run $N = 5000$ walkers for $K \in \{1, 2, 4, \dots, 64\}$, with $K = 1$ reproducing direct importance sampling across the full gap.

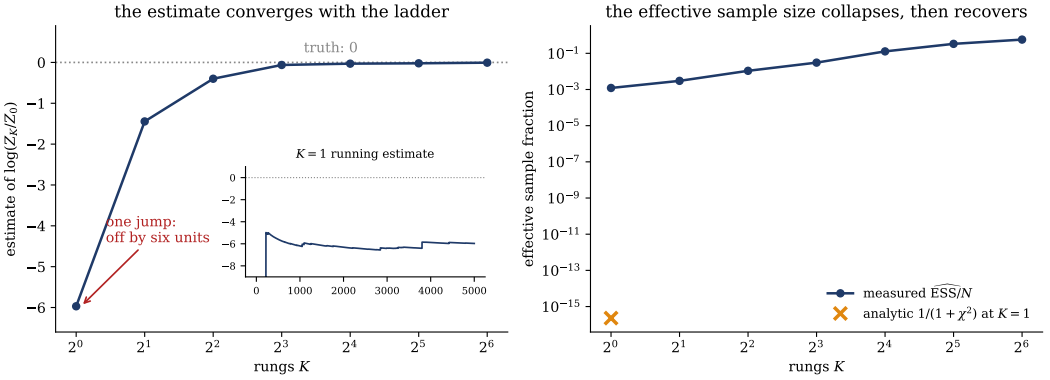

Figure 22.2: One jump versus the ladder, on the six-sigma Gaussian gap ($\pi_0 = \mathcal{N}(0, 1)$, $\pi_K = \mathcal{N}(6, 1)$, true $\log(Z_K/Z_0) = 0$; $N = 5000$ walkers, 20 Metropolis steps of width 1 per rung, fixed seed). Left: the AIS estimate of $\log(Z_K/Z_0)$ against the rung count K. Direct importance sampling ($K = 1$) sits at -5.97, six units below the truth; by $K = 64$ the estimate is -0.006. The inset shows the $K = 1$ running estimate against sample count: it drifts through -6.2 at $N = 1000$ and -6.0 at $N = 5000$, settled-looking throughout and wrong by six units — the tail that carries the mean has not been sampled, and nothing in the trace says so. Right: the measured effective-sample-size fraction $\widehat{\mathrm{ESS}}/N$ (navy); at $K = 1$ it reports about 6 usable walkers of 5000, while the analytic value $1/(1 + \chi^2) = 2.3 \times 10^{-16}$ (orange cross) shows the estimator for the diagnostic is itself fooled by thirteen orders of magnitude.

The left panel of Figure 22.2 shows the mechanism numerically. One jump lands at -5.97 against a true value of 0, and its running mean (inset) would pass any casual convergence check while doing so — the second pathology of Section 22.5, demonstrated explicitly. The ladder walks the estimate onto the truth: -0.03 by $K = 16$ and -0.006 at $K = 64$, at a total cost of 64×20 kernel steps per walker. The right panel prices the same story in effective samples, and adds the sharper warning: at $K = 1$ the *measured* ESS reports six usable walkers where the analytic

answer is 10^{-12} of one — the diagnostic collapses exactly when it is needed most, because the weights that would expose the degeneracy live in the unsampled tail.

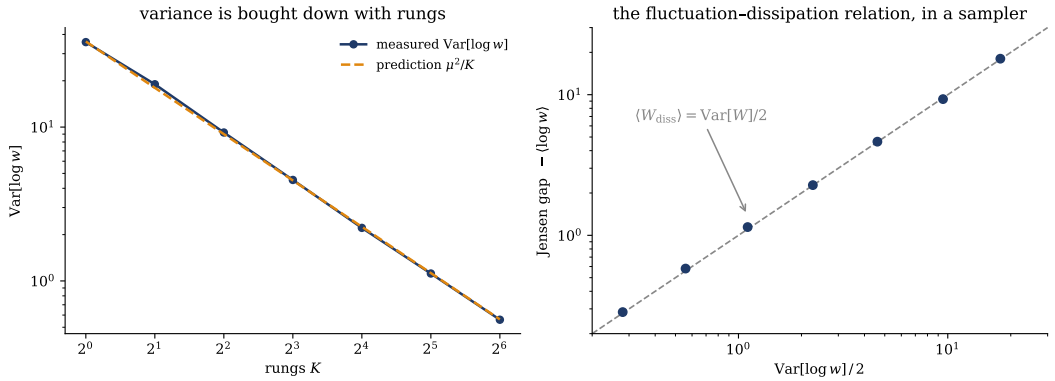

Figure 22.3: Dissipation under schedule control, same runs as Figure 22.2. Left: the variance of the log-weight falls as the predicted $\mu^2/K = 36/K$ (dashed) once the ladder resolves the gap — measured 0.56 at $K = 64$ against the predicted 0.5625. Right: the measured Jensen gap $-\langle \log w \rangle$ against half the log-weight variance, one point per K; the points collapse onto the diagonal (0.285 versus 0.280 at $K = 64$), which is the work fluctuation–dissipation relation $\langle W_{\text{diss}} \rangle = \text{Var}[W]/2$ observed inside a sampler.

Figure 22.3 closes the loop on the theory of Section 22.5. The log-weight variance tracks μ^2/K once the rungs resolve the gap (0.56 measured against 0.5625 predicted at $K = 64$), so the six-sigma gap that defeated one jump is tamed by $K \approx \mu^2$ gentle ones, as the near-equilibrium law promised. And the right panel is a piece of statistical mechanics observed inside an algorithm: the Jensen gap — the bias of the log-estimate, per Equation 22.4 the mean dissipated work — equals half the work variance, point by point along the schedule. The fluctuation–dissipation relation has become a convergence diagnostic.

> *Averages of W obey an inequality, but averages of e^{-W} obey an equality — and an equality is an estimator: the Z this course could never compute is the mean of an annealing weight.*

The tutorial (16–18, Ph12 106) is computation-led, and the machinery is now complete: the weight recursion Equation 22.2, the unbiasedness result Equation 22.3, and the dissipation accounting Equation 22.4. The hard problem computes a free-energy difference by AIS on a target whose answer you do *not* know by symmetry, and verifies the Jarzynski equality numerically along the way; today's example showed the failure mode and the scaling so that the tutorial can watch for both — the ESS estimator and the Jensen gap are the relevant diagnostics. The estimator is run by the groups, one machine each, on a prediction committed beforehand.

22.8 Outlook: from measuring transports to learning them

Week 11 completes the course's treatment of the partition function. The spine has been Z since Week 1: dodged by Hopfield's dynamics, bounded by mean field, cancelled by Metropolis, removed from training and generation by the score. Each of those was a way of *not needing* the number. This week the course computed it

— from unequilibrated, finite-time, deliberately irreversible runs, with the previous lecture's equality guaranteeing the mean and today's dissipation accounting pricing the fluctuations. The three open problems from the introduction are resolved: RBM likelihoods (the instrument of Salakhutdinov and Murray (2008)), free-energy differences, and model comparison all reduce to Z-ratios, and Z-ratios are means of annealing weights.

The last step of the arc is no longer a measurement but a design problem, and Section 22.6 has already posed it. A diffusion model is a *learned* annealing path: the forward corruption of Section 20.2 is the ladder, the reverse-time SDE of Equation 20.2 is the transport, and the training loss that makes the reversal learnable is exactly the program that section wrote down — a Kullback–Leibler divergence between a forward and a reverse path measure, a dissipation minimized by gradient descent over the reversal. The lineage is the historical record: Sohl-Dickstein et al. (2015) built the first diffusion models directly from this literature, and their title — *Deep Unsupervised Learning using Nonequilibrium Thermodynamics* — is a citation of this week. Week 12 assembles the industrial object, DDPM and the score-SDE (Ho, Jain, and Abbeel 2020; Song, Sohl-Dickstein, et al. 2021), the previous lecture quantified the cost of nonequilibrium transport, today that cost became an estimator, and next week a neural network learns the transport that minimizes it.

23 DDPM: the estimator becomes a training loss

Recommended reading

Core (~1 h). Ho, Jain, and Abbeel (2020) — §2–3, read this time as a derivation rather than a recipe: you have already trained this model, and today every line of its loss acquires a name from the last three weeks. Alongside it Sohl-Dickstein et al. (2015), §1–2 — the founding construction, which cites Jarzynski (1997) explicitly; its title, *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*, is a one-line summary of this module.
Optional background. Song, Durkan, et al. (2021) — which reweighting of the denoising loss restores an honest likelihood bound; today's engineering section states the result. Karras et al. (2022) — the modern audit of the design space (schedules, scalings, samplers) once the construction is understood. Welling, Lu, and Holdijk (2026) — the companion text's variational-diffusion chapters treat exactly this material.
Prerequisite reminder. From Week 10: denoising score matching (Equation 19.4), the forward noising process (Equation 20.1), the reverse-time SDE (Equation 20.2), and the transport-not-equilibration principle of Chapter 20. From Week 11: the Crooks ratio (Equation 21.3), dissipation as a forward/reverse path divergence (Equation 21.5), the AIS identity (Equation 22.3), and the Jensen-gap accounting (Equation 22.4). From Module 2: the ELBO as a variational free energy (Equation 14.6) and the closed-form Gaussian KL (Equation 14.4). New content: the discrete forward chain and its two-anchor posterior, the path-space bound and its per-step decomposition, the ε-parameterization and its identity with the score, and the audit of DDPM's engineering choices.

23.1 A loss you have already minimized

In *Machine Learning and Physics* you trained a denoising diffusion probabilistic model, and the loss you minimized was a mean-squared error on noise: draw a data point, corrupt it to a random level t, and train a network ε_θ to guess the noise that was added,

$$L_{\text{simple}}(\theta) \;=\; \Big\langle \big\| \varepsilon - \varepsilon_\theta\big(\sqrt{\bar\alpha_t}\, x_0 + \sqrt{1-\bar\alpha_t}\, \varepsilon,\; t\big) \big\|^2 \Big\rangle_{x_0,\,\varepsilon,\,t}.$$

Nothing in that expression looks like statistical mechanics. There is no partition function, no free energy, no temperature; it reads as a regression, and it was presented to you as one. L_{simple} is the terminal point of a derivation that runs through everything Module 4 has built: it is a *nonequilibrium free-energy estimator* in the

precise sense of Week 11 — a Jensen-gap bound whose slack is the dissipation of a transport, with the reverse path measure now carrying the parameters — and each of its terms is Week 10's denoising score matching at one noise level. Ho, Jain, and Abbeel (2020) wrote the recipe; Sohl-Dickstein et al. (2015) had built the construction five years earlier, directly from the fluctuation-theorem literature. Today we run the derivation forward, from the corruption chain to the regression, and keep a ledger of every step at which an *engineering decision* — a choice not forced by the mathematics — enters. There are four, catalogued in Section 23.5.

The required tools are all in hand — the forward process and its melting barriers (Section 20.2), the reverse-time SDE (Equation 20.2), denoising score matching (Equation 19.4), and the path-measure bookkeeping of Week 11's two lectures. The new content is how they assemble into an industrial object, and the discrete-time bookkeeping that assembly requires.

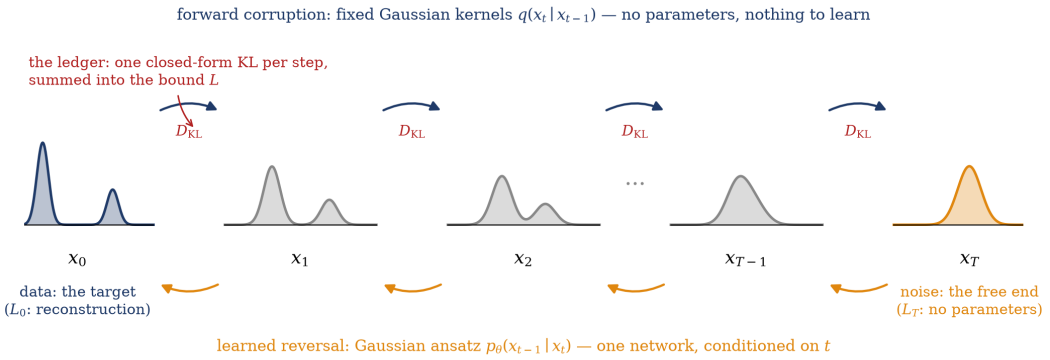

Figure 23.1: The central picture of the lecture — the DDPM as a double chain with a ledger. Top (navy): the forward corruption, a fixed Markov chain of Gaussian kernels $q(x_t \mid x_{t-1})$ carrying the two-mode data density through blurred intermediates to a unit Gaussian; it contains no parameters and is never trained. Bottom (orange): the learned reversal, a Gaussian ansatz $p_\theta(x_{t-1} \mid x_t)$ — one network, conditioned on t — run from noise back to data. Between the two chains sits the ledger: the variational bound decomposes into one closed-form Kullback–Leibler divergence per step, plus a parameter-free endpoint term L_T (does the corruption finish?) and a reconstruction term L_0. Training a diffusion model is bookkeeping on this ledger.

23.2 The forward chain: the ladder, discretized

The DDPM forward process is Week 10's corruption (Equation 20.1) written as a finite Markov chain. Fix a *schedule* $\beta_1, \dots, \beta_T \in (0,1)$ and corrupt in T steps,

$$q(x_t \mid x_{t-1}) \;=\; \mathcal{N}\!\left(x_t;\ \sqrt{1-\beta_t}\, x_{t-1},\ \beta_t I\right), \qquad t = 1, \dots, T.$$

A further notation warning: β_t here is a *variance schedule* — the fraction of variance replaced by fresh noise at step t — not an inverse temperature, exactly as the β_k of the AIS ladder (Section 22.3) was an annealing schedule. The field's notation is entrenched and we keep it; no temperature appears anywhere today.

Write $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Composing the kernels collapses the chain into a single Gaussian: if $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\, \varepsilon'$, then one more step gives fresh noise of variance $\alpha_t(1-\bar{\alpha}_{t-1}) + \beta_t = 1 - \bar{\alpha}_t$, and by induction

$$q(x_t \mid x_0) \;=\; \mathcal{N}(x_t;\; \sqrt{\bar{\alpha}_t}\, x_0,\; (1-\bar{\alpha}_t)\, I), \qquad \text{i.e.} \qquad x_t \;=\; \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{23.1}$$

the marginal: signal decays, total variance is preserved

The construction is *variance-preserving* — a unit-variance input stays at unit variance, $\alpha_t + \beta_t = 1$ — and it is precisely the Ornstein–Uhlenbeck corruption of Equation 20.1 sampled at discrete times: setting $\sqrt{\bar{\alpha}_t} = e^{-t_{\text{phys}}}$ reproduces Week 10's transition law exactly, with $\beta_t \approx 2\,\Delta t$ for small steps. Everything established there carries over unchanged — the endpoint is free ($\bar{\alpha}_T \to 0$ makes x_T pure noise, whatever the data were), the intermediates are blurs whose barriers have melted (Figure 20.2), and the intermediates are exactly where denoising score matching trains.

The chain adds a closed form that the continuous picture did not need. Because the chain is Markov and every factor is Gaussian, the corruption step is exactly invertible *when anchored at both ends*: by Bayes,

$$q(x_{t-1} \mid x_t, x_0) \;=\; \frac{q(x_t \mid x_{t-1})\; q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \;=\; \mathcal{N}(x_{t-1};\; \tilde{\mu}_t(x_t, x_0),\; \tilde{\beta}_t I), \tag{23.2}$$

the two-anchor posterior: the reversal, if you knew the answer

with — completing the square in x_{t-1}, whose precision collects α_t/β_t from the likelihood and $1/(1-\bar{\alpha}_{t-1})$ from the prior —

$$\tilde{\mu}_t(x_t, x_0) \;=\; \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})\, x_t + \sqrt{\bar{\alpha}_{t-1}}\,\beta_t\, x_0}{1-\bar{\alpha}_t}, \qquad \tilde{\beta}_t \;=\; \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t.$$

Equation 23.2 quantifies the challenge facing generation. Conditioned on where the trajectory started, one corruption step reverses in closed form — a Gaussian, with a mean that interpolates between the two anchors. But the starting point x_0 is exactly what a generator does not have; it is the thing being generated. The model's entire job will be to supply a substitute for the x_0-dependence of $\tilde{\mu}_t$, and the substitute will turn out to be Week 10's score.

23.3 The bound: dissipation acquires parameters

The generative model is the reverse chain of Figure 23.1: start at the free end and walk down,

$$p_\theta(x_{0:T}) \;=\; p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \qquad p(x_T) = \mathcal{N}(0, I),$$

with kernels to be specified in the next section. Its likelihood is a path integral, $p_\theta(x_0) = \int \mathrm{d}x_{1:T}\, p_\theta(x_{0:T})$, and we evaluate it the way Week 11 evaluated every partition function: importance-sample the paths with the one path measure we can draw from — the forward corruption —

$$p_\theta(x_0) \;=\; \Big\langle \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \Big\rangle_{q(x_{1:T}|x_0)}.$$

This display is Equation 22.3 with the roles reassigned: the forward corruption is the annealing protocol, the ratio is the AIS weight w, and the quantity in the role of Z_K/Z_0 is the model likelihood itself. In last week's Lecture 1 dictionary $\log w$ is a negative work, and Jensen's inequality plays the part it played there (Equation 22.4): $\log p_\theta(x_0) = \log\langle w\rangle \geq \langle \log w\rangle$, with the gap known exactly. Writing $L(x_0;\theta) = -\langle \log w\rangle$ for the negative evidence lower bound and splitting $p_\theta(x_{0:T}) = p_\theta(x_0)\, p_\theta(x_{1:T} \mid x_0)$,

$$\boxed{L(x_0;\theta) \;=\; \Big\langle \log \frac{q(x_{1:T} \mid x_0)}{p_\theta(x_{0:T})} \Big\rangle_q \;=\; -\log p_\theta(x_0) \;+\; D_{\mathrm{KL}}(q(x_{1:T} \mid x_0) \,\|\, p_\theta(x_{1:T} \mid x_0))} \tag{23.3}$$

the DDPM objective: a dissipation with parameters in the reverse measure

— the negative log-likelihood plus a Kullback–Leibler divergence between the forward path measure and the learned reverse one. The structure is Equation 21.5: *the slack in the DDPM training objective is the dissipation of its own transport.* In Week 11 the reverse path measure was fixed — the reversal of the forward kernels — and the dissipation was a cost to be accounted; here the reverse measure carries θ, and the dissipation is a cost to be *minimized by gradient descent.* Training pushes $D_{\mathrm{KL}}(q_F \| p_\theta) \to 0$, that is, it drives the learned chain toward the exact time reversal of the corruption — and Anderson's theorem (Equation 20.2) guarantees the target is reachable, since the exact reversal of a diffusion is again a diffusion, with Gaussian increments in the small-step limit. A model at zero dissipation is a reversible transport, and its bound is tight: $L = -\log p_\theta(x_0)$.

Two remarks place the object among its relatives. First, in Week 7's language Equation 23.3 is an ELBO (Equation 14.6): a diffusion model is a variational autoencoder whose latent variable is the entire path $x_{1:T}$ and whose encoder is *fixed* — the corruption chain contains nothing to learn, which is why the encoder side contributes no gradients, and neither the reparameterization trick nor an amortization network appears anywhere. Second, the historical order ran the other way from the pedagogical one: Sohl-Dickstein et al. (2015) constructed exactly this bound by importing the Jarzynski–Crooks machinery into unsupervised learning, and Ho, Jain, and Abbeel (2020) made it competitive at scale five years later; the fluctuation-theorem lineage is a matter of citation record rather than a reinterpretation.

The bound as written is a single number per data point; training needs it as a sum of local terms. The move is the telescoping of Section 22.4, executed once more. For $t \geq 2$, trade the forward kernel for the two-anchor posterior using Markovianity and Bayes (Equation 23.2),

$$q(x_t \mid x_{t-1}) \;=\; q(x_t \mid x_{t-1}, x_0) \;=\; q(x_{t-1} \mid x_t, x_0)\, \frac{q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)},$$

and the marginal ratios telescope across the product $\prod_{t=2}^{T}$, leaving $q(x_T \mid x_0)$ at the far end — the same regrouping-by-configuration that proved Equation 21.3 and Equation 22.3. The bound falls apart into a ledger:

$$L \;=\; \underbrace{D_{\mathrm{KL}}(q(x_T \mid x_0)\,\|\,p(x_T))}_{L_T} \;+\; \sum_{t=2}^{T} \underbrace{\Big\langle D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0)\,\|\,p_\theta(x_{t-1} \mid x_t))\Big\rangle_q}_{L_{t-1}} \;-\; \underbrace{\big\langle \log p_\theta(x_0 \mid x_1)\big\rangle_q}_{-L_0}. \tag{23.4}$$

the ledger: one KL per step, two endpoint terms

Each entry has a job. L_T contains no parameters at all: it measures whether the corruption finishes — whether $q(x_T \mid x_0)$ has actually reached the unit Gaussian the reverse chain starts from — and with $\bar{\alpha}_T \sim e^{-8}$ it is a fraction of a nat, fixed by the schedule. L_0 is a reconstruction term for the final hop to data; for image models Ho, Jain, and Abbeel (2020) handle it with a discrete decoder over pixel bins, and we set it aside at a sentence. The sum in the middle is the substance: one Kullback–Leibler divergence per rung, between the two-anchor posterior — the reversal *if you knew the answer* — and the model's kernel, which must manage without. Every distribution in sight is Gaussian, so every entry will be closed-form; no expectation in Equation 23.4 ever requires more than drawing (x_0, ε, t) and evaluating a formula. The partition-function obstruction is absent: the model was built normalized, kernel by kernel.

> Take-home 1
>
> A DDPM is a fixed corruption chain plus a learned reverse chain, and its training objective Equation 23.3 is the negative log-likelihood plus the forward/reverse path divergence of Equation 21.5 — Week 11's dissipation, with the reverse measure now the trainable object. Training minimizes the dissipation of the model's own transport; the AIS telescoping decomposes the bound into one closed-form Gaussian KL per step (Equation 23.4).

23.4 From ledger to noise prediction

Now the kernels. The ansatz of Ho, Jain, and Abbeel (2020) is Gaussian with a fixed covariance,

$$p_\theta(x_{t-1} \mid x_t) \;=\; \mathcal{N}(x_{t-1};\; \mu_\theta(x_t, t),\; \sigma_t^2 I),$$

σ_t^2 set by hand — the first engineering decision, examined with the others in the next section. The choice buys an immediate collapse: the KL between Gaussians of equal covariance (Equation 14.4, with the trace and log-determinant terms now θ-independent) reduces each ledger entry to a squared distance between means,

$$L_{t-1} \;=\; \frac{1}{2\sigma_t^2}\Big\langle \big\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\big\|^2 \Big\rangle_q \;+\; \text{const.}$$

The network's task is to hit the posterior mean, and the efficient parameterization emerges from writing that target in the right variable. Solve Equation 23.1 for the data point, $x_0 = (x_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon)/\sqrt{\bar{\alpha}_t}$, and substitute into $\tilde{\mu}_t$; the algebra runs through the same identity $\alpha_t(1-\bar{\alpha}_{t-1}) + \beta_t = 1 - \bar{\alpha}_t$ that built the marginal, and leaves

$$\tilde{\mu}_t \;=\; \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon\Big)\,:$$

the posterior mean is the current state, nudged against the *noise that was actually added*. Give the model the same functional form with a learned noise estimate, $\mu_\theta = (x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(x_t,t))/\sqrt{\alpha_t}$, and the means difference collapses onto the noise difference:

$$\boxed{L_{t-1} \;=\; \frac{\beta_t^2}{2\sigma_t^2\,\alpha_t(1-\bar{\alpha}_t)}\,\Big\langle \big\|\varepsilon - \varepsilon_\theta(x_t,t)\big\|^2 \Big\rangle_{x_0,\varepsilon}, \qquad s_\theta(x_t,t) \;=\; -\,\frac{\varepsilon_\theta(x_t,t)}{\sqrt{1-\bar{\alpha}_t}}} \tag{23.5}$$

each ledger entry is denoising score matching; ε-prediction is the score in disguise

The left half of the box is the loss you trained, one weight away (L_{simple} drops the prefactor — the second engineering decision). The right half is the identity that closes the module's circle, and it is not an analogy but Week 10 verbatim. The conditional score of the marginal Equation 23.1 is $\nabla_{x_t}\log q(x_t \mid x_0) = -(x_t - \sqrt{\bar{\alpha}_t}\,x_0)/(1-\bar{\alpha}_t) = -\varepsilon/\sqrt{1-\bar{\alpha}_t}$, so the ε-regression in Equation 23.5 *is* denoising score matching (Equation 19.4) at noise scale $\sigma^2 = 1-\bar{\alpha}_t$, rescaled: by Vincent's identity its minimizer is $\varepsilon^*_\theta(x_t,t) = \langle \varepsilon \mid x_t\rangle = -\sqrt{1-\bar{\alpha}_t}\;\nabla_{x_t}\log q_t(x_t)$, the score of the blurred marginal in different units. The network you trained to guess noise was estimating $\nabla \log p_t$ all along, and the whole objective is the syllabus sentence made precise: *denoising score matching at every noise level, tied together by the variational bound.* Figure 23.2 verifies the identity with no training in the loop — the conditional mean of the noise, estimated by binned averages from raw corruption samples, lands on the rescaled analytic score of the running two-mode target.

