MATH 285J: Topics in High-Dimensional Sampling and Generative Diffusions

Lecture and Personal Notes
(please use carefully as it is a rough draft;
literature and perspectives included not comprehensive,
and any mistakes and typos please let us know!)

Yifan Chen
UCLA, Department of Mathematics

May 2026

Contents

1 Course Goals

The central problem of this course is: *how do we sample from a target distribution* $\pi \propto e^{-V}$ *on* $\mathbb{R}^d$ *and compute statistics* $\mathbb{E}_\pi[h]$*, efficiently in high dimensions?* We assume access to π either via (1) **query access** to V or ∇V (with unknown normalization $Z = \int e^{-V}\, dx$), or (2) **data access** via existing samples $X_1, \ldots, X_N \sim \pi$. Motivating settings include:

- **Bayesian inverse problems**: given $y = F(\theta) + \xi$ where $\xi \sim \mathcal{N}(0, \sigma^2 I_d)$ and a prior $p_{\text{prior}}(\theta)$, sample the posterior $p_{\text{post}}(\theta) \propto p_{\text{prior}}(\theta)\, e^{-\|y - F(\theta)\|^2/(2\sigma^2)}$.
- **Statistical physics**: Gibbs measure $\pi \propto e^{-V/T}$ with potential V and temperature T; partition function $Z = \int e^{-V/T}\, dx$ is intractable.
- **Generative modeling**: π is an unknown data distribution accessible only through samples.
- **Reinforcement learning**: V relates to reward functions.

Course outline and references

- **Static methods**: rejection/importance sampling, curse of dimensionality.
- **Dynamical methods**: MCMC, Langevin SDEs, gradient flow interpretation, mixing time and bias, acceleration by kinetic Langevin, preconditioning by affine invariant methods, de-bias via Metropolis-Hasting.
- **Flow and diffusion methods**: diffusion models, flow matching (ODEs/SDEs), posterior sampling with generative priors, stochastic optimal control, variational inference, and discrete diffusion.

References: Sanz-Alonso & Al-Ghattas (Introduction to Monte Carlo) [61]; Chewi (Algorithmic Complexity of Log-concave Sampling) [22]; Liu (Monte Carlo Strategies in Scientific Computing) [51]; Sanz-Alonso, Stuart & Taeb (Inverse Problems and Data Assimilation) [62]; various diffusion/flow papers.

2 Static Methods: Rejection and Importance Sampling

We start with the setting where we have query access V or its gradients. Namely, we know the density up to normalization constants.

2.1 Generating Samples in One Dimension

We assume throughout that $U \sim \mathrm{Unif}(0,1)$ can be generated. This suffices to sample any one-dimensional distribution via the following standard fact.

Proposition 2.1 (Inverse CDF method)**.** *Let π be a probability distribution on $\mathbb{R}$ with CDF $F(x) = \mathbb{P}_\pi(X \le x)$. If $U \sim \mathrm{Unif}(0,1)$, then $X = F^{-1}(U) \sim \pi$, where $F^{-1}(u) = \inf\{x : F(x) \ge u\}$.*

Proof. $\mathbb{P}(F^{-1}(U) \le x) = \mathbb{P}(U \le F(x)) = F(x)$. □

This gives Gaussian, exponential, and other standard distributions. For general high-dimensional targets, the CDF is intractable and more powerful methods are needed.

2.2 Rejection Sampling

Setup. Let $\pi, g : \mathbb{R}^d \to [0,\infty)$ be probability densities with $\mathrm{supp}(\pi) \subseteq \mathrm{supp}(g)$, and fix $M \ge \sup_{x\in\mathbb{R}^d} \pi(x)/g(x) < \infty$.

Algorithm. Repeat until acceptance:

(1) Draw $Y \sim g$ and $U \sim \mathrm{Unif}(0,1)$ independently.

(2) Compute $r(Y) = \pi(Y)/(Mg(Y)) \in [0,1]$.

(3) If $U \le r(Y)$, return $X = Y$. Otherwise go to (1).