> **Take-home 2**
>
> With a fixed-covariance Gaussian ansatz for the reverse kernels, each ledger entry collapses to a squared error between posterior and model means, and in the noise variable to the weighted regression Equation 23.5. Its minimizer is $\langle \varepsilon \mid x_t\rangle = -\sqrt{1-\bar{\alpha}_t}\,\nabla \log p_t$: the noise-prediction network is a score model in different units, and the DDPM objective is denoising score matching at every rung, held together by the variational bound.

23.5 The engineering ledger

Every remaining feature of the recipe is an engineering decision layered on top of the exact construction. Four entries follow, in the order they appeared.

① **Fixed reverse variances.** Ho, Jain, and Abbeel (2020) set σ_t^2 to β_t or to $\tilde{\beta}_t$ and report similar results for both. The two are the ends of a bracket — $\tilde{\beta}_t$ is the posterior variance when x_0 is deterministic, β_t the correct value when p_0 is itself a unit Gaussian — and for the schedules in use they differ by little except at the smallest t. Learning the variances (interpolating inside the bracket) recovers a few hundredths of a nat of likelihood (Nichol and Dhariwal 2021); the mean is where the physics is.

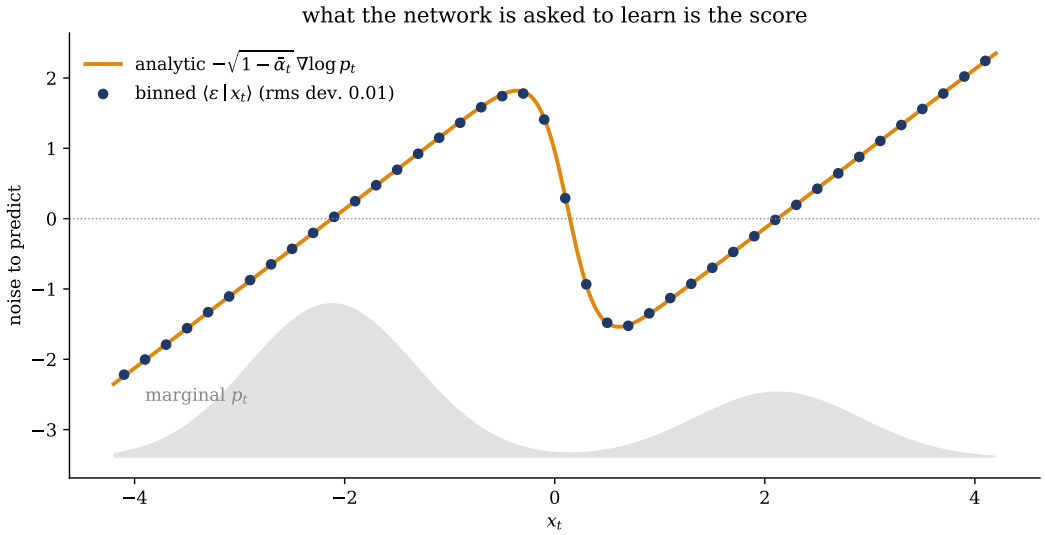

Figure 23.2: The ε-regression target is the rescaled score, verified without training. At noise level $\bar{\alpha}_t = 0.5$, corruption samples $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon$ of the running two-mode target are binned in x_t and the added noise ε is averaged within each bin (navy points, 4×10^5 samples, bins with at least 500 hits). The binned conditional mean $\langle \varepsilon \mid x_t \rangle$ — the minimizer of the DDPM regression Equation 23.5 — lands on the analytic curve $-\sqrt{1-\bar{\alpha}_t}\, \nabla \log p_t$ (orange) to an rms deviation of 0.01. The gray band underneath is the marginal p_t: where the blur has mass the estimate is tight, and the sparse bins near the density minimum scatter visibly — the same coverage effect that Week 10 flagged for learned scores.

② **Unit weights.** The full bound Equation 23.5 carries the prefactor $w_t = \beta_t^2/(2\sigma_t^2\alpha_t(1-\bar{\alpha}_t))$; L_{simple} replaces every w_t by one. Figure 23.3 computes the true weights for a $T = 1000$ chain: they span more than two orders of magnitude and concentrate at the near-clean steps, where the bound audits fine detail — exactly the terms that dominate a log-likelihood. Setting them to one *down-weights the clean end* and spends the network's capacity at mid and high noise, where the large-scale structure of a sample is decided. The result is a deliberate trade: better perceptual samples, a worse (indeed, no longer guaranteed) likelihood bound. The repair is also known precisely: Song, Durkan, et al. (2021) show that one specific reweighting — the likelihood weighting, $g^2(t)$ in the continuous-time notation of the next lecture — restores a rigorous upper bound on the negative log-likelihood. The weights are a dial between the two objectives.

③ **The schedule, and why $T \sim 10^3$.** The pair (schedule β_t, step count T) is a *protocol* in Week 11's sense, and it is priced by Week 11's laws. The per-step dissipation must be kept small for a reason specific to today's ansatz: the *true* one-step reversal $q(x_{t-1} \mid x_t)$ — without the x_0 anchor — is not Gaussian; it is a mixture as structured as the data themselves, and only in the small-β_t limit does it collapse onto a Gaussian (Sohl-Dickstein et al. 2015). The Gaussian ansatz is therefore not innocent: it is an approximation whose error is controlled by the step size, and the $1/K$ dissipation law of Section 22.5 is the reason a thousand gentle steps outperform ten violent ones at fixed total corruption. The worked example below makes this failure visible on demand.

④ **What the network predicts.** Regressing ε, regressing x_0, or regressing the mean μ are algebraically interchangeable — each is an affine function of the others given x_t — but they precondition the network differently: the ε-target has unit variance at every t, while an x_0-target collapses at the clean end and a μ-target nearly duplicates the input. Karras et al. (2022) audit these scalings, together with schedules and samplers, as one decoupled design space; the stat-mech construction fixes none of them, and says so.

Trap

The standard misreadings of the DDPM recipe. ① *L_{simple} is not a likelihood bound.* Dropping the weights w_t forfeits the inequality of Equation 23.3; reporting L_{simple} (or a naively reweighted variant) as a negative log-likelihood is a category error — the honest bound needs the weights, or the likelihood weighting of Song, Durkan, et al. (2021). ② *The Gaussian reverse kernel is an approximation, and the step size is its control parameter.* The exact reversal of one corruption step is as multimodal as the data; only small β_t makes it Gaussian. Large steps do not merely coarsen a sound method — they break its central ansatz (Figure 23.4). ③ *ε_θ is the score in disguise, units included.* Any sampler bolted onto a trained DDPM — Langevin refinement, the reverse SDE, the next lecture's probability-flow ODE — needs $s_\theta = -\varepsilon_\theta/\sqrt{1-\bar{\alpha}_t}$; dropping the rescaling is the classic silent bug, wrong by a factor that varies with t.

23.6 Example: the ansatz needs small steps

Entry ③ locates the construction's single structural approximation and merits a demonstration. We use the running two-mode target with its exact scores, so that

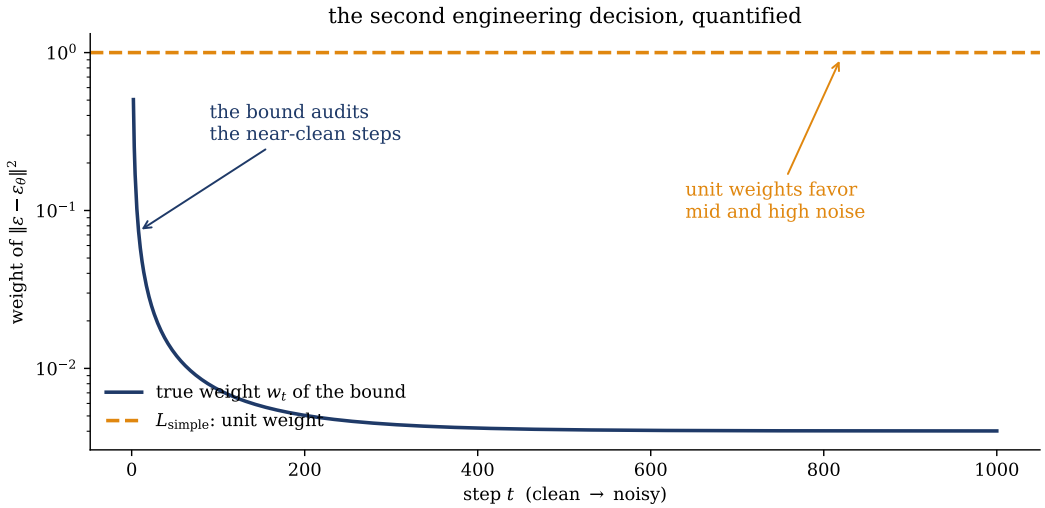

Figure 23.3: What L_{simple} actually reweights. True per-step weights $w_t = \beta_t^2/(2\sigma_t^2\alpha_t(1-\bar{\alpha}_t))$ of the bound Equation 23.5 (navy), for a $T = 1000$ discretization of the Week-10 corruption ($\sqrt{\bar{\alpha}_t} = e^{-4t/T}$, $\sigma_t^2 = \tilde{\beta}_t$), against the unit weight of L_{simple} (orange dashed). The bound concentrates its attention at the near-clean steps — $w_t \approx 0.50$ at $t = 2$ against 0.004 at $t = T$, a factor of 125, diverging as $t \to 1$ — because small corruptions carry the fine, likelihood-dominating detail. Replacing w_t by 1 shifts the training effort toward mid and high noise, where perceptual structure is decided: a deliberate engineering trade of likelihood for sample quality.

learning error is zero by construction and whatever fails is the ansatz. The sampler is exactly Ho's ancestral chain: start at $x_T \sim \mathcal{N}(0,1)$, step down with $x_{t-1} = \mu_\theta(x_t,t) + \sigma_t z$ — the mean built from the exact score via Equation 23.5, $\sigma_t^2 = \tilde{\beta}_t$, no noise on the final step — and vary only the number of steps T at fixed total corruption ($\sqrt{\bar{\alpha}_t} = e^{-4t/T}$, as throughout).

The left panel of Figure 23.4 is the anatomy of the failure. After a two-step corruption each step replaces 98% of the variance, and the true reversal $q(x_0 \mid x_1)$ at $x_1 = 0$ is the data distribution itself, barely tilted: two humps, six units apart. The Gaussian ansatz can place its mean correctly — with the exact score it does, at -1.19, the posterior mean — but a single Gaussian centered between two modes puts its mass where neither mode lives. The right panel shows the consequence and its cure: at $T = 2$ over half the generated mass lands in the inter-mode region that truly carries 10^{-3} of the probability, and the defect drains away as the steps shrink, three orders of magnitude by $T = 64$. Nothing was learned or mislearned anywhere in this figure; the error is purely the distance between the true reversal kernel and the Gaussian family, and step size is the only dial that controls it. This is Week 11's near-equilibrium lesson relocated into a generative model: gentle protocols are the ones whose reversals are simple.

Take-home 3

Four engineering decisions sit on top of the exact construction: fixed reverse variances, unit weights (L_{simple} trades the likelihood bound for sample quality; a specific likelihood weighting restores it (Song, Durkan, et al. 2021)), the

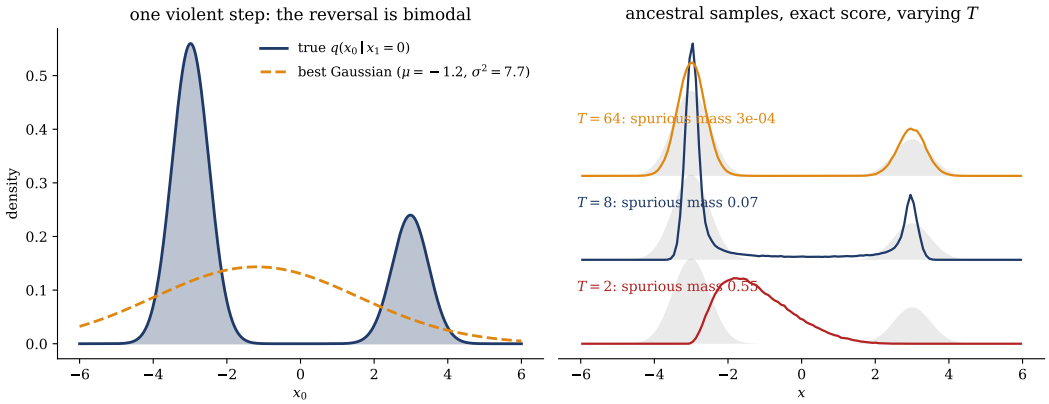

Figure 23.4: The Gaussian ansatz needs small steps — exact scores, so every failure is the ansatz. Left: the true one-step reversal $q(x_0 \mid x_1)$ for a two-step chain ($T = 2$, $\beta_t = 0.98$), evaluated at $x_1 = 0$ (navy): it is bimodal with modes at the two data modes and variance 7.7, and no Gaussian — the moment-matched one is shown in orange — can represent it. Right: ancestral samples against the true density (gray) for $T = 2$, 8, 64 at fixed total corruption. At $T = 2$ the sampler pours 55% of its mass into the region $|x| < 1.5$, which truly holds 0.001; at $T = 8$ the spurious mass is down to 7%; at $T = 64$ it is 3×10^{-4}, with the mode weight at 0.295 against the true 0.300. The steps must be small enough that the true reversal kernel is Gaussian; a thousand gentle steps is a requirement of the ansatz.

schedule with $T \sim 10^3$ (small steps keep the true reversal Gaussian — the ansatz's one structural requirement), and the choice of regression target (ε for its unit variance at every t). Everything else is theorem.

A diffusion model trains by minimizing its own dissipation: the DDPM bound is Week 11's forward/reverse path divergence with the reverse measure made learnable, its per-step ledger is Week 10's denoising score matching in closed form, and the noise the network predicts is the score in different units.

The warm-up card sends you into the framing sections of Song, Sohl-Dickstein, et al. (2021) — the paper the next lecture is built on — with today's dictionary in hand: their forward SDE is Equation 20.1, their discrete "DDPM" is today's chain, and their training loss is Equation 23.5 with a continuous t. The next lecture takes the small-step limit properly — renaming the step index to k, so that t can become continuous time: the score-SDE framework that unifies today's chain with Week 10's continuous picture, the probability-flow ODE — a deterministic sampler with the same marginals and an *exact* likelihood, no bound required — and a look at flow matching, which reaches the same object without any stochastic construction at all. The tutorial (Ph12 106) then puts the two samplers side by side and connects the likelihood machinery to Week 11's AIS estimator; the session uses the ledger Equation 23.4 and the reverse SDE Equation 20.2.

23.7 Outlook: one construction, many parametrizations

The lecture decomposed the workhorse of industrial image generation — a method you trained as a recipe — into objects this course built one at a time: a forward corruption chain (Week 10's Ornstein–Uhlenbeck process, discretized), a variational bound that is a nonequilibrium free-energy estimator (Week 11's Jensen gap, with the reverse path measure handed to a neural network), per-step terms that are denoising score matching in closed form (Week 10's regression), and a residue of four engineering decisions whose costs and repairs are each known precisely. The historical lineage is explicit: the founding paper cited Jarzynski before any of deep learning's own canon.

What the discrete bookkeeping cannot deliver is a likelihood without a bound; that requires the continuous limit. As $T \to \infty$ the chain converges to the score-SDE of Song, Sohl-Dickstein, et al. (2021), Anderson's reverse-time SDE returns as the sampler, and a deterministic sibling — the probability-flow ODE — carries the same marginals while permitting *exact* likelihood evaluation. The partition function that could not be computed in Week 3 has been dodged, bounded, cancelled, removed, measured, and now minimized against, by a model whose training loss is a dissipation.

24 The score SDE: two ways back from noise

Recommended reading

Core (~1 h). Song, Sohl-Dickstein, et al. (2021) — the framework paper, and the warm-up reading (introduction and the SDE-framework section): one forward SDE, its stochastic reversal, and the probability-flow ODE that shares every marginal. The probability-flow section is today's engine; the appendix derivations are worth a look after the lecture. Alongside it, re-read Ho, Jain, and Abbeel (2020) with today's eyes: the variance-preserving SDE, discretized, with a particular loss weighting.
Optional background. Chen et al. (2018) — the instantaneous change of variables $\frac{\mathrm{d}}{\mathrm{d}t}\log p = -\nabla \cdot v$ that makes the ODE's likelihood exact, and the continuous-normalizing-flow lineage behind it. Grathwohl et al. (2019) — how the divergence is estimated at scale (the Hutchinson trace). Sohl-Dickstein et al. (2015) — the founding paper, one last time: everything today unifies descends from its forward/reverse construction. Lipman et al. (2023) — flow matching, the contemporaneous alternative that learns the deterministic velocity directly (skim). Karras et al. (2022) — the modern design-space view: schedule, scaling, and sampler as independent knobs (named today, not developed). Welling, Lu, and Holdijk (2026) — the companion text's continuous-time and score-SDE chapters cover exactly this material.
Prerequisite reminder. From Week 10: the score $\nabla_x \log p_t$ and its Z-freeness (Equation 19.1), denoising score matching (Equation 19.4 and Section 20.4), the forward noising process (Equation 20.1), and Anderson's reverse-time SDE (Equation 20.2), whose Fokker–Planck derivation (Section 20.3) is re-run today with a different destination. From Week 9: the Fokker–Planck equation itself (Equation 17.3). From Week 11: the ELBO/AIS bound and its dissipation gap (Equation 22.4). From the previous lecture: DDPM — the discrete forward chain, the variational bound, and the ε-prediction loss. New content: the forward-SDE family and its VP/VE members, the probability-flow ODE derived from the Fokker–Planck equation, the exact likelihood by the instantaneous change of variables, and the unification of DDPM, NCSN, and the reverse SDE into one construction.

24.1 One model, three arbitrary choices

The previous lecture assembled the industrial object. DDPM fixes a forward Markov chain $q(x_k \mid x_{k-1}) = \mathcal{N}(\sqrt{1-\beta_k}\, x_{k-1},\, \beta_k)$ that corrupts data to noise in K discrete steps; trains a reverse chain by a variational bound (ELBO) on $\log p$; and collapses that bound, after reparameterization, into ε-prediction — which is denoising score matching (Equation 19.4) at each noise level, in different units. The lecture closed on

the structural identification that Week 11 prepared: the ELBO's slack is a Kullback–Leibler divergence between the forward corruption path and the learned reversal — a dissipation, the same quantity that priced annealed importance sampling in Equation 22.4. A DDPM is a learned nonequilibrium transport, trained by minimizing its own dissipated work.

Today's move is to notice how much of that construction is choice rather than law. DDPM fixes a *discretization* (a finite chain of K steps), a *schedule* (a particular sequence β_k), and a *loss weighting* (the "simple" ε-objective) — and, most consequentially, it only ever *bounds* the log-likelihood. Each choice can be stripped away, and Song, Sohl-Dickstein, et al. (2021) stripped all of them at once: underneath sits a single continuous object, a forward stochastic differential equation, whose reversal needs only the time-dependent score the model already learns. And the continuous object is more than tidier notation, because it admits not one way back but two. One is Anderson's reverse-time SDE (Equation 20.2), stochastic and familiar from Week 10. The other is new, and it is the reason for this lecture: a *deterministic* ordinary differential equation with exactly the same time-marginals — and a deterministic invertible flow can do the one thing no ELBO can, namely compute its model's log-likelihood *exactly*. The natural objection — how can a deterministic ODE and a noise-driven SDE possibly produce the same samples? — is the one the warm-up card poses. The answer is that they agree on every *marginal* p_t while disagreeing on every individual *path*, and a generative model is judged on marginals.

The history compresses two years of parallel invention. Sohl-Dickstein et al. (2015) built generation from nonequilibrium thermodynamics — destroy structure with a fixed diffusion, learn the reversal — and cited Jarzynski while doing it. Ho, Jain, and Abbeel (2020) made the construction work at scale, as the previous lecture's DDPM. Independently, Song and Ermon (2019) reached generation through the score itself: networks fitted at a ladder of noise scales, sampled by annealed Langevin dynamics (Section 20.4). Song, Sohl-Dickstein, et al. (2021) showed these were a single framework seen through two discretizations, and added the exact likelihood that none of the predecessors had. The question, then: *what is the schedule-free, discretization-free object beneath DDPM, and what does its deterministic reversal buy?*

24.2 The forward-SDE family

Write the continuous forward process in full generality:

$$\mathrm{d}x = f(x,t)\,\mathrm{d}t + g(t)\,\mathrm{d}W, \qquad x_0 \sim p_{\text{data}},$$

the forward-SDE family: one process, a schedule choice

run to a horizon T at which the marginal p_T is a tractable Gaussian. (Notation panel, once: T is the diffusion *horizon*, not a temperature, and W is a *Wiener process* again — Week 11's work variable is retired, and $\beta(t)$ below is a noise schedule, not an inverse temperature.) Two members of the family carry all the history. The *variance-preserving* (VP) choice takes $f = -\frac{1}{2}\beta(t)\,x$ and $g = \sqrt{\beta(t)}$: the drift shrinks every sample toward the origin at exactly the rate the noise inflates it, and $p_T \to \mathcal{N}(0, I)$. Week 10's workhorse (Equation 20.1) is the constant-schedule member $\beta \equiv 2$. The *variance-exploding* (VE) choice takes $f = 0$ and $g = \sqrt{\mathrm{d}[\sigma^2(t)]/\mathrm{d}t}$: no shrinkage, noise piled on top of the data until it drowns them, $p_T \to \mathcal{N}(0, \sigma_{\max}^2 I)$ — and this is precisely the noise ladder of Song and Ermon (2019), made continuous.

The claim that DDPM lives in this family is a two-line computation. Set $\beta_k = \beta(t_k)\,\Delta t$ with step $\Delta t = T/K$ and expand the previous lecture's chain for small Δt:

$$x_k = \sqrt{1-\beta_k}\,x_{k-1} + \sqrt{\beta_k}\,\varepsilon_k = x_{k-1} \underbrace{-\tfrac{1}{2}\beta(t_k)\,x_{k-1}\,\Delta t}_{f\,\Delta t} + \underbrace{\sqrt{\beta(t_k)\,\Delta t}\,\varepsilon_k}_{g\,\Delta W_k} + \mathcal{O}(\Delta t^2),$$

which is the Euler–Maruyama discretization of the VP-SDE, step by step. DDPM's forward chain is not *like* a diffusion; it *is* the VP diffusion, sampled at K points. The marginals stay closed-form, as they must for denoising score matching to work: for the VP member, $x_t \mid x_0 \sim \mathcal{N}(x_0\,e^{-B(t)/2},\, 1-e^{-B(t)})$ with $B(t) = \int_0^t \mathrm{d}s\;\beta(s)$ — data convolved with a known Gaussian at every t, exactly the setting of Section 20.4. One t-conditioned network $s_\theta(x,t) \approx \nabla_x \log p_t(x)$, trained by denoising score matching across the whole continuum of times, is the single learned object in everything that follows.

The stochastic way back we already own. Anderson's theorem (Equation 20.2), derived in Section 20.3 for exactly this family, reverses the forward SDE as

$$\mathrm{d}x = \left[f(x,t) - g(t)^2\,\nabla_x \log p_t(x)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{W},$$

integrated from T down to 0: a stochastic generator, injecting fresh noise at every step, reproducing every marginal p_t exactly. So one process with marginals p_t runs forward, and a second, noise-driven one runs backward. The question the lecture turns on is whether a *third* process — deterministic, no $\mathrm{d}\bar{W}$ at all — can carry the same marginals. If it can, it is an invertible map between noise and data, and invertible maps have exactly computable likelihoods. The answer is one rewrite of the Fokker–Planck equation away.

> Take-home 1
>
> DDPM is one instance of a continuous family: the forward SDE $\mathrm{d}x = f\,\mathrm{d}t + g\,\mathrm{d}W$, with VP ($f = -\frac{1}{2}\beta x$, $g = \sqrt{\beta}$; DDPM is its Euler–Maruyama discretization) and VE ($f = 0$; the NCSN noise ladder) as the two canonical schedules. One learned object serves the whole family — the time-marginal score $s_\theta(x,t)$, fit by denoising score matching — and the reverse-time SDE is its stochastic way back. Today's question: is there a deterministic way back with the same marginals?

24.3 The engine: Fokker–Planck as pure transport

We derive the deterministic generator in full; it runs entirely on Week 9's machinery. Start from the marginal flow of the forward SDE, the Fokker–Planck equation of Equation 17.3 (scalar $g(t)$ here; the matrix-valued case adds indices and nothing else):

$$\partial_t\, p_t = -\nabla_x \cdot (f\,p_t) + \tfrac{1}{2} g^2\,\nabla_x^2\, p_t.$$

The central step is a rewrite already used in Step 2 of the reversal argument in Section 20.3, where it served as scaffolding; here it is the result. The diffusion term is secretly a transport term: since $\nabla_x p_t = p_t\, \nabla_x \log p_t$,

$$\tfrac{1}{2} g^2\, \nabla_x^2\, p_t = \nabla_x \cdot \Big(\tfrac{1}{2} g^2\, \nabla_x p_t \Big) = \nabla_x \cdot \Big(\underbrace{\tfrac{1}{2} g^2\, (\nabla_x \log p_t)}_{\text{a velocity field}}\; p_t \Big),$$

and the whole Fokker–Planck equation collapses into a single probability current:

$$\partial_t\, p_t = -\nabla_x \cdot (v_t\, p_t), \qquad v_t(x) = f(x,t) - \tfrac{1}{2} g(t)^2\, \nabla_x \log p_t(x).$$

the marginal flow as a continuity equation

Now read the display as a physicist. A *continuity equation* $\partial_t p = -\nabla \cdot (vp)$ is precisely the statement that probability is carried by the deterministic flow $\dot{x} = v_t(x)$ — it is the Liouville transport equation of that ODE, the same bookkeeping that conserves phase-space density in classical mechanics. The display carries no trace of the velocity's noisy origin. Therefore the ordinary differential equation with velocity v_t transports an initial density p_0 through *exactly* the marginals p_t of the diffusion, and we box the master result (Song, Sohl-Dickstein, et al. 2021):

$$\boxed{\mathrm{d}x = \Big[f(x,t) \;-\; \tfrac{1}{2}\, g(t)^2\, \nabla_x \log p_t(x) \Big]\, \mathrm{d}t} \tag{24.1}$$

the probability-flow ODE — same marginals p_t as the forward and reverse SDE (Song et al. 2021)

In words: the diffusion term has been split in half — one half became deterministic transport, absorbed into the velocity as $-\frac{1}{2} g^2 \nabla_x \log p_t$, and the other half was the injected noise, now removed entirely. Compare the drifts, because the factor of $\frac{1}{2}$ is the central distinction: the reverse SDE carries the *full* $g^2 \nabla_x \log p_t$ and re-injects noise; the probability-flow ODE carries *half* and injects nothing. The missing $\frac{1}{2} g^2 \nabla_x \log p_t$ of drift is exactly compensated by the missing diffusion — same marginals, no randomness. Run backwards from $x_T \sim p_T$, the ODE is a deterministic generator; run forwards from a data point, it is a deterministic encoder; and because it is a smooth, invertible flow, it is about to become a likelihood machine.

24.4 The exact likelihood

What does a density do along a deterministic flow? The continuity equation answers in one line. Divide $\partial_t p = -(\nabla_x \cdot v)\, p - v \cdot \nabla_x p$ by p:

$$\partial_t \log p_t = -\nabla_x \cdot v_t - v_t \cdot \nabla_x \log p_t,$$

and evaluate along a trajectory $\dot{x} = v_t(x)$ of the flow itself, where the total derivative picks up a convective term of the opposite sign:

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(x(t)) = \partial_t \log p_t + \dot{x} \cdot \nabla_x \log p_t = -\nabla_x \cdot v_t(x(t)).$$

the instantaneous change of variables (Chen et al. 2018)

The convective terms cancel, and what remains is the *instantaneous change of variables* (Chen et al. 2018): along the flow, the log-density changes at exactly the negative divergence of the velocity — where the flow compresses ($\nabla \cdot v < 0$), density

rises; where it spreads, density falls. This is the continuous-time limit of the Jacobian determinant in an ordinary change of variables, accumulated infinitesimally instead of computed all at once. Integrate from 0 to T, where the endpoint density p_T is the known Gaussian, and rearrange:

$$\boxed{\log p_0(x_0) = \log p_T(x(T)) + \int_0^T \mathrm{d}t\, \nabla_x \cdot v_t(x(t))} \tag{24.2}$$

the exact model log-likelihood, by integrating the flow

with $x(t)$ the probability-flow trajectory launched from x_0. The recipe reads straight off the box: push the data point forward to noise along the ODE, accumulate the divergence of the velocity along the way, add the tractable Gaussian log-density at the far end — and the result is the *exact* log-likelihood of the model, not a bound on it. Normalization has kept this quantity out of reach since Week 1. The previous lecture's ELBO under-reports $\log p$ by its dissipation; Week 11's AIS delivers a stochastic lower bound whose gap is the dissipated work (Equation 22.4); the probability-flow ODE closes the gap to zero, because a deterministic, invertible transport dissipates nothing. Three practical remarks belong in the record. In high dimension the divergence $\nabla_x \cdot v$ is a trace of a $d \times d$ Jacobian, estimated unbiasedly by the *Hutchinson trace* $\nabla_x \cdot v = \langle \epsilon^\top (\partial_x v)\, \epsilon \rangle_{\epsilon \sim \mathcal{N}(0,I)}$ at the cost of one vector–Jacobian product (Grathwohl et al. 2019). The construction identifies the diffusion model as a *continuous normalizing flow* (Chen et al. 2018) whose velocity is fixed by the forward SDE and the learned score, rather than a free neural network. And there is a matching training-side result: a particular time-weighting of the denoising score-matching loss upper-bounds the negative log-likelihood, so the model can also be *trained* in likelihood units (Song, Durkan, et al. 2021) — named here, not derived.

Trap

The standard misreadings of the two reversals. ① *Marginals, not paths — and mind the* $\frac{1}{2}$. The reverse-SDE drift carries $f - g^2 \nabla_x \log p_t$ plus noise; the ODE velocity carries $f - \frac{1}{2} g^2 \nabla_x \log p_t$ and no noise. The two agree on every distribution p_t and on no individual trajectory; copying the SDE drift into the ODE (or the reverse) is the standard algebra error. ② *Exact is not the same as correct.* Equation 24.2 is the exact likelihood *of the model the learned score defines.* With an imperfect s_θ it can sit above or below the data's true log-density — it is not a lower bound, and it inherits every flaw of the score. ③ *The score needed is that of* p_t *for all* t — Week 10's trap, still load-bearing: the data score alone steers nothing out of noise. ④ *DDPM, NCSN, and the score SDE are one model, not three.* ε-prediction, the σ-scaled score, and $\nabla_x \log p_t$ differ by fixed rescalings; VP versus VE is a schedule choice, not a framework choice. Treating them as rivals misreads the whole construction.

Take-home 2

Rewriting the Fokker–Planck equation as a continuity equation yields the probability-flow ODE (Equation 24.1): a deterministic flow with velocity $v_t = f - \frac{1}{2} g^2 \nabla_x \log p_t$ that shares every marginal with the diffusion. Being an invertible flow, it computes the model's log-likelihood exactly by the instantaneous change of variables (Equation 24.2) — where the ELBO and AIS only bound it, with a gap equal to their dissipation. Same marginals, different

paths; note the $\frac{1}{2}$.

24.5 The unification: Module 4 from four sides

The module's inventory can now be read off one construction, and Figure 24.1 is that reading as a picture. A diffusion model *is* a forward SDE $\mathrm{d}x = f\,\mathrm{d}t + g\,\mathrm{d}W$ together with a single learned time-marginal score $s_\theta(x,t)$. DDPM is the VP member, discretized by Euler–Maruyama, trained with one loss weighting; its ELBO is the discrete-time shadow of the continuous bound. NCSN and its annealed-Langevin sampler (Song and Ermon 2019) are the VE member: the ladder of noise scales is the VE schedule discretized, and annealed Langevin is a discretization of the reverse dynamics. Both are trained by denoising score matching across noise levels (Equation 19.4) — ε-prediction and σ-scaling are unit conversions on one objective. And from the same learned score there are two ways back: the reverse SDE (stochastic; fresh noise at each step partially corrects accumulated score error, which is why it tends to win on sample quality) and the probability-flow ODE (deterministic; fewer, larger steps; a bijective encoder; and the exact likelihood of Equation 24.2). Choose the sampler by purpose — the SDE for pictures, the ODE for numbers — and the choice is independent of schedule, architecture, and loss, which is the design-space decoupling that Karras et al. (2022) later made systematic.

The Week 11 thread ties off in one sentence, and the tutorial makes it concrete: a diffusion model is a *learned annealed importance sampler* — the forward SDE is the annealing ladder, the reverse dynamics the transport, the ELBO the Jarzynski-style path-KL bound of Equation 22.4 — with the probability-flow ODE supplying, at last, the exact number that AIS could only bound. And one contemporaneous development should be named. If the deterministic velocity field is the object that matters, one can regress a network onto it *directly* — flow matching (Lipman et al. 2023) fits $v_\theta(x,t)$ to closed-form conditional velocities along prescribed interpolation paths, obtaining the transport without ever writing a stochastic process or a score. That the field's newest mainline arrived by deleting the noise from the construction is a fair measure of how central the probability-flow view has become; conditional generation and guidance are further engineering on the same score field, and we leave them to the literature.

Take-home 3

Four Module-4 objects are one construction. The forward SDE (VP = DDPM, VE = NCSN) fixes the marginals; denoising score matching fits their score, whatever the parameterization; the reverse SDE and the probability-flow ODE are two samplers for the same model — stochastic for sample quality, deterministic for speed, encoding, and the exact likelihood; and the training bound is the dissipation of a learned annealed importance sampler (Week 11). Schedule, discretization, loss weighting, and sampler are independent knobs on one machine.

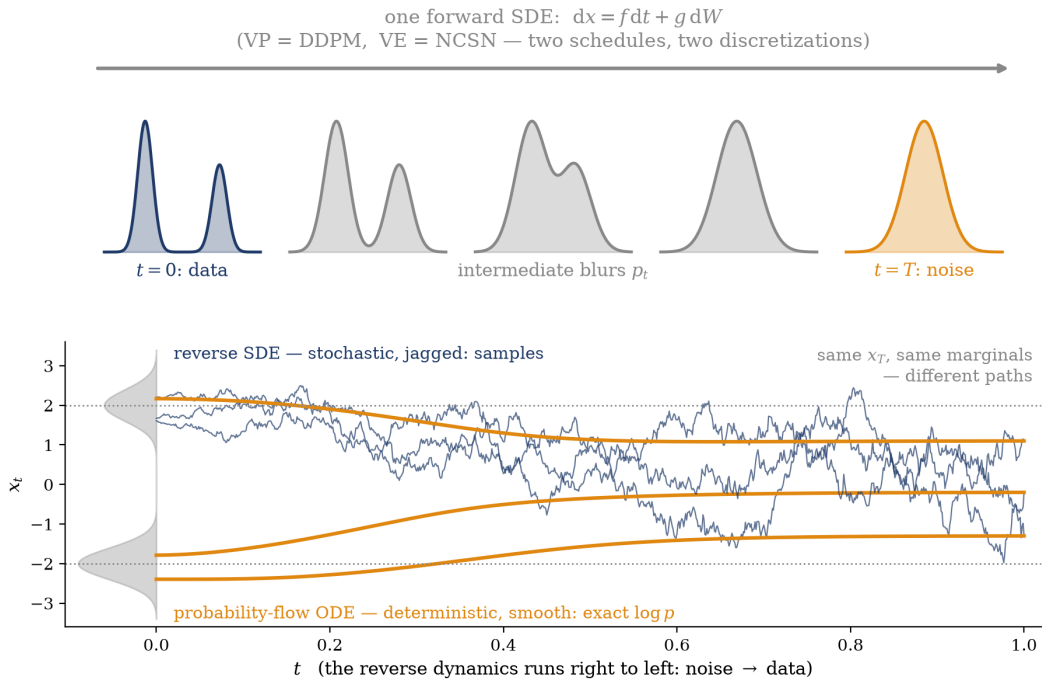

Figure 24.1: The whole of Module 4 in one picture. Top: a single forward SDE carries the data density (navy) through blurred intermediates p_t (gray) to a tractable Gaussian (orange); DDPM and NCSN are its VP and VE schedules, discretized. Bottom: from the same terminal noise, two reversals driven by the same learned score — the stochastic reverse SDE (navy, jagged) and the deterministic probability-flow ODE (orange, smooth) — reproduce the same marginals along visibly different paths, and land on the same data density (gray, left edge). Trajectories computed with the exact score of the running two-mode example under a VP schedule.

24.6 Example: two samplers and an exact likelihood

The framework's three claims — same marginals, exact likelihood, bound strictly below — are all checkable on a target where nothing is learned and nothing is hidden. Take the two-mode mixture $p_{\text{data}} = 0.6\,\mathcal{N}(-2, 0.4^2) + 0.4\,\mathcal{N}(2, 0.4^2)$ under a VP forward SDE with linear schedule $\beta(t) = 0.1 + 19.9\,t$ on $t \in [0, 1]$. A Gaussian mixture stays a Gaussian mixture under the VP kernel, so the marginals p_t, the score, and even the score's spatial derivative are closed-form — the demonstration isolates the framework from learning error, exactly as in Section 20.5. We sample 20,000 points through both reversals (Euler–Maruyama for the SDE, a fixed-step Runge–Kutta integrator for the ODE), then compute the ODE likelihood Equation 24.2 at a grid of test points, and finally a DDPM-style ELBO on a K-step discretization of the same model.

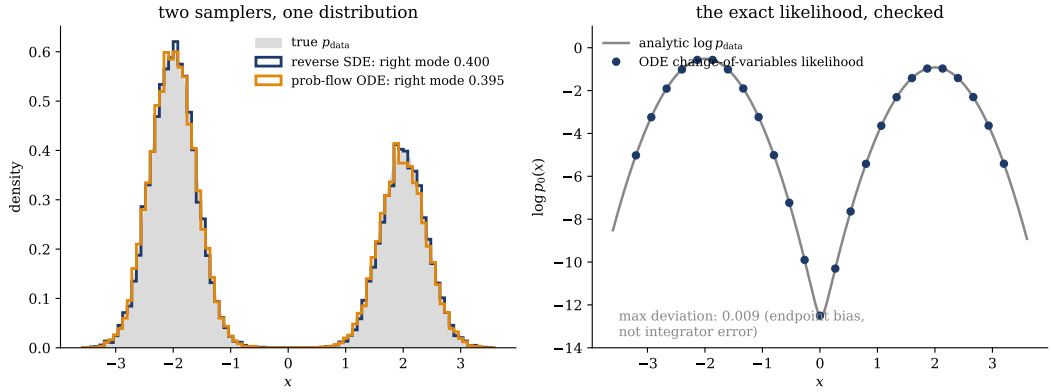

Figure 24.2: Two reversals of one diffusion, with the exact score (VP schedule, $\beta(t) = 0.1 + 19.9\,t$; 20,000 samples each, common initial draws from $\mathcal{N}(0, 1)$). Left: sample histograms from the reverse SDE (navy; Euler–Maruyama, 1000 steps) and the probability-flow ODE (orange; RK4, 400 steps) against the true data density (gray) — the measured right-mode weights are 0.400 and 0.395 against a true 0.400: same marginals, one stochastic path family and one deterministic. Right: the exact log-likelihood by the flow, Equation 24.2 — the ODE integrated forward from each test point with the divergence accumulated along the trajectory (navy dots) — against the analytic $\log p_{\text{data}}$ (gray curve). The agreement is a few times 10^{-3}; the visible residual is not integrator error but the endpoint bias $p_T \neq \mathcal{N}(0, 1)$ flagged in Section 20.2, here of order $e^{-B(T)/2} \approx 7 \times 10^{-3}$.

The left panel of Figure 24.2 settles that objection with data: a noise-driven and a noiseless process, started from the same Gaussian draws, deliver the same distribution — down to the mode weights, which Week 10 showed is where samplers fail. The right panel is the key result: the divergence integral of Equation 24.2 lands on the analytic log-density across five orders of magnitude in probability, and its small residual has a physical name (the terminal marginal has not quite reached the unit Gaussian) rather than a statistical one. A deterministic transport produces a likelihood with zero variance, which no stochastic estimator can.

Figure 24.3 makes the comparison quantitative. At $K = 32$ steps the DDPM-style bound under-reports the log-likelihood by about one nat at every test point — that offset *is* the dissipation of the discrete transport, Week 11's Jensen gap produced

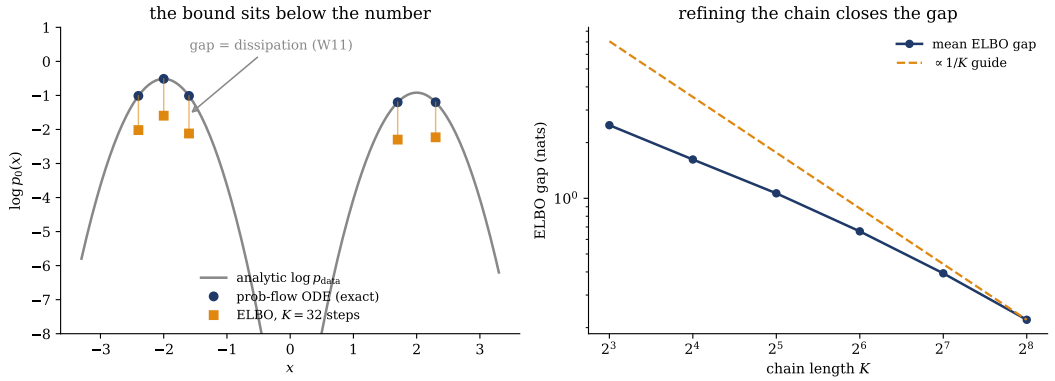

Figure 24.3: The bound and the number. Left: at five in-support test points, the K-step DDPM-style ELBO (orange: $K = 32$; 4000 forward paths per point, EM reverse kernels built from the exact score) sits strictly below the analytic $\log p_{\text{data}}$ (gray curve), which the deterministic flow reproduces exactly (navy dots); the vertical offset is the discretization's dissipated work, $D_{\text{KL}}(q_F \| q_R)$ of Equation 22.4. Right: the mean ELBO gap against the chain length K, with a $1/K$ guide (dashed) — refining the discretization recovers the continuum and the bound tightens toward the exact value, the near-equilibrium law of Section 22.5 operating inside a diffusion model.

inside a diffusion model — and lengthening the chain shrinks it toward the $1/K$ law that priced annealed importance sampling in Section 22.5. The deterministic flow stands outside this accounting: it sits on the exact value at any budget, because it never dissipates. Bound, dissipation, and exact number are on one plot, and they are the same three objects the course met as the second law, the dissipated work, and the free energy.

> *One forward SDE, one learned score, two ways back — a noisy reverse SDE for samples and a deterministic ODE for the exact likelihood; DDPM and score matching were the same model, discretized two ways.*

The tutorial (16–18, Ph12 106) is derivation-led, and it closes the module's practical arc: train a small score network on a two-dimensional density, sample it through *both* reversals, and compute its exact log-likelihood by the probability-flow ODE — the learned counterpart of everything the example above did with closed-form scores. The session uses Equation 24.1 and the change-of-variables identity, and ends by identifying the Week-11 ancestor of the training loss.

24.7 Outlook: the title, discharged

Week 12 closes the course's main construction. The partition function was raised as the central obstruction in Week 1 and has organized every module since: Module 1 dodged its gradient with contrastive divergence, Module 2 bounded it with the variational free energy, Module 3 cancelled it inside Metropolis ratios, Week 10 removed it from training and generation outright, and Week 11 measured it with nonequilibrium work. This week the generative model built on the Z-free score reported its own log-likelihood exactly (Equation 24.2), and Figure 24.4 closes the

figure the course has been extending since Figure 19.4. The chain of identifications runs from Boltzmann's e^{-E}/Z to a diffusion model's reverse-time flow: the energy defined a score, the score defined a drift, the drift defined a transport, and the transport's divergence integral yields the likelihood.

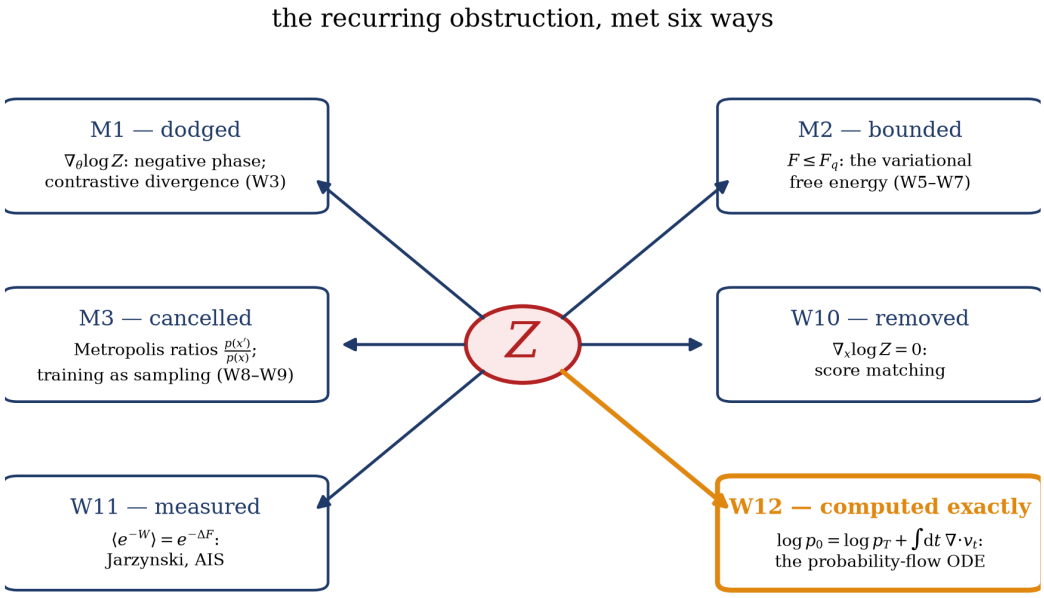

Figure 24.4: The course's spine, discharged. The partition function, met six ways: dodged (M1), bounded (M2), cancelled (M3), removed (W10), measured (W11) — and finally, this week, made irrelevant to the one question it always blocked: the model built on the Z-free score computes its own log-likelihood exactly, through the probability-flow ODE.