Proposition 2.2 (Correctness)**.** *The returned sample satisfies $X \sim \pi$.*

Proof. For any measurable $A \subseteq \mathbb{R}^d$,

$$\mathbb{P}(Y \in A \mid Y \text{ accepted}) = \frac{\int_A g(y)\, r(y)\, dy}{\int_{\mathbb{R}^d} g(y)\, r(y)\, dy} = \frac{\frac{1}{M}\int_A \pi(y)\, dy}{\frac{1}{M}} = \int_A \pi(y)\, dy.$$

□

The acceptance probability is $\mathbb{P}(\text{accept}) = 1/M$, so the waiting time follows $\mathrm{Geo}(1/M)$ with mean M. Given N i.i.d. proposals $Y_1, \dots, Y_N \sim g$ and $U_1, \dots, U_N \sim \mathrm{Unif}(0,1)$, with $r_i = \pi(Y_i)/(Mg(Y_i))$, the *RS estimator* of $\mathbb{E}_\pi[h]$ is

$$I^{\mathrm{RS}}_{\pi,N}(h) = \frac{\sum_{i=1}^{N} \mathbf{1}\{U_i \le r_i\}\, h(Y_i)}{\sum_{i=1}^{N} \mathbf{1}\{U_i \le r_i\}}. \tag{2.1}$$

Remark 2.3. Rejected proposals are completely discarded. There is no mechanism for moving bad samples toward π; one simply waits for a good draw. When M is large—as it inevitably is in high dimensions—this is very wasteful.

2.3 Importance Sampling

Known target. The key identity is

$$\mathbb{E}_\pi[h] = \int h(x)\,\pi(x)\,dx = \mathbb{E}_g\left[h(Y)\,\frac{\pi(Y)}{g(Y)}\right].$$

Setting $w_i = \pi(Y_i)/g(Y_i)$ and using $\mathbb{E}_g[w_i] = 1$, the *self-normalized IS estimator* is

$$I^{\mathrm{IS}}_{\pi,N}(h) = \frac{\displaystyle\sum_{i=1}^{N} w_i\,h(Y_i)}{\displaystyle\sum_{i=1}^{N} w_i}. \tag{2.2}$$

Unknown normalization (AIS). In practice $\pi = \tilde{\pi}/Z$ with $\tilde{\pi} = e^{-V}$ computable but Z unknown. The unnormalized weights $\tilde{w}_i = \tilde{\pi}(Y_i)/g(Y_i)$ give the *auto-normalized IS (AIS) estimator*

$$I^{\mathrm{AIS}}_{\pi,N}(h) = \frac{\displaystyle\sum_{i=1}^{N} \tilde{w}_i\,h(Y_i)}{\displaystyle\sum_{i=1}^{N} \tilde{w}_i}, \tag{2.3}$$

where Z cancels in the ratio.

Remark 2.4. Unlike RS, IS assigns every proposal a weight rather than accepting or rejecting it. A "bad" proposal (small π/g) contributes little but is not wasted. The cost is high weight variance when π and g differ significantly.

2.4 Asymptotic Theory

Both estimators are ratios of sample averages of i.i.d. variables. Their asymptotics follow from the multivariate CLT and the delta method for $(a, b) \mapsto a/b$.

Theorem 2.5 (CLT for RS and AIS)**.** *Let $h \in L^2(\pi)$. As $N \to \infty$,*

$$\sqrt{N}\big(I^{\mathrm{RS}}_{\pi,N}(h) - \mathbb{E}_\pi[h]\big) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\, G^2_{\mathrm{RS}}), \tag{2.4}$$

$$\sqrt{N}\big(I^{\mathrm{AIS}}_{\pi,N}(h) - \mathbb{E}_\pi[h]\big) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\, G^2_{\mathrm{IS}}), \tag{2.5}$$

where

$$G_{\mathrm{RS}}^2 = M \int_{\mathbb{R}^d} \pi(x) \left|h(x) - \mathbb{E}_\pi[h]\right|^2 dx, \tag{2.6}$$

$$G_{\mathrm{IS}}^2 = \int_{\mathbb{R}^d} \frac{\pi(x)^2}{g(x)} \left|h(x) - \mathbb{E}_\pi[h]\right|^2 dx. \tag{2.7}$$

Remark 2.6. The factor M in G_{RS}^2 is the price of rejection: only a fraction $1/M$ of proposals contribute, inflating the variance by M.