Two weeks remain, and they turn the machinery back on learning itself. Week 13 leaves generative modeling for the return direction of the bridge: its first lecture casts maximum entropy and the *information bottleneck* as constrained free-energy minimizations — the same variational language, applied to representation and compression rather than generation, with a precise account of what that framework establishes about deep networks and what it does not — and its second puts a partition function on the hypotheses of an inference problem, where sharp detectability thresholds separate the recoverable from the impossible, and a further threshold separates what is possible from what is efficiently achievable. That lecture does something the course has not done before: it uses statistical physics to decide what a learning problem admits, rather than to supply a method for one. Week 14 then reads all five modules off a single variational identity, prepares the exam, and surveys the research programme Week 13's material is drawn from (Zdeborová and Krzakala 2016). On the generative side the onward reading is the flow-matching line named above (Lipman et al. 2023), together with Schrödinger bridges and the optimal-transport view of diffusion, for which the companion text (Welling, Lu, and Holdijk 2026) is the natural start. The construction ends here; what follows uses it on questions of a different kind.

Part V

Module 5 — Information, free energy & bridges

25 The information bottleneck: representation at a temperature

Recommended reading

Core (~1 h). Tishby, Pereira, and Bialek (1999) — four pages of statistical physics under an information-theory title, and the warm-up reading for the next lecture: the rate–relevance Lagrangian, its variational solution, and the self-consistent equations that are today's engine. Read §2–3 knowing what to look for: a Boltzmann distribution, a partition function, and a Lagrange multiplier that is an inverse temperature. Alongside it, the rate–distortion section of Shannon (1948) — where "bits spent describing X" acquires its precise meaning, and the frame the bottleneck generalizes.
Optional background. Rose (1998) — deterministic annealing and the phase transitions of rate–distortion, the physics reading of today's material worked out a decade before the bottleneck paper; Alemi et al. (2017) — the trainable version, whose variational bound is Week 7's ELBO with a predictive target; Tishby and Zaslavsky (2015) and Shwartz-Ziv and Tishby (2017) — the claim that deep networks themselves traverse the information plane, and Saxe et al. (2018) — its rebuttal, the pair to be read together; Friston (2010) — the free-energy principle, the most ambitious extension of this module's language, read for the claim and its critique rather than as settled theory.
Prerequisite reminder. The Boltzmann distribution, Z, $\beta = 1/T$, and $F = -T \ln Z$ (Equation 1.2); entropy and information in nats, and Jaynes' maximum-entropy route to the canonical ensemble (Section 1.4); the variational free energy and the KL divergence as a free-energy difference (Equation 9.1); the ELBO and amortized inference (Equation 14.5, Equation 14.6); annealing as a cooling schedule (Chapter 16); the mean-field self-consistent equations (Equation 9.3). New content: the rate–relevance trade-off and the information plane, the bottleneck Lagrangian and its self-consistent equations, the free-energy reading with β as an inverse temperature, the structural phase transitions of the representation, the deep variational bottleneck, and the free-energy principle stated critically.

25.1 A constraint completes the principle

Week 12 closed the course's main construction, and Module 5 — this week, plus the synthesis of Week 14 — widens the view. Before the machinery is put away, we ask how far its central principle reaches, and the principle in question is the variational one. Three times the course has met it. Jaynes' maximum-entropy argument (Section 1.4) produced the Boltzmann distribution by maximizing entropy at fixed mean energy — a constrained optimization whose Lagrange multiplier turned out to be the inverse temperature. The Gibbs–Bogoliubov identity (Equation 9.1) turned

inference into minimization: the variational free energy exceeds the true one by exactly $T\,D_{\mathrm{KL}}(q\|p)$, so approximating a distribution and minimizing a free energy are the same act. And Week 7 trained that act, with the VAE descending the same functional by stochastic gradients (Equation 14.6). What none of these answers is a question that sits upstream of all of them: what should a *representation* be? A latent code, a layer's activations, a clustering, a sufficient statistic — every learned system keeps some aspects of its input and discards the rest, and the free-energy principle as stated so far offers no criterion for the choice.

The criterion, due to Tishby, Pereira, and Bialek (1999), falls out of the same principle once a constraint is attached. Pose learning, at its most general, as lossy compression toward a goal: given an input X and a target Y, find a compressed description $\tilde{X}$ of X that keeps what predicts Y and discards everything else. This is not the autoencoder's task — Week 7 asked $\tilde{X}$ to reconstruct X itself — but the sharper one of keeping only what matters, with "matters" defined by the prediction target. Shannon supplies the two measures. The *rate* $I(X;\tilde{X})$ counts the nats the description spends on X; the *relevance* $I(\tilde{X};Y)$ counts the nats it retains about Y; and the compressed variable is required to see the target only through the input, the Markov chain $X \to \tilde{X} \to Y$. The two measures are coupled. Zero rate forces zero relevance, since a description carrying no information about X carries none about anything downstream of it; the identity map $\tilde{X} = X$ keeps every relevant nat and every irrelevant one with it. The interesting encoders live between the extremes, and the *information bottleneck* is the principle that selects them: minimize the rate while maximizing the relevance, at an exchange rate β whose interpretation as an inverse temperature the derivation of Section 25.3 establishes.

Shannon (1948) defined the rate–distortion function — the least rate at which a source can be described without exceeding a prescribed distortion — but left the distortion measure to be chosen by hand, and the choice always smuggled in the answer to "what matters." Tishby, Pereira, and Bialek (1999) removed the hand: let the prediction task itself define the distortion, and the trade-off becomes intrinsic. The statistical mechanics predated the bottleneck paper — Rose (1998), reviewing a programme begun a decade earlier, had worked out rate–distortion theory as an equilibrium system, with the trade-off multiplier an inverse temperature and the optimal code reorganizing through genuine phase transitions as it cools. The identification below runs in the bottleneck's setting, and Alemi et al. (2017) turns the result into a trainable network (Section 25.5). The next lecture takes the phase-transition thread and makes it the topic; Week 14 states the whole course as one functional.

25.2 Rate, relevance, and the information plane

We fix notation first, because the bottleneck literature has a trap in it. The compressed variable is written $\tilde{X}$ throughout — as in Tishby, Pereira & Bialek's original paper; the deep-learning-era bottleneck literature calls it T, and we decline to follow it, since T has meant temperature for twelve weeks and is about to mean temperature again in the same equation. Both mutual informations are functionals of one object, the stochastic *encoder* $p(\tilde{x} \mid x)$:

$$I(X;\tilde{X}) = \sum_{x,\tilde{x}} p(x)\,p(\tilde{x}\mid x)\ln\frac{p(\tilde{x}\mid x)}{p(\tilde{x})}, \qquad I(\tilde{X};Y) = \sum_{\tilde{x},y} p(\tilde{x},y)\ln\frac{p(y\mid\tilde{x})}{p(y)},$$

where the marginal $p(\tilde{x}) = \sum_x p(x)\,p(\tilde{x}\mid x)$ and the joint $p(\tilde{x},y) = \sum_x p(x,y)\,p(\tilde{x}\mid x)$ are both induced by the encoder — the second equality using the Markov property, since $\tilde{x}$ learns about y only through x. Every possible encoder therefore maps to a point $(I(X;\tilde{X}),\ I(\tilde{X};Y))$ in the *information plane*, drawn in Figure 25.1: rate on the horizontal axis, relevance on the vertical. Two bounds fence the achievable region. The data-processing inequality caps relevance at the total $I(X;Y)$ — no processing of X can know more about Y than X does — and relevance can never exceed rate, the same inequality read along the chain $Y \to X \to \tilde{X}$ (the encoder cannot pack more about Y into $\tilde{X}$ than the bits it spends describing X). The achievable points fill a region whose upper boundary, the *IB curve*, is the object of the lecture: the most relevance obtainable at each rate, concave, rising steeply from the origin and saturating toward $I(X;Y)$.

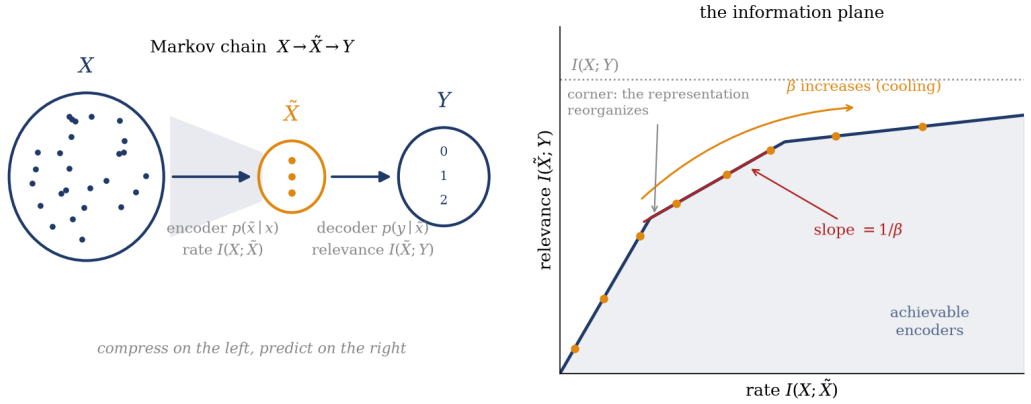

Figure 25.1: The central picture of the lecture. Left: the Markov chain $X \to \tilde{X} \to Y$, pinched at the compressed variable — the encoder spends rate $I(X;\tilde{X})$ describing the input, the decoder keeps relevance $I(\tilde{X};Y)$ about the target. Right: the information plane. Every encoder is a point; the achievable region (shaded) is bounded above by the concave IB curve, whose local slope is $1/\beta$ — the temperature. Increasing β cools the encoder up the frontier, and at discrete corners the optimal representation reorganizes; the worked example of Section 25.6 computes this curve, corners included, for a small joint distribution.

Optimal encoders sit on the frontier, and the standard device converts the constrained problem — maximize $I(\tilde{X};Y)$ at fixed rate — into an unconstrained one. Introducing a Lagrange multiplier β for the constraint gives the **information-bottleneck Lagrangian**,

$$\boxed{\mathcal{L}[p(\tilde{x}\mid x)] \;=\; I(X;\tilde{X}) \;-\; \beta\, I(\tilde{X};Y),} \tag{25.1}$$

information-bottleneck Lagrangian (Tishby, Pereira & Bialek 1999)

to be minimized over all normalized conditionals $p(\tilde{x}\mid x)$. Keep the signs straight, because flipping them inverts the whole construction: the rate is the *cost* and enters with a plus, the relevance is the *benefit* and is subtracted with weight β. Geometrically, β selects the operating point: minimizing Equation 25.1 at a given β picks

the point of the frontier where the tangent slope $\mathrm{d}I(\tilde{X};Y)/\mathrm{d}I(X;\tilde{X})$ equals $1/\beta$. Small β weights compression heavily and lands near the steep origin, where the first nats of rate buy relevance cheaply; large β demands fidelity and rides the flat tail toward saturation, where each additional nat of relevance costs many nats of rate. The slope of the frontier, in other words, behaves exactly like a temperature $1/\beta$. Jaynes' construction (Section 1.4) already showed that a multiplier enforcing a constraint on an entropy-like quantity *is* an inverse temperature; the bottleneck inherits the identification, and the derivation of the next section makes it literal, partition function included.

Two remarks before the calculus. First, minimizing Equation 25.1 is a variational problem, not an ordinary optimization: the unknown is an entire conditional distribution, one probability vector for every input x, and both terms depend on it also implicitly, through the induced $p(\tilde{x})$ and $p(\tilde{x},y)$. Second, β is our old symbol with its old meaning restored. It was the inverse temperature of Week 1, was flagged against the ML community's unrelated uses (the Adam moments, the -VAE weight), and spent Week 12 as a noise schedule β_t (Section 23.2); here it returns as a genuine $1/T$, and the identification is completed in Section 25.3, where a partition function appears beside it.

> Take-home 1
>
> Learning, posed at its most general, is lossy compression toward a target: over the Markov chain $X \to \tilde{X} \to Y$, minimize the rate $I(X;\tilde{X})$ while maximizing the relevance $I(\tilde{X};Y)$. The achievable encoders fill a region of the information plane bounded by the concave IB curve, and the Lagrangian $\mathcal{L} = I(X;\tilde{X}) - \beta\, I(\tilde{X};Y)$ (Equation 25.1) selects the frontier point of slope $1/\beta$ — the multiplier is an exchange rate between nats spent and nats predicted, and, as with Jaynes' maximum entropy, it will prove to be an inverse temperature.

25.3 The engine: stationarity of the Lagrangian

What encoder makes Equation 25.1 stationary? The answer's *form* is the bridge to statistical mechanics, so we carry the variation out in full. Write the Lagrangian with its normalization constraints attached,

$$\mathcal{L}_\lambda = \sum_{x,\tilde{x}} p(x)\,p(\tilde{x}\mid x)\ln\frac{p(\tilde{x}\mid x)}{p(\tilde{x})} - \beta\sum_{\tilde{x},y} p(\tilde{x},y)\ln\frac{p(\tilde{x},y)}{p(\tilde{x})\,p(y)} + \sum_x \lambda(x)\Big(\sum_{\tilde{x}} p(\tilde{x}\mid x) - 1\Big),$$

and differentiate with respect to one entry $p(\tilde{x}\mid x)$, at fixed $(x,\tilde{x})$. The one technical point of the derivation — Theorem 4 of Tishby, Pereira, and Bialek (1999) — is that the induced objects must be differentiated too: $p(\tilde{x})$ and $p(\tilde{x},y)$ both move when the encoder does, with $\delta p(\tilde{x})/\delta p(\tilde{x}\mid x) = p(x)$ and $\delta p(\tilde{x},y)/\delta p(\tilde{x}\mid x) = p(x)\,p(y\mid x)$. Carrying both dependences through, the rate term gives

$$\frac{\delta I(X;\tilde{X})}{\delta p(\tilde{x}\mid x)} = \underbrace{p(x)\big[\ln p(\tilde{x}\mid x)+1\big]}_{\text{explicit dependence}} - \underbrace{p(x)\big[\ln p(\tilde{x})+1\big]}_{\text{through } p(\tilde{x})} = p(x)\ln\frac{p(\tilde{x}\mid x)}{p(\tilde{x})},$$

the two dangling $+1$'s cancelling. For the relevance term, expand $I(\tilde{X};Y) = \sum p(\tilde{x},y)\ln p(\tilde{x},y) - \sum_{\tilde{x}} p(\tilde{x})\ln p(\tilde{x}) - \sum_y p(y)\ln p(y)$ and note that the last sum does not involve the encoder at all. The same bookkeeping then yields

$$\frac{\delta I(\tilde{X};Y)}{\delta p(\tilde{x}\mid x)} = p(x)\Big[\sum_y p(y\mid x)\ln p(\tilde{x},y)+1\Big] - p(x)\big[\ln p(\tilde{x})+1\big] = p(x)\sum_y p(y\mid x)\ln p(y\mid \tilde{x}),$$

again with the constants cancelling in pairs. Setting $\delta\mathcal{L}_\lambda/\delta p(\tilde{x}\mid x) = 0$ and dividing out $p(x)$,

$$\begin{aligned}\ln\frac{p(\tilde{x}\mid x)}{p(\tilde{x})} &= \beta\sum_y p(y\mid x)\ln p(y\mid\tilde{x}) \;-\; \tilde{\lambda}(x)\\ &= -\,\beta\underbrace{\sum_y p(y\mid x)\ln\frac{p(y\mid x)}{p(y\mid\tilde{x})}}_{\equiv\; d(x,\tilde{x})} \;+\; \underbrace{\beta\sum_y p(y\mid x)\ln p(y\mid x) - \tilde{\lambda}(x)}_{\text{independent of }\tilde{x}},\end{aligned}$$

where the second step adds and subtracts $\beta\sum_y p(y\mid x)\ln p(y\mid x)$ — a quantity that does not depend on $\tilde{x}$ and can therefore be absorbed into the normalization. This is the central step, because what it isolates is a Kullback–Leibler divergence: the *effective distortion* $d(x,\tilde{x}) = D_{\mathrm{KL}}[p(y\mid x)\,\|\,p(y\mid\tilde{x})]$, measuring how badly the decoder attached to representation $\tilde{x}$ predicts the target compared to the input itself. Nobody chose this distortion; the variation manufactured it from the prediction task. Exponentiating and normalizing, we box the master result:

$$\boxed{\begin{aligned} p(\tilde{x}\mid x) &= \frac{p(\tilde{x})}{Z(x,\beta)}\,e^{-\beta\,d(x,\tilde{x})}, \qquad & d(x,\tilde{x}) &= D_{\mathrm{KL}}[p(y\mid x)\,\|\,p(y\mid\tilde{x})],\\ p(\tilde{x}) &= \sum_x p(x)\,p(\tilde{x}\mid x), \qquad & p(y\mid\tilde{x}) &= \frac{1}{p(\tilde{x})}\sum_x p(y\mid x)\,p(x)\,p(\tilde{x}\mid x),\end{aligned}} \tag{25.2}$$

with

$$Z(x,\beta) = \sum_{\tilde{x}} p(\tilde{x})\,e^{-\beta\,d(x,\tilde{x})}.$$

the information-bottleneck equations (Tishby, Pereira & Bialek 1999)

The optimal soft assignment of input x to representation $\tilde{x}$ is a Boltzmann distribution — the prior $p(\tilde{x})$ tilted by $e^{-\beta d}$ — whose energy is the prediction error of $\tilde{x}$'s decoder on x's conditional, whose temperature $1/\beta$ sets how softly the assignment commits, and whose normalizer $Z(x,\beta)$ is a partition function, one per input. The second and third equations close the system: the marginal is induced by the encoder, and the decoder is the Bayes rule of the encoder. All three are coupled — the energy in the first equation is computed against the decoder of the third, which is built from the encoder of the first — so Equation 25.2 is a fixed point to be reached by iteration.

Before the free-energy reading, one identity confirms that the manufactured distortion is the canonical one. Averaging d over the joint and splitting the logarithm,

$$\langle d(x,\tilde{x})\rangle = \sum_{x,\tilde{x},y} p(x)\,p(\tilde{x}\mid x)\,p(y\mid x)\ln\frac{p(y\mid x)}{p(y\mid\tilde{x})} = I(X;Y) - I(\tilde{X};Y)\,:$$

the mean distortion is *exactly* the relevance lost in compression, nat for nat, so maximizing relevance and minimizing this particular distortion are the same act. With d in hand, the whole Lagrangian acquires a thermodynamic form. A short rearrangement using this identity and $I(X;\tilde{X}) = \sum_x p(x)\,D_{\mathrm{KL}}[p(\tilde{x}\mid x)\|p(\tilde{x})]$ gives, for *any* encoder,

$$\mathcal{L} = \beta\sum_x p(x)\left\{\langle d(x,\tilde{x})\rangle_{p(\tilde{x}|x)} + \tfrac{1}{\beta}\,D_{\mathrm{KL}}\big[p(\tilde{x}\mid x)\,\|\,p(\tilde{x})\big]\right\} \;-\; \beta\,I(X;Y),$$

and the braced quantity is a free energy: mean energy plus temperature times a relative entropy, $F = \langle E\rangle - TS$ with the entropy measured relative to the prior $p(\tilde{x})$. Minimizing it over the encoder of one input, at fixed marginal and decoder, is the Gibbs variational problem of Section 1.4, and its solution and minimum are the ones every partition function announces:

$$f(x,\beta) \;=\; -\frac{1}{\beta}\ln Z(x,\beta) \;=\; \min_{q(\tilde{x})}\Big\{\langle d(x,\tilde{x})\rangle_q + \tfrac{1}{\beta}\,D_{\mathrm{KL}}[q\,\|\,p(\tilde{x})]\Big\}, \tag{25.3}$$

the rate–distortion free energy of one input

attained precisely by the Boltzmann encoder of Equation 25.2. Since $I(X;Y)$ is a constant of the problem, minimizing the IB Lagrangian is minimizing the mean of these per-input free energies, $\bar{F}(\beta) = \sum_x p(x)\,f(x,\beta)$ — the central identity. The Boltzmann distribution of Week 1 is the encoder of Week 13, and representation learning is constrained free-energy minimization.

> **Take-home 2**
>
> The IB Lagrangian is stationary at $p(\tilde{x}\mid x) = \frac{p(\tilde{x})}{Z(x,\beta)}\,e^{-\beta\,d(x,\tilde{x})}$ with $d(x,\tilde{x}) = D_{\mathrm{KL}}[p(y\mid x)\|p(y\mid\tilde{x})]$, coupled to the induced marginal and the Bayes decoder (Equation 25.2). The distortion is manufactured by the variation, not chosen, and its mean equals the relevance lost; the normalizer $Z(x,\beta)$ is a partition function whose free energy $-\frac{1}{\beta}\ln Z$ (Equation 25.3) the optimal encoder minimizes. The optimal representation is the Boltzmann distribution of a rate–distortion free energy.

25.4 A temperature for representation

The two limits of β now read as the two limits of any thermal system. As $\beta \to 0$ — infinite temperature — the exponential in Equation 25.2 flattens toward one, the encoder relaxes onto the prior, $p(\tilde{x}\mid x) \to p(\tilde{x})$ for every input, and a single effective description swallows the whole source: $I(X;\tilde{X}) = I(\tilde{X};Y) = 0$, the origin of the information plane. As $\beta \to \infty$ — zero temperature — the assignment freezes onto the distortion's ground state, each x committing deterministically to its best representation; the encoder becomes a hard partition, the relevance saturates toward $I(X;Y)$, and in the limit $\tilde{X}$ approaches a minimal *sufficient statistic* of X for Y, the coarsest description that predicts as well as the data allow. Between the limits, sweeping β upward traces the IB curve from the compressed end to the faithful end,

and doing the sweep gradually — solving at one β, then warm-starting the next from the converged solution — is *deterministic annealing* (Rose 1998), the cooling of Chapter 16 repurposed from optimization to representation.

The cooling is not smooth, and this is the observation the next lecture will generalize. At high temperature the one-cluster solution is a genuine fixed point of Equation 25.2, and one can ask when it stops being stable: perturbing the uniform encoder and linearizing the update, the perturbation grows once β exceeds a critical value set by the leading eigenvalue of a covariance-like operator built from the conditionals $p(y \mid x)$ — inputs whose conditionals align push the same direction, and the most self-reinforcing direction condenses first (Rose 1998). At that β_c the single description splits in two; further critical temperatures split the daughters in turn; and the number of *distinct* representations in use climbs a staircase as the system cools. These are structural *phase transitions* in the precise sense of Week 5 — a symmetry of the solution breaking at a critical temperature, the order parameter now information-theoretic — and between them the representation is qualitatively rigid: cooling sharpens the assignments but does not reorganize them. On the IB curve the transitions appear as the corners of Figure 25.1, where the frontier's slope changes discontinuously; the worked example below computes the staircase and both of its critical temperatures on a system small enough to watch.

The algorithm falls out of the equations' own structure. Since each of the three relations in Equation 25.2 is the exact optimum given the other two, alternate them: ① update the encoder from the current marginal and decoder by the Boltzmann rule, ② update the marginal from the new encoder, ③ update the decoder by Bayes. Each sweep lowers $\mathcal{L}$ monotonically, and the iteration converges to a fixed point — this is the *Blahut–Arimoto* algorithm of rate–distortion theory, transplanted. The same alternating self-consistency appeared in the mean-field equations (Equation 9.3) and in message passing (Section 12.3); applied here to a rate–distortion free energy, it inherits the standard caveat. The joint problem is not convex, fixed points need not be global minima, and the practical remedy is exactly Week 8's: anneal β from the hot side and let each solution seed the next, so the iteration tracks the branch that has been optimal since high temperature rather than falling into a cold local trap.

25.5 The trainable bottleneck, and two large claims

For anything beyond a small discrete joint, Equation 25.2 cannot be iterated as written: the equations require the conditionals $p(y \mid x)$ exactly, and a continuous $\tilde{X}$ turns the sums into intractable integrals. The escape is the one Week 7 built. The *deep variational information bottleneck* of Alemi et al. (2017) replaces the exact encoder by a neural family $q_\phi(\tilde{x} \mid x)$ and bounds both informations variationally,

$$I(X;\tilde{X}) \;\le\; \mathbb{E}_{p(x)}\, D_{\mathrm{KL}}\big[q_\phi(\tilde{x} \mid x)\,\|\, r(\tilde{x})\big], \qquad I(\tilde{X};Y) \;\ge\; \mathbb{E}\big[\ln q_\psi(y \mid \tilde{x})\big] + H(Y),$$

with $r(\tilde{x})$ a tractable stand-in for the marginal and q_ψ a neural decoder — each bound tight when its variational player matches the true object. Assembling them bounds the Lagrangian by a per-sample loss of the form $-\langle \ln q_\psi(y \mid \tilde{x})\rangle + \frac{1}{\beta}\, D_{\mathrm{KL}}[q_\phi \| r]$: a prediction term plus a weighted KL to a prior, trained end to end with the reparameterization trick. This is Week 7's objective (Equation 14.5) with the reconstruction replaced by prediction — the ELBO, pointed at Y — and it resolves a point left

open in Week 7, where the -VAE's KL weight was flagged as "not an inverse temperature, despite this course's reflexes." Set $Y = X$ in the VIB and it *becomes* the -VAE (Higgins et al. 2017; Alemi et al. 2017), with weight equal to $1/\beta$: the knob is not the inverse temperature but the temperature itself, and the reflex was right after a change of units. Representation learning by deep networks thus rejoins the free-energy thread with every piece identified: the objective is a constrained free energy, the temperature dials compression, and the training loop is amortized variational inference.

The identification licenses two further claims in the literature, and the course states both without endorsing either. The first is the *deep-representation reading*: Tishby and Zaslavsky (2015) proposed the bottleneck as a theory of what deep networks *do* — each layer a stage of compression, training a trajectory in the information plane — and Shwartz-Ziv and Tishby (2017) reported a characteristic two-phase dynamics, a fast fitting phase followed by a long compression phase that they linked to generalization. The claim is attractive and contested. Saxe et al. (2018) found the compression phase absent for widely-used architectures: it appears with saturating nonlinearities such as tanh and disappears with ReLU, depends on the estimator used to measure mutual information in deterministic networks (where $I(X;\tilde{X})$ is, strictly, infinite or constant), and networks were exhibited that generalize without compressing. The *principle* — an optimal representation trades rate against relevance at a temperature — stands on the derivation of Section 25.3; the *dynamical claim* — that stochastic gradient descent implements the trade-off, layer by layer — remains an open empirical question, and the information plane itself is a diagnostic whose reading depends on how it is measured.

The second claim is larger. The *free-energy principle* of Friston (2010) proposes that the variational functional this course has minimized for eight weeks governs not just learning machines but brains and organisms at large: perception is inference (the recognition model of Week 7, with lineage running through the Helmholtz machine of Dayan et al. (1995)), action is the other knob — behave so that predictions come true — and both descend a single variational free energy, the organism's bound on the surprise of its sensory stream. As a unifying language the proposal is broad, which is precisely its standard critique: with enough freedom in the choice of generative model, recognition family, and bound, nearly any observed behavior can be redescribed as free-energy minimization after the fact, and critics have pressed the question of what the principle forbids. We name it as the outer limit of the idea whose inner, checkable cases this course has computed — Jaynes' ensembles, the ELBO, the bottleneck — and leave its status to a literature that is still arguing.

Trap

Four ways to misread the bottleneck. ① *The equations are coupled, and are solved by iteration.* The encoder's energy is a KL computed against the current decoder, which is built from the current encoder; Equation 25.2 is a fixed point (Blahut–Arimoto), the joint problem is nonconvex, and the reliable route is to anneal β from the hot side, as in Chapter 16. ② *The cardinality $|\tilde{X}|$ is a cap, not the number of clusters.* How many representations *may* exist is set by hand; how many are *used* is set by the temperature, and it changes only at the critical β's. ③ *Rate is minimized, relevance maximized.* $I(X;\tilde{X})$ enters $\mathcal{L}$ with a plus sign and $I(\tilde{X};Y)$ with $-\beta$; swapping their roles inverts the curve and both temperature limits. ④ *The principle is sound; the grand*

claims are open. The compression-phase account of deep training is disputed (Saxe et al. 2018), and the free-energy principle is criticized as unfalsifiable — cite both as proposals, not results.

Take-home 3

Sweeping β upward anneals the representation from one cluster (hot) toward a sufficient statistic (cold), through structural phase transitions at critical temperatures where the description splits — Week 5's symmetry breaking on an information-theoretic order parameter, reached by Week 8's cooling. The exact algorithm is mean-field self-consistency (Blahut–Arimoto), and the trainable version (Alemi et al. 2017) is Week 7's ELBO with a predictive target, the -VAE's knob revealed as a temperature. The two extensions — deep networks traverse the information plane while training (Shwartz-Ziv and Tishby 2017), and brains minimize variational free energy (Friston 2010) — are suggestive, contested, and to be held critically (Saxe et al. 2018).

25.6 Example: the IB curve of a twelve-state source

We now put numbers to the mechanism, on a joint small enough that everything is exact. Take $X \in \{1, \dots, 12\}$ uniform and $Y \in \{0, 1, 2\}$, with conditionals $p(y \mid x)$ built from three group profiles — inputs 1–4 lean toward $y = 0$ with profile $(0.6, 0.3, 0.1)$, inputs 5–8 toward $y = 1$ with $(0.3, 0.6, 0.1)$, inputs 9–12 toward $y = 2$ with $(0.1, 0.1, 0.8)$ — plus a small random perturbation (uniform in ± 0.02, fixed seed) so that no two inputs are exactly alike. The design has a deliberate asymmetry: the third group is well separated, while the first two lean toward each other, so the description should split in two stages rather than one. The total information available is $I(X;Y) = 0.286$ nats. We cap the representation at $|\tilde{X}| = 8$ — deliberately more than three, to make the point that the temperature, not the cap, decides how many are used — and run Blahut–Arimoto to convergence at each of sixty-three values of β between 0.1 and 50, warm-starting each solution from the last with a small dose of symmetry-breaking noise: deterministic annealing, exactly as prescribed above.

The left panel of Figure 25.2 is the schematic of Figure 25.1 made quantitative. The frontier rises steeply while relevance is cheap, passes the two-cluster plateau at $(0.59, 0.24)$ — a rate just below the $\ln 3 - \frac{2}{3}\ln 2 = 0.637$ nats of a hard three-way split's first stage, buying 84% of the available relevance — and saturates: at $\beta = 50$ the three-cluster encoder holds $I(\tilde{X};Y) = 0.285$ of the 0.286 nats available, at a rate of $\ln 3$. Eight representation slots were on offer and the solution declines five of them; the cap never binds, because the temperature decides. The right panel shows that decision unfold. The number of representations in use climbs $1 \to 2 \to 3$ at $\beta_c \approx 2.0$ — where the well-separated third group breaks away — and $\beta_c \approx 9.8$, where the two similar groups finally part; between the transitions the count is rigid while the free energy $\bar{F}(\beta)$ (orange) glides down, kinking at each split. Its hot end is a check on Equation 25.3 done by hand: with one cluster the decoder is $p(y)$, the distortion of every input is $D_{\mathrm{KL}}[p(y \mid x) \| p(y)]$, and the mean distortion is exactly $I(X;Y) = 0.286$ nats — the figure's orange curve starts there to the fourth decimal.

Figure 25.3 watches the same physics from inside one conditional. At $\beta = 0.5$ the

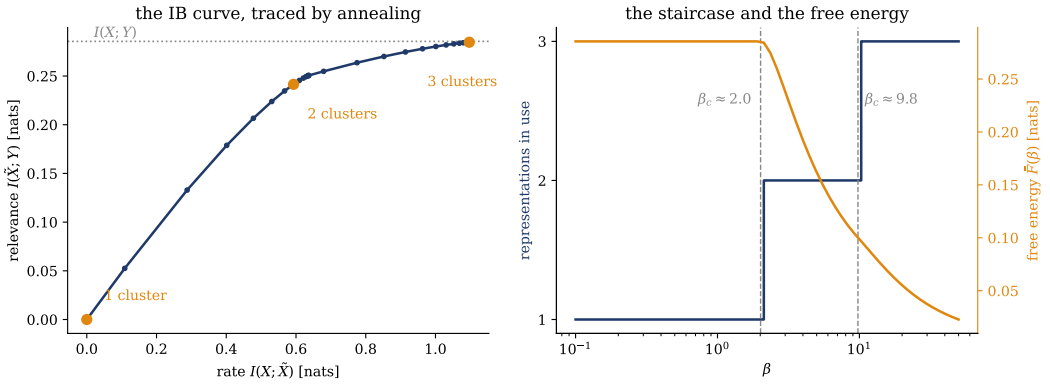

Figure 25.2: The IB curve and its phase transitions, computed exactly by annealed Blahut–Arimoto on the twelve-state toy ($|\tilde{X}| = 8$, sixty-three values of $\beta \in [0.1, 50]$, warm-started; fixed seed). Left: the information plane. The optimal encoders trace the concave frontier from the origin to saturation at $I(X;Y) = 0.286$ nats (dotted); the two-cluster plateau passes through $(0.59, 0.24)$ at $\beta = 4$, and by $\beta = 50$ the three-cluster solution reaches $I(\tilde{X};Y) = 0.285$ nats — 99.7% of the total — at rate $I(X;\tilde{X}) = 1.098 \approx \ln 3$. Right: the number of distinct representations in use (navy, left axis) climbs the staircase $1 \to 2 \to 3$ at the critical temperatures $\beta_c \approx 2.0$ and $\beta_c \approx 9.8$ (dashed), while the annealed free energy $\bar{F}(\beta)$ (orange, right axis) descends from exactly $I(X;Y)$ in the hot limit — where the single decoder is $p(y)$ and the mean distortion is the whole of the information — toward zero, kinking at each transition.

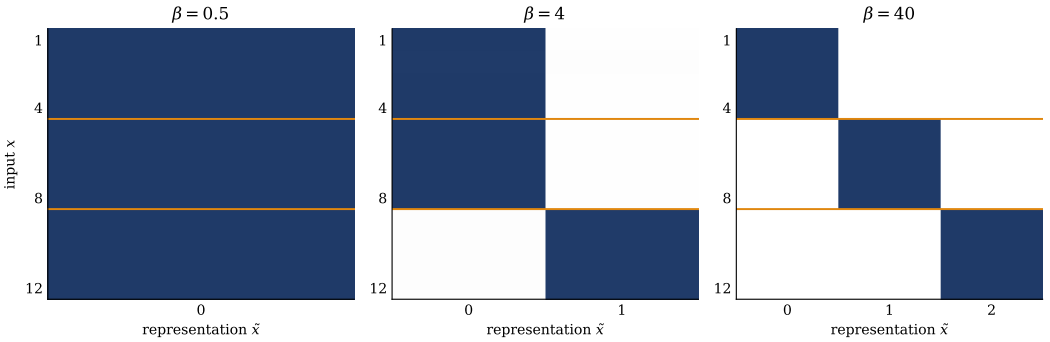

Figure 25.3: The encoder is a Boltzmann distribution sharpening as it cools: the optimal $p(\tilde{x} \mid x)$ of the same sweep, at $\beta = 0.5$, 4, and 40 (left to right; rows are inputs x, columns the representations in use, duplicate slots merged; darker is more probable, orange rules mark the three input groups). Hot, every input relaxes onto the same single description. At $\beta = 4$, past the first transition, the third group has committed to its own representation while the first two share one, a percent-level leak still crossing the divide. Cold, the assignment is a near-hard three-way partition — the sufficient-statistic limit approached from finite temperature.

encoder matrix is a single uniform column — the Boltzmann distribution at high temperature, flat over its states. At $\beta = 4$ the first symmetry has broken: inputs 9–12 occupy their own representation, inputs 1–8 share a second, about one percent of each conditional still leaks across the divide, and the two similar groups are not yet distinguished at all. At $\beta = 40$ the partition is all but hard, three blocks for three groups, and further cooling would only sharpen the edges: within the range plotted, no fourth description condenses, because resolving the ± 0.02 perturbations that distinguish inputs *within* a group carries a distortion gain too small to pay for at these temperatures. The tutorial (16–18, Ph12 106) runs this entire construction where the answer is not transparent: an IB curve computed on a real classifier, with $p(y \mid x)$ a trained network's output rather than a table, and its reorganizations found by the same annealing.

> *A representation is the Boltzmann distribution of a rate–distortion free energy — compression traded against relevance at inverse temperature β — so representation learning is the variational free-energy minimization of Weeks 5 and 7, one module later.*

25.7 Outlook: from one system's transitions to inference at large

The warm-up card covers a hand calculation: the IB Lagrangian for a two-symbol source, minimized to the point where the coupled dependence forces iteration — Tishby, Pereira, and Bialek (1999) §2–3 read against the derivation of Section 25.3.

That lecture generalizes today's most physical observation. The bottleneck's staircase — a representation reorganizing discontinuously at critical values of a control parameter — is not a curiosity of this toy but the signature of a broad and quantitative theory: inference problems at large undergo *phase transitions*. Detecting communities in a network, recovering a planted low-rank signal in a noisy matrix, learning a rule from examples — each has sharp thresholds in signal strength or data volume separating an impossible phase, where no method can succeed, from a recoverable one, and often a further threshold separating what is information-theoretically possible from what any known efficient algorithm achieves. The tools are this course's: partition functions over hypotheses, free energies whose competing minima are "signal found" and "signal lost," and message-passing algorithms (Section 12.3) whose fixed points delimit the algorithmic phase (Zdeborová and Krzakala 2016). The next lecture derives the flagship example and closes with the outlook onto the field's current frontier.

Week 14 then ends the course: Lecture 1 reads every module off a single free-energy functional — the synthesis this week prepares — and the next lecture is review and exam preparation. From the Boltzmann distribution of Week 1 to the encoder of Week 13, one variational principle has recurred with different constraints attached, and the last constraint — an information budget — turned it into a theory of representation.

26 Phase transitions in inference: solvability has a phase diagram

Recommended reading

Core (~1 h). Bahri et al. (2020) §V–VI — the course's backbone review on the theory side: phase transitions in inference, and the random-matrix and spin-glass pictures of learning that close today's outlook. Alongside it, the opening two sections of Zdeborová and Krzakala (2016) — the review written from inside the programme this lecture surveys: the easy/hard/impossible picture, message passing as the algorithmic frontier, and the statistical-to-algorithmic gap. This is the one week whose core reading is a pair of reviews rather than a founding paper; the material is a living research programme, not a settled chapter.

Optional background. Decelle et al. (2011) — the detectability threshold derived today, in the original; the flagship inference phase transition. Baik, Ben Arous, and Péché (2005) — the BBP transition, the random-matrix face of the same story. Krzakala et al. (2013) and Saade, Krzakala, and Zdeborová (2014) — the non-backtracking operator and the Bethe–Hessian: how the physics repaired spectral clustering on sparse graphs (today's numeric uses the latter). Jacot, Gabriel, and Hongler (2018) and Choromanska et al. (2015) — the two outlook pointers, named at the close. Mézard and Montanari (2009) — the textbook home of the whole toolkit.

Prerequisite reminder. From Week 6: belief propagation — the message equation (Equation 12.1), its marginals (Equation 12.2), the Bethe free energy (Equation 12.3), and AMP with its state evolution (Section 12.5). From Week 2: the Hopfield capacity $\alpha_c \approx 0.138$ (Equation 4.6) and the phase diagram of memory (Figure 4.4). From Week 5: the instability of the disordered state at T_c (Section 10.2) and Landau's landscape (Section 10.3). From Week 8: rugged landscapes and metastable traps (Figure 16.1). From the previous lecture: the information bottleneck's representation transitions in β (Section 25.6). New content: inference problems as disordered systems, the stochastic block model and its detectability threshold, the Kesten–Stigum stability argument, the easy/hard/impossible trichotomy and the statistical-to-algorithmic gap, and the named frontier — neural tangent kernels, random-matrix theory, spin-glass loss landscapes.

26.1 The thread becomes the subject

The course has been saying "phase transition" about learning systems since Week 2, and each time the phrase carried real theorems. Hopfield memory, far from degrading gracefully, shatters at a sharp capacity $\alpha_c \approx 0.138$ (Equation 4.6), computed by

the same replica method that solves spin glasses. The mean-field ferromagnet orders at a sharp T_c (Section 10.2), and Landau's expansion (Section 10.3) explained why sharpness is generic. Week 6's message-passing algorithms converge or fail across sharp boundaries of their own. The previous lecture added a subtler instance: as the trade-off parameter β of the information bottleneck is swept, the optimal representation does not deform smoothly but reorganizes at discrete critical values (Section 25.6). Until today these were features of *methods* — of memories, approximations, algorithms we built. Today the thread becomes the subject. The *problems* of inference themselves have phases: tune the signal-to-noise ratio of a statistical-estimation problem and its solvability changes at sharp thresholds, with an order parameter, a symmetry-breaking transition, and — in the most interesting regime — metastability, exactly as in the magnets where this language was invented.

The stance shifts. Modules 1 through 4 used statistical physics to *build* things: energy-based models, variational approximations, samplers, diffusion processes. This lecture uses it to *describe the difficulty of a task* — to compute, for a given inference problem, where recovery of a hidden structure is possible at all, and where it is possible *efficiently*. The object of study is not a new algorithm but the phase diagram of a problem, and the instruments that draw it are precisely the ones this course has already built: belief propagation and its stability (Week 6), the cavity and replica methods (Weeks 2 and 5), approximate message passing and its state evolution (Week 6). This body of work — the *statistical physics of inference*, associated above all with the programme of Krzakala, Zdeborová, and their collaborators (Zdeborová and Krzakala 2016) — is where the course's mean-field toolkit is doing frontier research at the time of writing, and it is the last new object of the semester: next week synthesizes and reviews.

The question, then: *when does a dataset stop carrying its own explanation — and is the boundary set by information, or by computation?*

26.2 Two thresholds, not one

Consider an inference problem with a planted truth — a hidden structure that generated the data — and a signal-to-noise knob. Two different thresholds live on that axis. The *information-theoretic threshold* is where recovery becomes possible in principle: below it, the posterior simply does not know the answer — no method, however slow, exhaustive, or clever, beats random guessing, because the data are statistically indistinguishable from pure noise. The *algorithmic threshold* is where some *efficient* (polynomial-time) method succeeds. The two can coincide, and for the cleanest problem of the day they do. But in general they split, and between them opens a *hard phase*: a region where the answer is present in the data — an exhaustive search would find it — yet no known efficient algorithm can extract it, and where there is mounting evidence that none exists. Figure 26.1 draws the resulting map, which is today's central picture.

The phase boundaries in Figure 26.1 are not metaphors drawn to look thermodynamic; they are computed, and the computation is a free-energy analysis of the posterior. The bridge is one we have crossed repeatedly since Week 3. A posterior distribution over hidden variables is a Boltzmann distribution, $p(\sigma \mid \text{data}) \propto e^{-E(\sigma)}$ with the negative log-likelihood as energy; the data play the role of quenched disorder, fixed while the hidden variables fluctuate — precisely the structure of the spin glasses of Week 2. Inference questions then *are* statistical-mechanics questions.

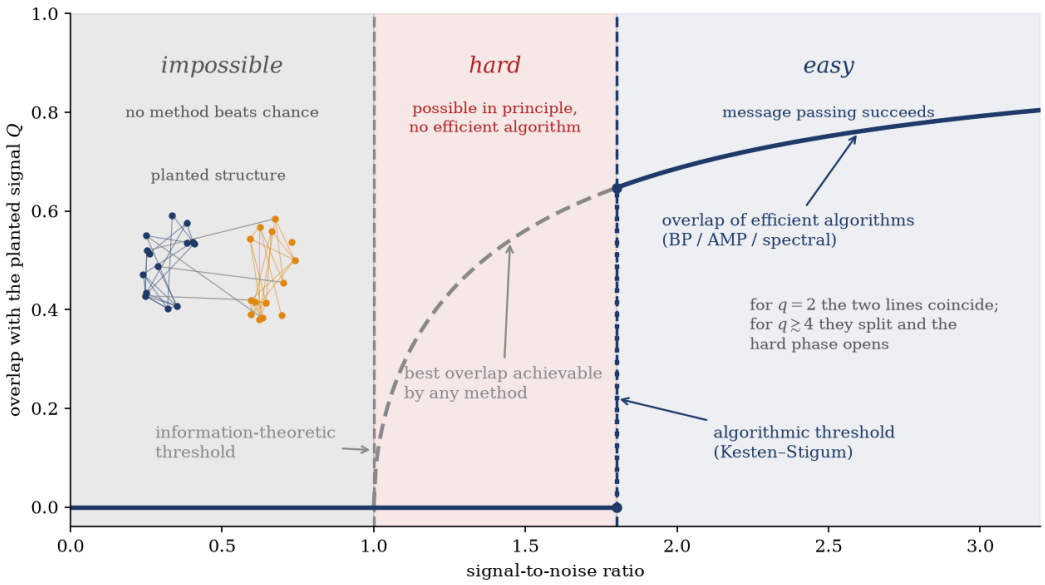

Figure 26.1: Solvability is a phase diagram. Sweeping the signal-to-noise ratio of a planted inference problem, the overlap between the best reconstruction and the hidden truth (the order parameter) passes through three phases: *impossible*, where no method beats chance and the achievable overlap is zero; *hard*, where a positive overlap is achievable in principle (gray dashed) but every known efficient algorithm still returns zero (navy); and *easy*, where message passing and spectral methods reach the optimum. The two boundaries are the information-theoretic and the algorithmic (Kesten–Stigum) thresholds; for the two-group problem solved in this lecture they coincide, and for $q \geq 5$ groups (assortative) they separate, opening the hard phase. Inset: the stochastic block model — two planted communities, dense within, sparse between.

Whether the posterior carries any information about the planted truth is the question of whether an order parameter — the *overlap* between a posterior sample and the truth — is nonzero in the thermodynamic limit; the information-theoretic threshold is a phase boundary of this disordered system, and the algorithmic threshold, as we now derive, is the stability boundary of the dynamics that Week 6 taught us to run on it.

> Take-home 1
>
> Inference problems have phases. With a signal-to-noise ratio as control parameter and the overlap with the planted truth as order parameter, recovery switches sharply between impossible and possible — and there are two thresholds, not one: statistically possible, and achievable by an efficient algorithm. When they separate, a hard phase opens between them: the answer is in the data, and no known polynomial-time method can reach it.

26.3 The engine: the detectability threshold

The derivation belongs to the cleanest planted problem in the field: finding two hidden communities in a sparse random graph. The result — a sharp detectability threshold, located by a two-line stability analysis of Week 6's belief propagation — is due to Decelle et al. (2011), and it is the flagship of the programme.

The model. The *stochastic block model* (SBM) plants a partition and then hides it in randomness. Take n nodes and assign each a hidden label $\sigma_i = \pm 1$, balanced between the two groups. Draw each edge independently: with probability a/n if the endpoints share a label, b/n if they differ. The scaling by $1/n$ keeps the graph *sparse* — the average degree is $c = (a+b)/2$, a constant, as in real networks where a person's friend count does not grow with the population. For $a > b$ the construction plants assortative communities (Figure 26.1, inset). The inference task: given one graph G and the parameters, recover the labels better than a coin flip. The natural measure of success is the *overlap*, $Q = \frac{2}{n}\left|\sum_i \delta_{\hat{\sigma}_i,\sigma_i} - n/2\right|$ in chance-corrected form: $Q = 0$ is random guessing, $Q = 1$ is perfect recovery. A single number controls everything below, and we name it now: the *reduced signal*

$$\lambda = \frac{a-b}{a+b},$$

the contrast between in-group and out-group connectivity. (Notation panel, once: λ is a signal-to-noise parameter throughout this section, not an eigenvalue — although in the spiked-matrix section below the signal strength *will* be an eigenvalue, which is no coincidence. The letter c is the average degree, not a capacity; Week 2's α is retired.)

The posterior is a Boltzmann distribution. Bayes with a flat prior gives $p(\sigma \mid G) \propto P(G \mid \sigma)$, and the likelihood factorizes over pairs — present edges and absent ones both count as evidence:

$$P(G \mid \sigma) = \prod_{(ij)\in E} \frac{c_{\sigma_i\sigma_j}}{n} \prod_{(ij)\notin E} \Big(1 - \frac{c_{\sigma_i\sigma_j}}{n}\Big), \qquad c_{\sigma\sigma'} = \begin{cases} a & \sigma = \sigma', \\ b & \sigma \neq \sigma'. \end{cases}$$

Take the log and use $\sigma_i\sigma_j = \pm 1$ to write each factor in Ising form. For an edge, $\ln c_{\sigma_i\sigma_j} = \frac{1}{2}\ln(ab) + \frac{1}{2}\ln(a/b)\,\sigma_i\sigma_j$: a coupling. For a non-edge, $\ln(1 - c_{\sigma_i\sigma_j}/n) = -c_{\sigma_i\sigma_j}/n + \mathcal{O}(n^{-2})$, which contributes $-\frac{a-b}{2n}\,\sigma_i\sigma_j$ per pair: a coupling as well, weak but acting between *all* pairs. Collecting constants into the normalization,

$$p(\sigma \mid G) = \frac{1}{Z(G)} \exp\left[J \sum_{(ij)\in E} \sigma_i\sigma_j \;-\; \frac{a-b}{2n} \sum_{i<j} \sigma_i\sigma_j \right], \qquad J = \tfrac{1}{2}\ln\frac{a}{b}. \tag{26.1}$$

the SBM posterior: a dilute ferromagnet with a balance constraint

Read the two terms as a physicist. The first is an Ising ferromagnet ($a > b$ makes $J > 0$) living on the *observed graph* — the data are the lattice, which is the quenched disorder of this problem. The second is an infinitesimal all-to-all antiferromagnet; summed over n neighbors it acts as a self-consistent field $-\frac{a-b}{2}\,\bar{m}$ opposing the mean magnetization $\bar{m}$, and its job is bookkeeping: it encodes the knowledge that the groups are balanced, so the posterior must not simply magnetize. Sampling this Boltzmann distribution at temperature 1 *is* Bayesian inference on the SBM. The community structure we seek is a hidden ordering of a dilute magnet, and "can the communities be found?" becomes "does this magnet order along the planted direction?"

The Nishimori simplification. One structural fact makes the analysis tractable. We are inferring with the *same* model that generated the data — the true a, b enter the posterior coupling J. This Bayes-optimal setting places the system on its *Nishimori line*, where identities special to the planted ensemble hold: the planted configuration is statistically indistinguishable from a typical posterior sample, and the overlap between two independent posterior samples equals, on average, the overlap of either with the truth. Critically, on this line the equilibrium free energy is exactly replica-symmetric — the static glass transitions that haunt Week 2's memory at overload cannot occur at Bayes optimality (Zdeborová and Krzakala 2016). That is why the thresholds below are sharp and computable: the replica-symmetric cavity analysis — belief propagation, by Week 6's dictionary — is asymptotically exact here. Metastable states are demoted rather than excluded: they may still trap *dynamics*, and that loophole is exactly where the hard phase will come from.

Belief propagation and the fixed point of ignorance. Equation 26.1 is a pairwise Ising model, so Week 6's machinery applies verbatim: messages $u_{k\to i}$ on the directed edges of G, the message equation Equation 12.1 with $\beta = 1$ and uniform coupling J, cavity fields $h_{i\to j} = \sum_{k\in\partial i\setminus j} u_{k\to i}$, and the weak all-pairs term treated at the level of Week 5's naive mean field — a single global field that pins the total magnetization to zero and otherwise sits out the argument. The sparse graph is locally tree-like — the neighborhood of a typical node contains no loop out to distance $\mathcal{O}(\ln n)$ — so BP is asymptotically exact here, as promised in Week 6 and guaranteed by the Nishimori property.

Whatever the parameters, this BP always admits one particular fixed point: all messages zero, $u_{k\to i} = 0$ on every directed edge — by Equation 12.1, $\tanh(0) = 0$ propagates to itself. At this fixed point every marginal is $1/2$ and the algorithm asserts total ignorance: every node is equally likely in either group. This solution must exist because the model Equation 26.1 is exactly symmetric under a global flip of all labels, so the uninformative state is protected by symmetry at every signal strength, however strong the planted structure. Detection therefore cannot be a question of whether an informative solution *appears*; it is a question of whether the

state of ignorance is *stable*. If a small perturbation of the messages dies out under iteration, BP relaxes back to knowing nothing. If it grows, the symmetric state collapses and the messages flow toward an informative fixed point correlated with the planted labels. We have met this logic twice: it is the instability of $m = 0$ at the Curie point (Section 10.2), and the birth of Landau's ordered minima (Section 10.3), replayed on an inference problem.

Linearize. Perturb around ignorance: small cavity fields $h_{k\to i} = \varepsilon_{k\to i}$. Week 6's message equation linearizes in one step,

$$u_{k\to i} = \operatorname{artanh}\big[\tanh(J)\tanh(\varepsilon_{k\to i})\big] = \tanh(J)\,\varepsilon_{k\to i} + \mathcal{O}(\varepsilon^3),$$

and the bond factor evaluates to something we have already named:

$$\tanh(J) = \tanh\Big(\tfrac{1}{2}\ln\tfrac{a}{b}\Big) = \frac{\sqrt{a/b}-\sqrt{b/a}}{\sqrt{a/b}+\sqrt{b/a}} = \frac{a-b}{a+b} = \lambda.$$

The reduced signal *is* the bond attenuation. Week 6 read the message equation as "a weak bond whispers"; here every bond whispers with the same factor λ, and the linearized recursion for the cavity fields is simply

$$\varepsilon_{i\to j} = \lambda \sum_{k\in\partial i\setminus j} \varepsilon_{k\to i}. \tag{26.2}$$

BP linearized about the uninformative fixed point

Count. Each iteration multiplies a perturbation by λ and branches it through the graph. On the locally tree-like SBM the number of neighbors feeding a message — the excess degree — is Poisson with mean c, so after d generations a disturbance at the root has descended a tree with c^d leaves, each reached along a path of attenuation λ^d. Whether the disturbance survives depends on how the leaf contributions combine, and here sits the one subtle step of the derivation. ① A *coherent* perturbation — all leaves nudged toward + — would return to the root amplified by $(c\lambda)^d$. But that mode is the global magnetization, which the balance field of Equation 26.1 holds at zero, and ordering "everyone in group +" carries no information about the partition anyway. ② The perturbations that matter are the *incoherent* ones the random graph itself seeds: independent, random-signed nudges of the leaves, mean zero. Their sum at the root has zero mean, and the information rides on its variance — c^d independent contributions, each of size λ^d:

$$\boxed{\langle \varepsilon_d^2 \rangle \;\simeq\; (c\lambda^2)^d \,\langle \varepsilon_0^2 \rangle} \tag{26.3}$$

the Kesten–Stigum growth factor

The uninformative fixed point is unstable — a whisper of community structure is amplified rather than forgotten — precisely when $c\lambda^2 > 1$, the *Kesten–Stigum condition*, first found in the theory of multitype branching processes (Kesten and Stigum 1966), later identified as the reconstruction threshold for broadcasts on trees, and rediscovered here as the stability boundary of an algorithm. Substituting $\lambda = (a-b)/(a+b)$ and $c = (a+b)/2$ turns the condition into the parameters of the model:

$$\boxed{(a-b)^2 = 2\,(a+b) \quad\Longleftrightarrow\quad c\,\lambda^2 = 1} \tag{26.4}$$

the detectability threshold (Decelle et al. 2011)

Above this line a positive overlap with the planted labels is achievable, and belief propagation achieves it in linear time; below it, for two balanced groups, the physics analysis of Decelle et al. (2011) concluded that the sparse SBM is information-theoretically indistinguishable from an Erdős–Rényi graph of the same density — the communities are present in the generator and absent from the data — a conjecture since proved rigorously, completing one of the programme's cleanest exchanges between physics and mathematics (Zdeborová and Krzakala 2016). **The detectability threshold is thus the point where the posterior's symmetric state loses stability — a symmetry-breaking phase transition, the mathematics of a magnet ordering, evaluated on an inference problem, with Week 6's algorithm as the order-parameter dynamics.**

The numbers show how sharp the line is. At average degree $c = 3$, the threshold sits at $\lambda = 1/\sqrt{3} \approx 0.577$, i.e. $a \approx 4.73$, $b \approx 1.27$. Now plant communities with $a = 4.5$ and $b = 1.5$: every node has *three times* more neighbors inside its group than outside, a contrast that would be unmistakable in any drawing of the model — and yet $c\lambda^2 = 0.75 < 1$, so no method, spectral, message-passing, or exhaustive, recovers anything. A three-to-one contrast is completely invisible below the line. That is what "the data stop carrying their own explanation" means quantitatively, and Section 26.6 measures it.

> Take-home 2
>
> Two planted communities in a sparse graph are detectable if and only if $(a - b)^2 > 2(a + b)$ (Equation 26.4) — the point where the uninformative fixed point of Week 6's belief propagation goes linearly unstable, with growth factor $c\lambda^2$ (Equation 26.3). Solvability is a symmetry-breaking phase transition in the posterior, and the mean-field toolkit of Module 2 is the instrument that locates it.

26.4 Easy, hard, impossible

For two balanced groups the story ends cleanly: the Kesten–Stigum line is *both* thresholds at once — where efficient detection begins and where any detection begins — and the phase diagram has just two regions, easy and impossible. The general picture is richer, and the enrichment is the field's central discovery. Increase the number of groups q, and the two thresholds separate: for $q \geq 5$ in the symmetric assortative model ($q \geq 4$ in the disassortative / planted-colouring case, and earlier still with unbalanced groups or broad degree distributions), the information-theoretic threshold slides *below* the Kesten–Stigum line (Decelle et al. 2011; Zdeborová and Krzakala 2016). In the window between them, a typical instance contains enough information to recover the communities — the posterior, exhaustively explored, knows the answer — but belief propagation, AMP, spectral methods, and every other known polynomial-time algorithm return noise. This is the *hard phase* of Figure 26.1, and the *statistical-to-algorithmic gap* between the two lines is the object the field now studies most intently, because it appears to be a law about computation itself, discovered with the tools of statistical mechanics.

Why should a possible problem be hard? The physics answer is drawn in Figure 26.2, and it reuses a landscape the course has already built. Compute the Bethe free energy (Equation 12.3) of the posterior as a function of the overlap Q. In the hard phase

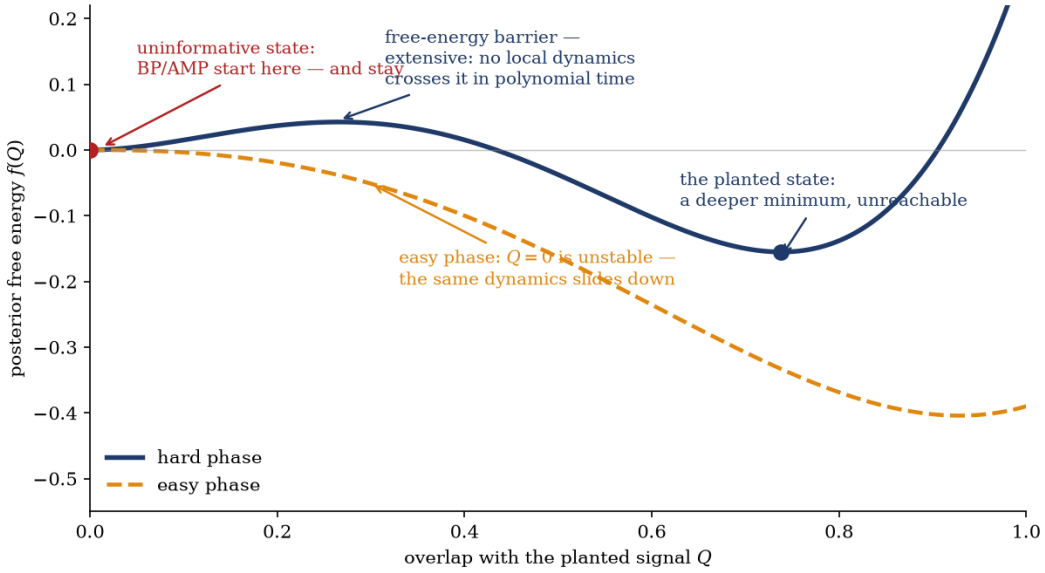

Figure 26.2: Why "hard" is a physics statement. In the hard phase (navy) the posterior free energy over the overlap coordinate has two minima: the uninformative state at $Q = 0$, locally stable, where message passing and every local dynamics begin — and where they stay, because the barrier between the minima is extensive and cannot be crossed in polynomial time. The informative state exists and is thermodynamically favored; it is simply unreachable. In the easy phase (orange, dashed) the barrier is gone, $Q = 0$ is unstable (Equation 26.3), and the same dynamics slides down to the answer. The hard phase is Week 8's metastability, relocated from configuration space to the space of inference states.

it has two local minima: the uninformative state at $Q = 0$, still locally stable since $c\lambda^2 < 1$, and an informative state at $Q > 0$, now *lower* in free energy — the global minimum, which is why exhaustive search succeeds and the information-theoretic threshold has passed. Between them stands an extensive free-energy barrier. Every algorithm we possess for such models is local — BP and AMP iterate small updates, MCMC flips spins, gradient methods take small steps — and a local dynamics started at (or near) ignorance equilibrates into the metastable minimum and stays: the escape time is exponential in n, Week 8's quench-into-a-glass (Figure 16.1) with the annealing schedule powerless because the barrier does not soften. The transition at the information-theoretic threshold is first-order in the Landau sense (Section 10.3): the informative minimum appears at finite Q and crosses below the uninformative one while both remain locally stable, and algorithmic hardness lives exactly in the coexistence window. Hardness, in this picture, is metastability — a statement about the *geometry of the posterior*, not about any particular algorithm's cleverness, which is why the conjecture that no polynomial algorithm beats Kesten–Stigum here has survived a decade of attempts and now anchors average-case complexity theory well outside physics (Zdeborová and Krzakala 2016).

26.5 One transition, many problems

The SBM would be a curiosity if its phase diagram were its own. It is not; the same trichotomy, computed by the same methods, organizes the canonical estimation problems of high-dimensional statistics. We state the two landmark cases result-first — each is a lecture in the source material, and the point here is the pattern.

Spiked matrices and the BBP transition. Plant a rank-one signal in symmetric noise: observe $Y = \lambda vv^{\mathsf{T}} + W$ with v a hidden unit vector, λ the signal strength, and W a Wigner matrix whose spectrum fills the semicircle on $[-2, 2]$. (As promised, the signal strength of this paragraph *is* an eigenvalue.) This is low-rank matrix estimation stripped to its core — the planted ancestor of PCA, factor analysis, and the low-rank decompositions of recommender systems. The spectrum of Y undergoes a sharp transition, first located by Baik, Ben Arous, and Péché (2005) in the sample-covariance setting and carrying their initials (BBP): for $\lambda \le 1$ the signal is swallowed by the bulk and the top eigenvector is asymptotically orthogonal to v — PCA returns pure noise; for $\lambda > 1$ an outlier eigenvalue detaches from the bulk edge at $\lambda + 1/\lambda$ and the leading eigenvector locks onto the signal with squared overlap $1 - \lambda^{-2}$. A rank-one perturbation of vanishing relative norm either vanishes entirely from the spectrum or reorganizes it, with nothing in between. For the symmetric ± 1 signal the three lines — spectral, message-passing, information-theoretic — coincide at $\lambda = 1$; make the signal sparse, and they separate, opening the same hard phase as the many-group SBM, this time in sparse PCA (Zdeborová and Krzakala 2016). AMP (Section 12.5), whose state evolution tracks its own accuracy exactly, is the instrument that draws these Bayes-optimal curves.

Planted clique, the starkest gap. Hide a clique of size k in a uniformly random graph on n nodes. Information-theoretically the clique is detectable once $k \gtrsim 2\log_2 n$, since that is where it begins to exceed the largest clique the noise produces by chance. Every known polynomial-time algorithm, spectral or message-passing or otherwise, needs k of order $\sqrt{n}$. Between a logarithmic and a polynomial clique size stretches the widest hard phase known — not a sliver between nearby thresholds but essentially the entire parameter range — which has made planted

clique the drosophila of average-case hardness: a problem now used as a *primitive*, with the hardness of other tasks established by reduction to it.

The physics also repairs the algorithms. The programme's traffic is not one-way from problems to phase diagrams; the free-energy picture has produced better algorithms. Naive spectral clustering — take the second eigenvector of the adjacency matrix, threshold its signs — fails well above the detectability threshold on sparse graphs, for a concrete reason: with Poisson degrees the largest degree grows like $\ln n / \ln \ln n$, and the top adjacency eigenvectors localize on these hubs (Week 4 met the same pathology in another guise), drowning the community mode. The repair came from the cavity method: the *non-backtracking operator* — the linearization of BP, Equation 26.2, read as a matrix on directed edges, forbidden from immediately retracing its last step — has a clean spectrum whose informative eigenvalues detach exactly at Kesten–Stigum (Krzakala et al. 2013). Its symmetric, $n \times n$ stand-in is the *Bethe–Hessian* $H(r) = (r^2 - 1)\mathbb{1} - rA + D$ with $r = \sqrt{c}$ (here A is the adjacency matrix, D the diagonal degree matrix): literally the Hessian of Week 6's Bethe free energy (Equation 12.3) at the uninformative point, whose negative directions *are* the unstable community modes (Saade, Krzakala, and Zdeborová 2014). The result is a spectral algorithm that works down to the theoretical limit, obtained by differentiating a free energy twice; the numeric below runs it.

26.6 Example: the threshold, measured

The boxed line Equation 26.4 makes a falsifiable claim: plant communities below it and every method returns noise; cross it and a linear-time algorithm finds them. The experiment generates symmetric two-group SBMs at fixed average degree $c = 3$, sweeps the signal-to-noise ratio $\mathrm{SNR} = c\lambda^2 = (a-b)^2/[2(a+b)]$ through 1, recovers labels with the Bethe–Hessian, and measures the chance-corrected overlap — including, at $\mathrm{SNR} = 0.75$, the three-to-one graphs the engine section promised were invisible.

Figure 26.3 confirms the threshold quantitatively. Left panel: below the red line the overlap is indistinguishable from noise at every size — the $a/b = 3$ graphs sit at 0.07 — and the rise beyond it steepens as n grows, the familiar finite-size rounding of a genuine transition (compare the Hopfield fold of Week 2 and the Ising peak of Week 1's tutorial). Below the line there is no algorithmic parameter left to tune; the information is gone from the data, not hidden from the method. Right panel: the gap between the navy and orange curves is the practical payoff of Section 26.5 — two eigenvector computations of identical cost, one guided by the Bethe free energy and one not, separated by a factor of 1.6 in overlap through the transition region.

Trap

Four ways to misquote a detectability threshold. ① *Detection is not exact recovery.* Equation 26.4 is the line for beating chance — a positive overlap. Recovering *every* label correctly is a different, stricter threshold, living in a denser regime (average degree growing like $\ln n$); quoting the constant-degree formula for it is a category error. ② *"Impossible" and "no efficient algorithm" are different lines.* The information-theoretic and algorithmic thresholds coincide for two balanced groups only; conflating them in general erases the hard phase, which is the field's central object. ③ *The uninformative fixed point*

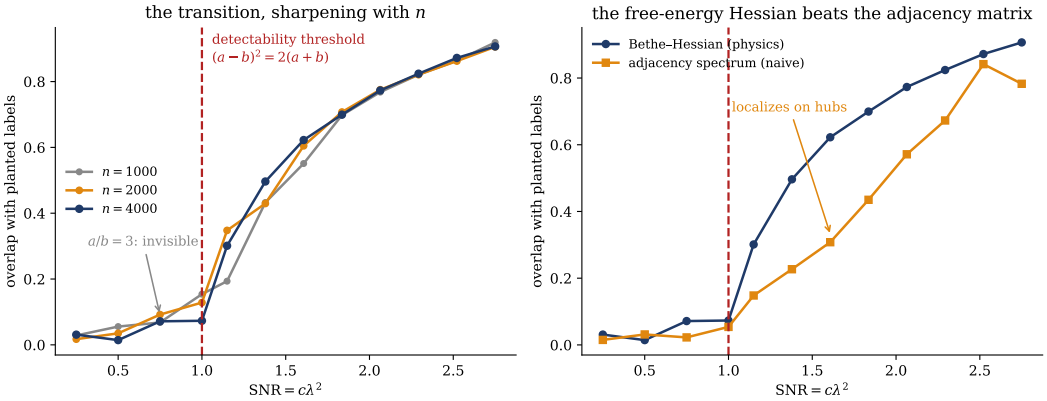

Figure 26.3: The detectability threshold, measured (symmetric two-group SBM, average degree $c = 3$, four graphs per point). Left: chance-corrected overlap between recovered and planted labels for the Bethe–Hessian spectral method (Saade, Krzakala, and Zdeborová 2014), against the signal-to-noise ratio $\mathrm{SNR} = c\lambda^2 = (a-b)^2/[2(a+b)]$. Below the threshold $\mathrm{SNR} = 1$ of Equation 26.4 (red line) the overlap sits at the noise floor — at most 0.07 for $n = 4000$, including the $a/b = 3$ graphs at $\mathrm{SNR} = 0.75$ — and above it the overlap rises continuously from zero. The residual overlap at the threshold itself shrinks with system size (0.16 → 0.07 from $n = 1000$ to 4000): the transition is sharp only as $n \to \infty$, like every phase transition of the course. Right: the physics-derived operator against naive adjacency clustering at $n = 4000$ — near $\mathrm{SNR} \approx 1.8$ the Bethe–Hessian reaches an overlap of 0.70 while the adjacency eigenvector, partially localized on high-degree nodes, delivers 0.44; the naive spectrum needs roughly half again as much signal for the same accuracy.

always exists — detectability is about its stability, not its presence. The symmetric solution is protected by the label-flip symmetry at every SNR; nothing "appears" at the threshold except an instability, cf. Equation 26.3. ④ *The threshold statement is a sparse-regime statement.* $(a-b)^2 = 2(a+b)$ is derived at constant average degree c; in dense scalings the constants and even the questions change. State the regime before quoting the line.

Take-home 3

On sparse two-group SBMs the measured overlap sits at the noise floor below $(a-b)^2 = 2(a+b)$ and rises continuously above it, sharpening with system size — a phase transition observed in an inference problem. The same physics that predicts the line also builds the better algorithm: the Bethe–Hessian (the curvature of Week 6's Bethe free energy at the point of ignorance) detects down to the threshold where the naive adjacency spectrum, localized on hubs, fails. In the multi-group and sparse-signal versions the statistical and algorithmic lines separate, and the gap between them is the frontier.

Inference has phases: below a sharp signal threshold the data hide a structure that no efficient algorithm — and sometimes no method at all — can recover; statistical physics draws that line and characterizes the region between the possible and the efficient.

26.7 Outlook: where the physics is going

The detectability threshold is one exhibit from an active research programme, and this section names three neighbors without deriving them, following Bahri et al. (2020) §V–VI. The *neural tangent kernel* (Jacot, Gabriel, and Hongler 2018) is the infinite-width limit in which a deep network's training linearizes: the network trains as a kernel machine with a kernel fixed at initialization, generalization becomes computable — and the limit simultaneously proves that in this *lazy regime* no features are learned at all, which converts "how do networks learn features?" from a slogan into a sharply posed open problem about finite-width corrections. *Random-matrix theory* — the BBP transition's home discipline — now computes learning curves for high-dimensional regression and classification, including the double-descent phenomenon in which test error peaks at the interpolation threshold and falls again beyond it: generalization behavior organized, once more, by the spectrum of a random matrix. And the *loss landscapes* of deep networks have a spin-glass description: under simplifying assumptions, Choromanska et al. (2015) mapped deep-network losses to the Sherrington–Kirkpatrick-family models of Week 2, with exponentially many critical points stratified by index and the low-loss minima concentrated in a narrow band — a picture whose assumptions are unrealistic and whose qualitative predictions have nonetheless kept matching experiments, in the same spirit as Week 9's flat-minima story. Each of these is the same intellectual move this lecture made on the SBM: take a question about learning, identify the disordered system it secretly is, and compute.

The tutorial (16–18, Ph12 106) belongs to the previous lecture, not this one: the information-bottleneck objective derived as constrained free-energy minimization,

and the IB curve computed for a small classifier, starting from the previous lecture's Lagrangian (Equation 25.1) rather than today's threshold. The IB's representation transitions in β are, in hindsight, the first inference phase transition of the week. And with today, the course's new material ends. Next week's Lecture 1 is the synthesis: the whole semester read as one free-energy calculation, from Hopfield's magnet to the diffusion model's reverse SDE to today's phase diagram of solvability. Next week's Lecture 2 is the exam review. Statistical physics entered machine learning as a supplier of models and methods, but its deepest current contribution runs the other way — it is becoming the theory of *what learning problems are*, and the phase diagram of inference is its first map.

27 Synthesis: thirteen weeks, one variational principle

Recommended reading

Core (light — this lecture derives one identity and reads the course off it). Jaynes (1957) §1–4, re-read. The paper that Week 1 leaned on for the maximum-entropy derivation of Equation 1.2 is also the origin of today's frame: the Boltzmann distribution as the solution of a variational problem, with everything else a consequence. Alongside it, Welling, Lu, and Holdijk (2026) — the companion text tells the same story this lecture tells, from the free-energy face rather than the Z face, and reading the two voices against each other is the best preparation for the exam's translation questions.
Optional (retrospective pointers, not new reading). The Gibbs–Bogoliubov section of Week 5 (Section 9.3) — the engine below was met there as a tool; today it is the frame. The ELBO identity (Equation 9.2) and its VAE incarnation (Equation 14.6). The six-station spine figure of Week 12 (Figure 24.4), which today's central picture completes. Tishby, Pereira, and Bialek (1999) — the seventh stance.
Prerequisite reminder. Everything the course boxed. New content: none — the work of this lecture is to show that the identity $F = -T \log Z$ and the variational free energy $F[q]$ were the hidden subject of all thirteen weeks, and to leave behind the one map the exam is organized around.

27.1 The whole map

Week 1 opened with a thesis: statistical physics supplied machine learning with much of its modeling and inferential machinery, and the course would demonstrate the claim method by method. Thirteen weeks later the demonstration is complete, and it faces the standard risk of any demonstration by cases — that it reads as a catalog. We built energy-based memories, a variational toolbox, samplers, a nonequilibrium theory of generative models, and an information-theoretic coda, and each module closed with its own boxed results. The question a good physics course must answer at the end is whether these are thirteen techniques or one idea worked out thirteen times. The organizing object is the partition function $Z = \sum_x e^{-\beta E(x)}$, easy to write and generally hard to compute. The course's modules are not merely about Z — they are a sequence of stances toward it, and the sequence has a logic. Figure 27.1 draws it, extending the six-station figure that closed Week 12 (Figure 24.4) by the station Module 5 added.

The map also settles a motif the course has used once per module and never yet named as a theme. Four times, a statistical-physics result sat in the literature for decades until machine learning needed it: Onsager's reaction field of 1936 (Onsager

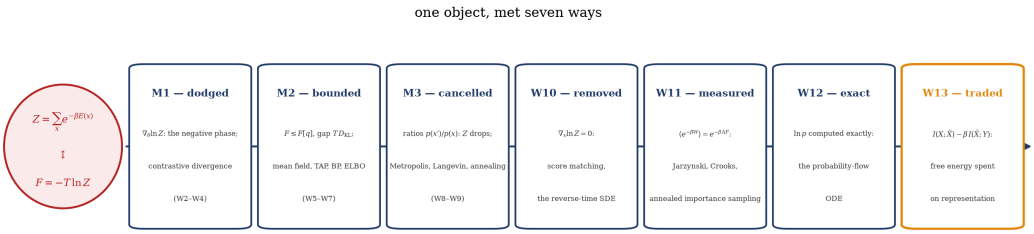

Figure 27.1: One object, met seven ways: the completed spine of the course. Module 1 **dodged** the obstruction — the negative phase $\nabla_\theta \log Z$ of Equation 5.5 blocked maximum-likelihood training, and contrastive divergence sidestepped it with a short chain (Week 3). Module 2 **bounded** it: $F \le F[q]$ for every trial distribution, with mean field, TAP, belief propagation, and the ELBO as four choices of family (Weeks 5–7). Module 3 **cancelled** it inside Metropolis ratios and Fokker–Planck stationary measures, at the price of mixing time (Weeks 8–9). Week 10 **removed** it from the learning problem altogether, since $\nabla_x \log Z = 0$. Week 11 **measured** it, delivering $\log Z$ as a nonequilibrium free-energy difference. Week 12 made the likelihood **exact** through the probability-flow ODE. Week 13 **traded** it: free energy spent on representation under an information constraint. The red ellipse is the claim of Section 27.5 — the two faces at the left are one object.

1936) became the TAP correction of spin-glass theory (Thouless, Anderson, and Palmer 1977) and then the stabilizing term of approximate message passing (Week 6); Bethe's 1935 superlattice approximation (Bethe 1935) became belief propagation's free energy (Yedidia, Freeman, and Weiss 2003); Anderson's 1982 reverse-time diffusion theorem (Anderson 1982) became the generative half of every diffusion model (Week 10); and Jarzynski's 1997 work relation (Jarzynski 1997) became annealed importance sampling (Neal 2001) and, structurally, the diffusion training objective (Weeks 11–12). The pattern is the practical content of the course's thesis: the infrastructure transfers because the *problem* transfers — a Boltzmann distribution with an intractable normalizer is the same mathematical object whether x is a spin configuration or an image, and results proved about the object travel with it.

27.2 Two faces of one object

Before the derivation, we state the claim it will earn. The course has appeared to run on two rails. The narrative rail is Z: the normalization that made Boltzmann-machine training intractable (Equation 5.5), the constant that Metropolis ratios cancel (Equation 15.2), the term whose configuration-gradient vanishes for score matching (Equation 19.1). The methodological rail is the free energy: Module 2 minimized a variational $F[q]$, Week 8 cooled one, Week 11 measured differences of one, Week 13 constrained one. The two rails are connected by an identity the course has carried since its first lecture,

$$\boxed{F \;=\; -T \log Z \;=\; \langle E \rangle - T\,S,} \tag{27.1}$$

the two faces of the spine

The identity is an identification: there was never a second object. Whenever a week led with Z, it was because the obstruction was easiest to point at on that face; whenever a week led with F, it was because the variational structure was easiest to state there.

Week 1 obtained the Boltzmann distribution Equation 1.2 by maximizing the entropy $S = -\sum_x p(x) \log p(x)$ at fixed mean energy (Jaynes 1957) — a variational statement. Since our entropies are in nats ($k_B = 1$), the S of that derivation is simultaneously the information entropy of Week 13; there is no conversion factor anywhere in the course. The whole spine therefore descends from one variational principle, and the engine below is that principle run in reverse: instead of maximizing entropy to *find* p, we minimize free energy to *approach* it.

> Take-home 1
>
> The partition function and the free energy are one object seen two ways, $F = -T \log Z$. The course's modules are seven stances toward it — dodge, bound, cancel, remove, measure, make exact, trade — and Figure 27.1 is the whole course on one line.

27.3 The engine: the variational free energy, promoted

Week 5 met the Gibbs–Bogoliubov bound as the license for mean-field theory, derived it (Equation 9.1), and spent a module exploiting it. What Week 5 could not yet say is what the bound would become: the single inequality that every subsequent method of the course either tightens, saturates, or sidesteps. We therefore derive it once more, as the frame rather than the tool.

A notation panel, once, since the exam spans all of it. The trial distribution is q throughout — always the *variational* object, never a data distribution — and we write $F[q]$ for what Week 5 called F_q, to emphasize that it is a functional on distributions. The letter T is a temperature again (Week 12's use of T as the diffusion horizon is retired), and the course has now used three distinct β's: the thermal $\beta = 1/T$ of every module, the noise schedule β_t of the diffusion weeks, and the information-bottleneck trade-off multiplier β of Week 13. They are cousins — each one prices a constraint — but they are not equal, and confusing them is the fastest route to a wrong exam answer. Sums over x carry over to integrals $\int \mathrm{d}x$ everywhere, as they have since Module 3.

For any distribution $q(x)$, define the *variational free energy* by transplanting the equilibrium formula $F = \langle E \rangle - TS$ onto the trial state:

$$F[q] \;=\; \langle E \rangle_q \;-\; T\,S[q] \;=\; \sum_x q(x)\,E(x) \;+\; T \sum_x q(x) \log q(x).$$

The target of every inference problem in this course is the Boltzmann distribution $p(x) = e^{-\beta E(x)}/Z$ (Equation 1.2). Inverting it expresses the energy in terms of the target, $E(x) = -T\big[\log p(x) + \log Z\big]$, and substituting into $F[q]$ is the entire derivation:

$$
\begin{aligned}
F[q] &= -T\sum_x q(x)\big[\log p(x) + \log Z\big] \;+\; T\sum_x q(x)\log q(x) \\
&= -T\log Z\;\underbrace{\sum_x q(x)}_{=1} + T\sum_x q(x)\log\frac{q(x)}{p(x)} \\
&= -T\log Z \;+\; T\,D_{\mathrm{KL}}(q\,\|\,p).
\end{aligned}
$$

The Kullback–Leibler term is nonnegative — Gibbs' inequality, which deserves its two lines now that it carries the course: since $\log u \le u-1$ with equality only at $u=1$,

$$
-D_{\mathrm{KL}}(q\,\|\,p) \;=\; \sum_x q(x)\log\frac{p(x)}{q(x)} \;\le\; \sum_x q(x)\Big(\frac{p(x)}{q(x)} - 1\Big) \;=\; 1-1 \;=\; 0,
$$

with equality if and only if $q = p$ everywhere. Combining the two displays gives the master result:

$$
\boxed{F[q] \;=\; -T\log Z \;+\; T\,D_{\mathrm{KL}}(q\,\|\,p) \;\ge\; -T\log Z \;=\; F[p], \qquad \text{equality iff } q = p.} \tag{27.2}
$$

the course's equation: Gibbs–Bogoliubov, promoted to the frame

The variational free energy of any trial state sits above the true free energy by exactly T times the information-theoretic distance to the target — the approximation error of a variational method is not vague; it is a KL divergence you can name before you compute anything.

Equation 27.2 admits three readings. Read as a *bound*, it says $\log Z \ge -F[q]/T$: any tractable q yields a certificate on the log-partition function, which is Module 2. Read as an *objective*, it says that minimizing $F[q]$ over a family simultaneously tightens the bound and finds the family member closest to p in KL — the dual role that connects magnetism (Week 5) with Bayesian inference (Week 7). Read as a *limit*, it says the floor is reached only at $q = p$ itself: a method that insists on zero gap must produce the exact Boltzmann distribution, which is what sampling does and why Module 3 exists.

> **Take-home 2**
>
> $F[q] = -T\log Z + T\,D_{\mathrm{KL}}(q\|p) \ge -T\log Z$, with equality iff $q = p$ (Equation 27.2). Minimizing F over a restricted family bounds $\log Z$ and the gap is exactly $T\,D_{\mathrm{KL}}$; saturating the bound is exact sampling; and any method that only ever differentiates $\log p$ with respect to x never sees Z at all. This single line organizes the course.

27.4 The seven-way read-off

Every method the course built is a specialization of Equation 27.2 — a choice of what family q may live in, and of what is held fixed while $F[q]$ is pushed down. Table 27.1 sets them side by side; each circled entry below references the boxed result of its home lecture. It cuts finer than the seven verbs of Figure 27.1 rather

than matching them one-to-one — *bound* fans out into mean field, BP, and the ELBO, and *cancel* into MCMC and annealing — and its first row restores Module 1's contrastive divergence, the lone *dodge* whose gap is a truncation bias rather than a controlled KL. No new mathematics appears below.

Table 27.1: Every method is a choice of family for q and of what is held fixed in $F[q]$; its approximation error is a KL divergence that can be named in advance.

method (home)	the family for q	held fixed	the gap $T\,D_{\mathrm{KL}}(q\|p)$ is
contrastive divergence (W3)	the chain's law after k steps from data	k Gibbs sweeps, data start	a truncation bias — *not* a controlled bound
mean field, TAP (W5–W6)	products $\prod_i q_i(x_i)$	E, T	the discarded correlations
belief propagation (W6)	pair-consistent beliefs on a graph	the factor graph	the loop corrections
ELBO, VAE (W5, W7)	amortized $q_\phi(z \mid x)$	data x, decoder	the posterior + amortization gap
MCMC (W8–W9)	states reachable by a chain in finite time	the target p	transient — it decays at the mixing rate
simulated annealing (W8)	the equilibria p_T along a schedule	E	zero at each rung; the schedule pays instead
score matching, diffusion (W10–W12)	reverse-path measures P_R^θ	the forward noising P_F	the dissipation $\beta(\langle W\rangle - \Delta F)$
information bottleneck (W13)	encoders $q(\tilde{x} \mid x)$	the source $p(x, y)$	relevant information left uncaptured

① **Mean field and TAP (Weeks 5–6).** Restrict q to a product $\prod_i q_i(x_i)$ and minimize: stationarity gives the self-consistency equations Equation 9.3, and the gap $T\,D_{\mathrm{KL}}$ is exactly the correlation structure the factorization cannot carry — largest near criticality, where correlations are long-ranged, which is why mean field fails worst there. TAP (Equation 11.2) does not change the family; it corrects the *functional*, adding the Onsager reaction term as the next order of the Plefka expansion (Plefka 1982), and Week 6's AMP transplanted the same correction to compressed sensing.

② **Belief propagation (Week 6).** Widen the family from single-site products to locally consistent pair beliefs and replace F by its Bethe approximation (Equation 12.3): the stationary points are exactly the fixed points of the message-passing equations (Yedidia, Freeman, and Weiss 2003). On a tree the Bethe free energy is the true F and the gap closes; on loopy graphs the loops are the gap.

③ **The ELBO (Weeks 5 and 7).** Transcribing Equation 27.2 to a Bayesian posterior gives Equation 9.2: the evidence lower bound is $-F[q]$, so maximizing a likelihood bound *is* minimizing a free energy, term for term (Equation 14.6). The VAE's amortized family $q_\phi(z \mid x)$ makes the minimization a single neural-network

pass (Kingma and Welling 2014), and its training curve is a free-energy descent whose remaining height above $-\log p(x)$ is a KL (Equation 14.1).

④ **MCMC (Weeks 8–9).** Refuse any restriction and demand the saturating $q = p$: detailed balance (Equation 15.1) builds a chain whose stationary state is exactly the Boltzmann distribution, with Z cancelling in every acceptance ratio (Equation 15.2). The bound's slack is traded for a new price, the mixing time — the wall did not fall; it moved from bias to variance-per-unit-time. Week 9 read SGD itself in this column: a discretized Langevin equation whose stationary measure (Equation 17.4), *for isotropic noise*, is a Boltzmann distribution on weight space — real SGD noise is anisotropic and generically out of equilibrium (Chaudhari and Soatto (2018)), so the identification is the leading approximation, not an equality.

⑤ **Simulated annealing (Week 8).** Hold the family at the exact equilibria p_T and move the *temperature*: in $F = \langle E \rangle - TS$, high T weights the entropy (broad, easy to sample) and $T \to 0$ concentrates all probability on the energy minima (Equation 16.1) (Kirkpatrick, Gelatt, and Vecchi 1983). The gap is zero at every rung; what costs instead is the schedule, since cooling faster than the mixing time of ④ quenches into metastable states (Figure 16.1).

⑥ **Score matching and diffusion (Weeks 10–12).** Differentiate the logarithm with respect to configuration and Z leaves the problem: $\nabla_x \log p = -\beta \nabla_x E$ (Equation 19.1), the training target that needs no normalizer, and Anderson's reverse SDE (Equation 20.2) turns the learned score into a generator. The variational structure returns on *trajectories*: the DDPM objective (Equation 23.3) is Equation 27.2 written on path space, its gap the dissipation $\beta(\langle W \rangle - \Delta F) = D_{\mathrm{KL}}(P_F \| P_R)$ of the fluctuation theorems (Equation 21.5, via Equation 21.4 and Equation 21.3). The same machinery settles Module 3's debt — Jarzynski/AIS deliver $\log Z$ itself as an average over driven paths (Equation 21.1, Equation 22.3) — and Week 12's probability-flow ODE completes the arc with an exact likelihood (Equation 24.2).

The information bottleneck (Week 13). Constrain the minimization by an information budget: the IB Lagrangian (Equation 25.1) is a free energy with mutual informations in the roles of energy and entropy, its multiplier β a trade-off inverse temperature, and its self-consistent solution a Boltzmann distribution over codewords built on the rate–distortion free energy Equation 25.3. The seventh stance spends free energy on representation rather than generation — the same functional, pointed at a different question.

27.5 Z and F were never two

The read-off resolves two questions that have recurred since Week 1. The first asked whether the spine of the course is the partition function or the variational free-energy principle. The question was malformed: by Equation 27.1 they are the same object, and the course simply used whichever face made the current obstruction visible — Z when the problem was normalization, F when the method was variational. The second asked whether the course has one recurring obstruction or two, since normalization blocks training (Module 1) while sampling hardness blocks estimation (Module 3). These are also one wall seen from two sides: if Z were cheap, p could be normalized and sampled directly; if p were cheap to sample, Z could be estimated by the free-energy methods of Week 11; and both statements are the assertion that $F[q]$ is hard to push to its floor. The bound of Module 2, the sampler of Module 3, and the

removal of Module 4 are three escapes from the same wall, priced respectively in bias (a KL gap), in time (mixing), and in scope (the score gives generation and likelihood, but never a thermodynamic potential unless Week 11's machinery is added).

> *Z and the free energy were never two spines — $F = -T \log Z$, and every method the course built either bounds F over a family, samples the $q = p$ that saturates the bound, or removes Z by differentiating a logarithm. Thirteen weeks were one variational principle.*

27.6 Example: the whole course on one landscape

The synthesis fits in a single executable figure, and we build it from the course's recurring stage set: a bistable energy, the continuous cousin of the two-mode target that Modules 3 and 4 sampled all semester. Take the double well $E(x) = (x^2 - a^2)^2/4$ with $a = 2$, so the wells sit at $x = \pm 2$ and the barrier is 4 (in units of T at $\beta = 1$). Everything below is deterministic grid quadrature — no sampling, so every number is exact to plotting accuracy and the whole cell runs in seconds. We compute the floor $-T \log Z$ once, then put three stances on it: the *bound* of Module 2, evaluated for the family of single Gaussians $q_\theta = \mathcal{N}(\theta, \sigma^2)$ with σ optimized at each mean θ (the moments of a quartic energy under a Gaussian are analytic, so $F[q_\theta]$ needs no integrals at all); the *cooling* of Module 3, as the exact Boltzmann distribution p_T along a temperature ladder; and the *target itself*, which the sampling and score routes reach and the restricted family cannot.

Figure 27.2 is the semester in three panels, and its numbers are worth reading once as a physicist. The left panel is Week 5's projection made quantitative on a target where we can check everything: the best product-family member stops at $F[q^*] = 0.095$, which is 0.734 nats above the floor $-T \log Z = -0.640$, and the decomposition of that gap is exact — a bimodal target approximated by a unimodal family costs at least the $\log 2 = 0.693$ nats of the abandoned mode, and the measured excess of 0.041 nats is the residual shape mismatch between a Gaussian and one quartic well. (Week 5's small-lattice experiment in Section 9.6 found the same anatomy on the Ising model.) The variational curve $F[q_\theta]$ is itself a double well in θ with a maximum at the symmetric point: the restricted family cannot straddle both modes, so it spontaneously breaks the symmetry — the pitchfork of Equation 9.3 and Equation 10.1, reappearing uninvited in a one-parameter fit, and a reminder that the mean-field magnet, the two-cluster posterior of Week 13's block model, and this toy all order for the same reason. The center panel is the annealing mechanism with no algorithm attached: the exact minimizer of $F[q]$ at each temperature is p_T itself, broad and barrier-crossing at $T = 8$, essentially two disconnected wells by $T = 0.5$ — and the vanishing of the between-well weight is exactly why Week 8's chains mix exponentially slowly when quenched. The right panel is what the unrestricted stances deliver: MCMC (Week 8) and the learned score with its reverse SDE (Weeks 10 and 12) both target the navy curve, both wells included, because neither method ever wrote down a family — one saturates the bound by construction, the other bypasses Z by differentiation. The mixture $0.7\,\mathcal{N}(-3, 0.5^2) + 0.3\,\mathcal{N}(3, 0.5^2)$ that served as Week 10's standing target is this panel with unequal wells.

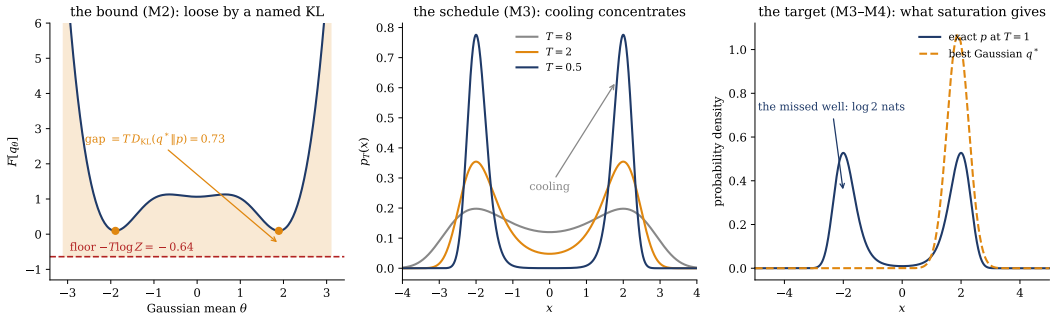

Figure 27.2: The whole course on one free-energy landscape (double well $E(x) = (x^2 - a^2)^2/4$, $a = 2$, barrier $= 4T$ at $T = 1$; deterministic grid quadrature). Left: the variational free energy $F[q_\theta]$ of the best Gaussian $\mathcal{N}(\theta, \sigma^2(\theta))$ against its mean θ, with the exact floor $-T \log Z = -0.640$ (red dashed) and the gap $T\, D_{\mathrm{KL}}(q_\theta \| p)$ shaded. The bound Equation 27.2 holds everywhere; the best single Gaussian ($\theta^* = \pm 1.89$, $\sigma^* = 0.37$) stops 0.734 nats above the floor — $\log 2 = 0.693$ of that gap is the well the ansatz cannot represent, the remaining 0.041 the shape mismatch inside the well it keeps. The variational landscape is itself a double well: the restricted family breaks the symmetry it cannot describe, as mean field did in Week 5. Center: the exact Boltzmann $p_T \propto e^{-E/T}$ at $T = 8, 2, 0.5$ — cooling concentrates the target onto its wells (the barrier-to-well probability ratio falls from 0.61 to 3×10^{-4}), the annealing mechanism of Equation 16.1, with the mixing price of Week 8 hiding in the vanishing weight between the wells. Right: at $T = 1$, the exact bimodal target (navy) against the best single Gaussian (orange dashed) — the panel-left gap made visible as the missing mode, which the sampling route (Week 8) and the score route (Weeks 10 and 12) recover because neither restricts the family.

Trap

Four ways to lose the synthesis on the exam. ① *The bound runs opposite to the ELBO by a sign.* $F[q] \geq -T \log Z$ is *minimized*; the ELBO $= -F[q]$ is *maximized* and lower-bounds the log-evidence (Equation 9.2). They are the same inequality read from opposite faces — keep the convention $p \propto e^{-\beta E}$ fixed and state which face you are on before manipulating signs. ② *The gap is a KL divergence, not hand-waving.* "Mean field is approximate" means precisely $F[q] - F[p] = T\, D_{\mathrm{KL}}(q \| p) \geq 0$, computable in principle and computed above (0.734 nats); an answer that treats the error as unquantifiable has missed the point of Equation 27.2. ③ *Saturation is not free.* Reaching $q = p$ exactly (Module 3) exchanges the bound's bias for mixing time; on the landscape above, the same barrier that starves the between-well weight is what traps the sampler. ④ *Removed is not measured.* Score matching makes Z irrelevant because $\nabla_x \log Z = 0$ (Equation 19.1); it does not deliver $\log Z$. Producing the number took Week 11's work relations (Equation 21.1) — the stances "removed" and "measured" are distinct stations of Figure 27.1, and collapsing them is a category error.

Take-home 3

On one double-well landscape, the variational bound is visibly loose for a single-Gaussian family (a gap of 0.734 nats, of which $\log 2$ is the missed mode), tightens to the floor only at $q = p$, and is walked downhill by cooling T — mean field, annealing, and the exact target are three readings of one figure, and the figure is Equation 27.2 drawn.

27.7 The bridge out: the exam, the frontier, and the title

This week the tutorial machinery changes shape for the first time since Week 1: there is no hard problem and no warm-up card — the stuck-point sequence ended at Week 13 — and both the next lecture and tutorial slots are review. The next lecture is the structured exam preparation: question-driven, with two or three derivations walked end to end in the format the written exam expects, drawing on the accumulated tutorial problems of Weeks 1–13, reused and extended in the exam's pattern. The 16–18 session continues the same work. The exam itself tests computational reasoning on that corpus — interpret a result, predict an output, locate the flaw — and it is organized around exactly the scaffold this lecture boarded: for any method you are handed, say what q it optimizes, what family q is restricted to, what is held fixed, and what the gap $T\, D_{\mathrm{KL}}$ is — then you have located it on Table 27.1 and on Figure 27.1, and the rest is Weeks 2 through 13. That question set is also the recommended architecture for the hand-written aid sheet: the stances table, filled in with your own boxed results, is one page.

Alongside the Z spine, a second thread ran the length of the semester — *phase transitions in learning systems*: the capacity α_c at which memory shatters (Equation 4.6), the T_c at which mean field orders (Equation 10.1), the convergence boundaries of message passing, and finally Week 13's detectability threshold (Equation 26.4), where the thread became the subject. That is the course's live frontier, surveyed in Section 26.7: neural-tangent-kernel and feature-learning theory (Jacot, Gabriel, and

Hongler 2018), random-matrix analyses of learning curves, the spin-glass picture of deep loss landscapes (Choromanska et al. 2015), and the statistical-physics-of-inference programme (Zdeborová and Krzakala 2016). Those pointers, together with the free-energy synthesis of Welling, Lu, and Holdijk (2026), are where a student of this course can start doing research with the toolkit rather than learning it.

The title of the course names the route from Boltzmann to diffusion: the Boltzmann distribution $e^{-\beta E}/Z$ entered as a modeling choice (Week 1, Equation 1.2); its normalizer blocked learning (1985) and was dodged (2002), bounded (1977–2014, from TAP to the VAE), and cancelled (1953); differentiating its logarithm removed it (2005); reversing its relaxation generated data (1982, used 2015–2021); driving it out of equilibrium measured it (1997); a change of variables made its likelihood exact (2021); and constraining its free energy by information turned the same machinery into a theory of representation (1999) and of solvability itself (2011). Every arrow in Figure 27.1 carries such a date, and most of the dates precede the applications by decades. Statistical physics is not an analogy for machine learning but its working infrastructure — and the infrastructure, as Module 5 showed, is still producing.

References

Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. "A Learning Algorithm for Boltzmann Machines." *Cognitive Science* 9 (1): 147–69. https://doi.org/10.1207/s15516709cog0901_7.

Alemi, Alexander A., Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. "Deep Variational Information Bottleneck." In *ICLR 2017*.

Amari, Shun-ichi, and Kenjiro Maginu. 1988. "Statistical Neurodynamics of Associative Memory." *Neural Networks* 1 (1): 63–73. https://doi.org/10.1016/0893-6080(88)90022-6.

Amit, Daniel J. 1989. *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge: Cambridge University Press.

Amit, Daniel J., Hanoch Gutfreund, and Haim Sompolinsky. 1985. "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks." *Physical Review Letters* 55 (14): 1530–33. https://doi.org/10.1103/PhysRevLett.55.1530.

———. 1987. "Statistical Mechanics of Neural Networks Near Saturation." *Annals of Physics* 173 (1): 30–67. https://doi.org/10.1016/0003-4916(87)90092-3.

Anderson, Brian D. O. 1982. "Reverse-Time Diffusion Equation Models." *Stochastic Processes and Their Applications* 12 (3): 313–26. https://doi.org/10.1016/0304-4149(82)90051-5.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate." In *International Conference on Learning Representations (ICLR)*.

Bahri, Yasaman, Jonathan Kadmon, Jeffrey Pennington, Sam S. Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. 2020. "Statistical Mechanics of Deep Learning." *Annual Review of Condensed Matter Physics* 11: 501–28. https://doi.org/10.1146/annurev-conmatphys-031119-050745.

Baik, Jinho, Gérard Ben Arous, and Sandrine Péché. 2005. "Phase Transition of the Largest Eigenvalue for Nonnull Complex Sample Covariance Matrices." *Annals of Probability* 33 (5): 1643–97. https://doi.org/10.1214/009117905000000233.

Baum, Leonard E., Ted Petrie, George Soules, and Norman Weiss. 1970. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains." *The Annals of Mathematical Statistics* 41 (1): 164–71. https://doi.org/10.1214/aoms/1177697196.

Bennett, Charles H. 1976. "Efficient Estimation of Free Energy Differences from Monte Carlo Data." *Journal of Computational Physics* 22 (2): 245–68. https://doi.org/10.1016/0021-9991(76)90078-4.

Bethe, Hans A. 1935. "Statistical Theory of Superlattices." *Proceedings of the Royal Society of London A* 150 (871): 552–75. https://doi.org/10.1098/rspa.1935.0122.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. 2017. "Variational Inference: A Review for Statisticians." *Journal of the American Statistical Association* 112 (518): 859–77. https://doi.org/10.1080/01621459.2017.1285773.

Carreira-Perpiñán, Miguel Á., and Geoffrey E. Hinton. 2005. "On Contrastive Divergence Learning." In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, R5:33–40. Proceedings of Machine Learning Research.

Černý, Vladimír. 1985. "Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm." *Journal of Optimization Theory and Applications* 45 (1): 41–51. https://doi.org/10.1007/BF00940812.

Chaudhari, Pratik, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. 2017. "Entropy-SGD: Biasing Gradient Descent into Wide Valleys." In *International Conference on Learning Representations (ICLR)*.

Chaudhari, Pratik, and Stefano Soatto. 2018. "Stochastic Gradient Descent Performs Variational Inference, Converges to Limit Cycles for Deep Networks." In *International Conference on Learning Representations (ICLR)*.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. "Neural Ordinary Differential Equations." In *NeurIPS*, 31:6572–83.

Choromanska, Anna, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. 2015. "The Loss Surfaces of Multilayer Networks." In *AISTATS 2015, PMLR*, 38:192–204.

Collin, Delphine, Felix Ritort, Christopher Jarzynski, Steven B. Smith, Jr. Tinoco Ignacio, and Carlos Bustamante. 2005. "Verification of the Crooks Fluctuation Theorem and Recovery of RNA Folding Free Energies." *Nature* 437 (7056): 231–34. https://doi.org/10.1038/nature04061.

Crooks, Gavin E. 1999. "Entropy Production Fluctuation Theorem and the Nonequilibrium Work Relation for Free Energy Differences." *Physical Review E* 60 (3): 2721–26. https://doi.org/10.1103/PhysRevE.60.2721.

Dayan, Peter, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. 1995. "The Helmholtz Machine." *Neural Computation* 7 (5): 889–904. https://doi.org/10.1162/neco.1995.7.5.889.

Decelle, Aurélien, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. 2011. "Asymptotic Analysis of the Stochastic Block Model for Modular Networks and Its Algorithmic Applications." *Physical Review E* 84 (6). https://doi.org/10.1103/PhysRevE.84.066106.

Demircigil, Mete, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. 2017. "On a Model of Associative Memory with Huge Storage Capacity." *Journal of Statistical Physics* 168 (2): 288–99. https://doi.org/10.1007/s10955-017-1806-y.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society, Series B* 39 (1): 1–38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Dinh, Laurent, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. 2017. "Sharp Minima Can Generalize for Deep Nets." In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 70:1019–28. PMLR.

Donoho, David L., Arian Maleki, and Andrea Montanari. 2009. "Message-Passing Algorithms for Compressed Sensing." *Proceedings of the National Academy of Sciences* 106 (45): 18914–19. https://doi.org/10.1073/pnas.0909892106.

Edwards, Samuel F., and Philip W. Anderson. 1975. "Theory of Spin Glasses." *Journal of Physics F: Metal Physics* 5 (5): 965–74. https://doi.org/10.1088/0305-4608/5/5/017.

Foret, Pierre, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. "Sharpness-Aware Minimization for Efficiently Improving Generalization." In *International Conference on Learning Representations (ICLR)*.

Friston, Karl J. 2010. "The Free-Energy Principle: A Unified Brain Theory?" *Nature Reviews Neuroscience* 11 (2): 127–38. https://doi.org/10.1038/nrn2787.

Gallager, Robert G. 1962. "Low-Density Parity-Check Codes." *IRE Transactions on Information Theory* 8 (1): 21–28. https://doi.org/10.1109/TIT.1962.1057683.

Gardner, Elizabeth. 1988. "The Space of Interactions in Neural Network Models." *Journal of Physics A: Mathematical and General* 21 (1): 257–70. https://doi.org/10.1088/0305-4470/21/1/030.

Geman, Stuart, and Donald Geman. 1984. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (6): 721–41. https://doi.org/10.1109/TPAMI.1984.4767596.

Georges, Antoine, and Jonathan S. Yedidia. 1991. "How to Expand Around Mean-Field Theory Using High-Temperature Expansions." *Journal of Physics A: Mathematical and General* 24 (9): 2173–92. https://doi.org/10.1088/0305-4470/24/9/024.

Gibbs, J. Willard. 1902. *Elementary Principles in Statistical Mechanics*. New Haven: Yale University Press.

Glauber, Roy J. 1963. "Time-Dependent Statistics of the Ising Model." *Journal of Mathematical Physics* 4 (2): 294–307. https://doi.org/10.1063/1.1703954.

Goldenfeld, Nigel. 1992. *Lectures on Phase Transitions and the Renormalization Group*. Frontiers in Physics. Reading, MA: Addison-Wesley.

Grathwohl, Will, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. 2019. "FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models." In *ICLR*.

Hajek, Bruce. 1988. "Cooling Schedules for Optimal Annealing." *Mathematics of Operations Research* 13 (2): 311–29. https://doi.org/10.1287/moor.13.2.311.

Hastings, W. Keith. 1970. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika* 57 (1): 97–109. https://doi.org/10.1093/biomet/57.1.97.

Hebb, Donald O. 1949. *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

Hertz, John, Anders Krogh, and Richard G. Palmer. 1991. *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley.

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." In *International Conference on Learning Representations (ICLR)*.

Hinton, Geoffrey E. 2002. "Training Products of Experts by Minimizing Contrastive Divergence." *Neural Computation* 14 (8): 1771–1800. https://doi.org/10.1162/089976602760128018.

———. 2012. "A Practical Guide to Training Restricted Boltzmann Machines." In *Neural Networks: Tricks of the Trade*, edited by Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, 2nd ed., 7700:599–619. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/978-3-642-35289-8_32.

Hinton, Geoffrey E., and Drew van Camp. 1993. "Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights." In *Proceedings of the Sixth Annual Conference on Computational Learning Theory (COLT)*, 5–13. https://doi.org/10.1145/168304.168306.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. 2006. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation* 18 (7): 1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. 2006. "Reducing the Dimensionality of Data with Neural Networks." *Science* 313 (5786): 504–7. https:

//doi.org/10.1126/science.1127647.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. "Denoising Diffusion Probabilistic Models." In *Advances in Neural Information Processing Systems*. Vol. 33. https://arxiv.org/abs/2006.11239.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Flat Minima." *Neural Computation* 9 (1): 1–42. https://doi.org/10.1162/neco.1997.9.1.1.

Hopfield, John J. 1982. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences* 79 (8): 2554–58. https://doi.org/10.1073/pnas.79.8.2554.

———. 1984. "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons." *Proceedings of the National Academy of Sciences* 81 (10): 3088–92. https://doi.org/10.1073/pnas.81.10.3088.

Hubbard, John. 1959. "Calculation of Partition Functions." *Physical Review Letters* 3 (2): 77–78. https://doi.org/10.1103/PhysRevLett.3.77.

Hyvärinen, Aapo. 2005. "Estimation of Non-Normalized Statistical Models by Score Matching." *Journal of Machine Learning Research* 6: 695–709.

Jacot, Arthur, Franck Gabriel, and Clément Hongler. 2018. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks." In *NeurIPS*. Vol. 31.

Jarzynski, Christopher. 1997. "Nonequilibrium Equality for Free Energy Differences." *Physical Review Letters* 78 (14): 2690–93. https://doi.org/10.1103/PhysRevLett.78.2690.

———. 2011. "Equalities and Inequalities: Irreversibility and the Second Law of Thermodynamics at the Nanoscale." *Annual Review of Condensed Matter Physics* 2: 329–51. https://doi.org/10.1146/annurev-conmatphys-062910-140506.

Jaynes, Edwin T. 1957. "Information Theory and Statistical Mechanics." *Physical Review* 106 (4): 620–30. https://doi.org/10.1103/PhysRev.106.620.

Jordan, Michael I., Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. "An Introduction to Variational Methods for Graphical Models." *Machine Learning* 37 (2): 183–233. https://doi.org/10.1023/A:1007665907178.

Kappen, Hilbert J., and Francisco B. Rodríguez. 1998. "Efficient Learning in Boltzmann Machines Using Linear Response Theory." *Neural Computation* 10 (5): 1137–56. https://doi.org/10.1162/089976698300017386.

Karras, Tero, Miika Aittala, Timo Aila, and Samuli Laine. 2022. "Elucidating the Design Space of Diffusion-Based Generative Models." In *NeurIPS*. Vol. 35.

Keskar, Nitish Shirish, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima." In *International Conference on Learning Representations (ICLR)*.

Kesten, Harry, and Bernt P. Stigum. 1966. "Additional Limit Theorems for Indecomposable Multidimensional Galton-Watson Processes." *Annals of Mathematical Statistics* 37 (6): 1463–81. https://doi.org/10.1214/aoms/1177699139.

Kingma, Diederik P., and Max Welling. 2014. "Auto-Encoding Variational Bayes." In *International Conference on Learning Representations (ICLR)*.

———. 2019. "An Introduction to Variational Autoencoders." *Foundations and Trends in Machine Learning* 12 (4): 307–92. https://doi.org/10.1561/2200000056.

Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671–80. https://doi.org/10.1126/science.220.4598.671.

Krotov, Dmitry, and John J. Hopfield. 2016. "Dense Associative Memory for

Pattern Recognition." In *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 29.

———. 2021. "Large Associative Memory Problem in Neurobiology and Machine Learning." In *International Conference on Learning Representations (ICLR)*.

Krzakala, Florent, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. 2013. "Spectral Redemption in Clustering Sparse Networks." *Proceedings of the National Academy of Sciences* 110 (52): 20935–40. https://doi.org/10.1073/pnas.1312486110.

Landau, Lev D. 1937. "On the Theory of Phase Transitions." *Zhurnal Eksperimentalnoi i Teoreticheskoi Fiziki* 7: 19–32.

Langevin, Paul. 1908. "Sur La Théorie Du Mouvement Brownien." *Comptes Rendus de l'Académie Des Sciences* 146: 530–33.

Liphardt, Jan, Sophie Dumont, Steven B. Smith, Jr. Tinoco Ignacio, and Carlos Bustamante. 2002. "Equilibrium Information from Nonequilibrium Measurements in an Experimental Test of Jarzynski's Equality." *Science* 296 (5574): 1832–35. https://doi.org/10.1126/science.1071152.

Lipman, Yaron, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. "Flow Matching for Generative Modeling." In *ICLR*.

Little, William A. 1974. "The Existence of Persistent States in the Brain." *Mathematical Biosciences* 19 (1–2): 101–20. https://doi.org/10.1016/0025-5564(74)90031-5.

MacKay, David J. C. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press.

Mandt, Stephan, Matthew D. Hoffman, and David M. Blei. 2017. "Stochastic Gradient Descent as Approximate Bayesian Inference." *Journal of Machine Learning Research* 18 (134): 1–35.

McCulloch, Warren S., and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5: 115–33. https://doi.org/10.1007/BF02478259.

McEliece, Robert J., Edward C. Posner, Eugene R. Rodemich, and Santosh S. Venkatesh. 1987. "The Capacity of the Hopfield Associative Memory." *IEEE Transactions on Information Theory* 33 (4): 461–82. https://doi.org/10.1109/TIT.1987.1057328.

Mehta, Pankaj, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. 2019. "A High-Bias, Low-Variance Introduction to Machine Learning for Physicists." *Physics Reports* 810: 1–124. https://doi.org/10.1016/j.physrep.2019.03.001.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. "Equation of State Calculations by Fast Computing Machines." *Journal of Chemical Physics* 21 (6): 1087–92. https://doi.org/10.1063/1.1699114.

Mézard, Marc, and Andrea Montanari. 2009. *Information, Physics, and Computation*. Oxford: Oxford University Press.

Mézard, Marc, Giorgio Parisi, and Miguel A. Virasoro. 1987. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*. Singapore: World Scientific.

Neal, Radford M. 2001. "Annealed Importance Sampling." *Statistics and Computing* 11 (2): 125–39. https://doi.org/10.1023/A:1008923215028.

Nichol, Alexander Quinn, and Prafulla Dhariwal. 2021. "Improved Denoising Diffusion Probabilistic Models." In *Proceedings of the 38th International Conference on Machine Learning (ICML), PMLR 139*, 8162–71.

Nishimori, Hidetoshi. 2001. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford: Oxford University Press.

Onsager, Lars. 1936. "Electric Moments of Molecules in Liquids." *Journal of the American Chemical Society* 58 (8): 1486–93. https://doi.org/10.1021/ja01299a050.

———. 1944. "Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition." *Physical Review* 65 (3–4): 117–49. https://doi.org/10.1103/PhysRev.65.117.

Opper, Manfred, and David Saad, eds. 2001. *Advanced Mean Field Methods: Theory and Practice*. Cambridge, MA: MIT Press.

Parisi, Giorgio. 1979. "Infinite Number of Order Parameters for Spin-Glasses." *Physical Review Letters* 43 (23): 1754–56. https://doi.org/10.1103/PhysRevLett.43.1754.

Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann.

Plefka, Timm. 1982. "Convergence Condition of the TAP Equation for the Infinite-Ranged Ising Spin Glass Model." *Journal of Physics A: Mathematical and General* 15 (6): 1971–78. https://doi.org/10.1088/0305-4470/15/6/035.

Ramsauer, Hubert, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, et al. 2021. "Hopfield Networks Is All You Need." In *International Conference on Learning Representations (ICLR)*.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. 2014. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 32:1278–86. PMLR.

Risken, Hannes. 1989. *The Fokker–Planck Equation: Methods of Solution and Applications*. 2nd ed. Vol. 18. Springer Series in Synergetics. Berlin: Springer.

Rose, Kenneth. 1998. "Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems." *Proceedings of the IEEE* 86 (11): 2210–39. https://doi.org/10.1109/5.726788.

Saade, Alaa, Florent Krzakala, and Lenka Zdeborová. 2014. "Spectral Clustering of Graphs with the Bethe Hessian." In *NeurIPS*, 27:406–14.

Salakhutdinov, Ruslan, and Iain Murray. 2008. "On the Quantitative Analysis of Deep Belief Networks." In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 872–79. ACM. https://doi.org/10.1145/1390156.1390266.

Saxe, Andrew M., Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. 2018. "On the Information Bottleneck Theory of Deep Learning." In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.1088/1742-5468/ab3985.

Schwarz, Ulrich S. 2023. "Theoretical Statistical Physics." Lecture-notes script, Heidelberg University.

Sethna, James P. 2021. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*. 2nd ed. Oxford: Oxford University Press.

Shannon, Claude E. 1948. "A Mathematical Theory of Communication." *Bell System Technical Journal* 27: 379–423, 623–56. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Sherrington, David, and Scott Kirkpatrick. 1975. "Solvable Model of a Spin-Glass." *Physical Review Letters* 35 (26): 1792–96. https://doi.org/10.1103/PhysRevLett.35.1792.

Shwartz-Ziv, Ravid, and Naftali Tishby. 2017. "Opening the Black Box of Deep

Neural Networks via Information."

Smolensky, Paul. 1986. "Information Processing in Dynamical Systems: Foundations of Harmony Theory." In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, edited by David E. Rumelhart and James L. McClelland, 194–281. Cambridge, MA: MIT Press.

Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics." In *Proceedings of the 32nd International Conference on Machine Learning*, 37:2256–65. Proceedings of Machine Learning Research. https://arxiv.org/abs/1503.03585.

Song, Yang, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. "Maximum Likelihood Training of Score-Based Diffusion Models." In *NeurIPS*. Vol. 34.

Song, Yang, and Stefano Ermon. 2019. "Generative Modeling by Estimating Gradients of the Data Distribution." In *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32.

Song, Yang, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. 2020. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation." In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference (UAI)*, 115:574–84. PMLR.

Song, Yang, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. "Score-Based Generative Modeling Through Stochastic Differential Equations." In *International Conference on Learning Representations*. https://arxiv.org/abs/2011.13456.

Stratonovich, Ruslan L. 1957. "On a Method of Calculating Quantum Distribution Functions." *Soviet Physics Doklady* 2: 416–19.

Sutskever, Ilya, and Tijmen Tieleman. 2010. "On the Convergence Properties of Contrastive Divergence." In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:789–95. Proceedings of Machine Learning Research.

Thouless, David J., Philip W. Anderson, and Richard G. Palmer. 1977. "Solution of 'Solvable Model of a Spin Glass'." *Philosophical Magazine* 35 (3): 593–601. https://doi.org/10.1080/14786437708235992.

Tieleman, Tijmen. 2008. "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient." In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 1064–71. https://doi.org/10.1145/1390156.1390290.

Tipping, Michael E., and Christopher M. Bishop. 1999. "Probabilistic Principal Component Analysis." *Journal of the Royal Statistical Society, Series B* 61 (3): 611–22. https://doi.org/10.1111/1467-9868.00196.

Tishby, Naftali, Fernando C. Pereira, and William Bialek. 1999. "The Information Bottleneck Method." In *Proc. 37th Allerton Conference on Communication, Control, and Computing*, 368–77.

Tishby, Naftali, and Noga Zaslavsky. 2015. "Deep Learning and the Information Bottleneck Principle." In *2015 IEEE Information Theory Workshop (ITW)*, 1–5. https://doi.org/10.1109/ITW.2015.7133169.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems (NeurIPS)*, 30:6000–6010.

Vincent, Pascal. 2011. "A Connection Between Score Matching and Denoising Autoencoders." *Neural Computation* 23 (7): 1661–74. https://doi.org/10.1162/

neco_a_00142.

Wainwright, Martin J., and Michael I. Jordan. 2008. "Graphical Models, Exponential Families, and Variational Inference." *Foundations and Trends in Machine Learning* 1 (1–2): 1–305. https://doi.org/10.1561/2200000001.

Weiss, Pierre. 1907. "L'hypothèse Du Champ Moléculaire Et La Propriété Ferromagnétique." *Journal de Physique Théorique Et Appliquée* 6 (1): 661–90. https://doi.org/10.1051/jphystap:019070060066100.

Welling, Max, Sirui Lu, and Lars Holdijk. 2026. *Generative AI and Stochastic Thermodynamics: A Tale of Free Energies*. Cambridge: Cambridge University Press. https://doi.org/10.1017/9781009709071.

Welling, Max, and Yee Whye Teh. 2011. "Bayesian Learning via Stochastic Gradient Langevin Dynamics." In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 681–88.

Yedidia, Jonathan S., William T. Freeman, and Yair Weiss. 2003. "Understanding Belief Propagation and Its Generalizations." In *Exploring Artificial Intelligence in the New Millennium*, edited by Gerhard Lakemeyer and Bernhard Nebel, 239–69. San Francisco: Morgan Kaufmann.

Yuille, Alan L., and Anand Rangarajan. 2003. "The Concave-Convex Procedure." *Neural Computation* 15 (4): 915–36. https://doi.org/10.1162/08997660360581958.

Zdeborová, Lenka, and Florent Krzakala. 2016. "Statistical Physics of Inference: Thresholds and Algorithms." *Advances in Physics* 65 (5): 453–552. https://doi.org/10.1080/00018732.2016.1211393.

Zwanzig, Robert W. 1954. "High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases." *The Journal of Chemical Physics* 22 (8): 1420–26. https://doi.org/10.1063/1.1740409.