Proposition 2.7 (Importance beats rejection)**.** $G_{\mathrm{IS}}^2 \le G_{\mathrm{RS}}^2$.

See Exercise 1(c) for the proof via Cauchy–Schwarz.

2.5 Non-Asymptotic Bound and the Role of χ^2

The CLT describes large-N fluctuations but does not say how large N must be. The following finite-N bound makes the dependence on N explicit and non-asymptotic.

Theorem 2.8 (Non-asymptotic MSE bound)**.** *For* $\|h\|_\infty \le 1$,

$$\mathbb{E}\Big[\big(I_{\pi,N}^{\mathrm{AIS}}(h) - \mathbb{E}_\pi[h]\big)^2\Big] \le \frac{4}{N}\big(\chi^2(\pi \,\|\, g) + 1\big), \tag{2.8}$$

where $\chi^2(\pi \,\|\, g) = \int \pi(x)^2/g(x)\,dx - 1$.

To achieve MSE $\le \varepsilon^2$, one needs $N \gtrsim (\chi^2(\pi \,\|\, g)+1)/\varepsilon^2$. Since $G_{\mathrm{IS}}^2 \le 4(\chi^2(\pi \,\|\, g)+1)$ for $\|h\|_\infty \le 1$, the χ^2-divergence controls both the asymptotic and non-asymptotic efficiency of AIS. See the next section for more definitions and discussions of χ^2 and KL.

2.6 The Curse of Dimensionality

Example 2.9 (Product measures)**.** *Let* $\pi = \pi_1^{\otimes d}$, $g = g_1^{\otimes d}$, $M_1 = \sup_{x\in\mathbb{R}} \pi_1(x)/g_1(x)$, $c_1 = \chi^2(\pi_1 \,\|\, g_1)$. *Then*

$$\sup_{x\in\mathbb{R}^d} \frac{\pi(x)}{g(x)} = M_1^d, \qquad \chi^2(\pi \,\|\, g) = (1 + c_1)^d - 1.$$

Both grow exponentially *in d: RS needs M_1^d proposals per accepted sample, and AIS needs $N \sim (1 + c_1)^d/\varepsilon^2$ samples to achieve MSE $\le \varepsilon^2$.*

Example 2.10 (Shifted Gaussians)**.** *Let* $\pi = \mathcal{N}(m_1, I_d)$, $g = \mathcal{N}(m_2, I_d)$. *Completing the square gives*

$$\chi^2(\pi \,\|\, g) = e^{\|m_1 - m_2\|^2} - 1, \qquad D_{\mathrm{KL}}(\pi \,\|\, g) = \tfrac{1}{2}\|m_1 - m_2\|^2.$$

For $m_1 - m_2 = (1, \ldots, 1)^\top$ *so* $\|m_1 - m_2\|^2 = d$*: AIS requires $N \sim e^d/\varepsilon^2$ samples, even though $D_{\mathrm{KL}}(\pi \,\|\, g) = d/2$ grows only linearly.*

The χ^2-divergence, not KL, is the correct measure of IS difficulty: a proposal "close" to π in KL may have exponentially large χ^2, rendering IS completely impractical.

The fundamental obstacle. Both RS and IS use a *static* proposal: independent draws from g are accepted or reweighted. There is no mechanism to move bad proposals toward π. This motivates *dynamical* approaches, where a distribution ρ_t evolves from an easy ρ_0 toward π via an ODE or SDE:

$$dX_t = b_t(X_t)\,dt \qquad \text{(deterministic flow)}, \tag{2.9}$$

$$dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t \quad \text{(stochastic flow)}. \tag{2.10}$$

The rest of the course constructs and analyzes such dynamics for high-dimensional sampling and generative modeling.

3 Probability Divergences

3.1 f-Divergences

Having seen that $\chi^2(\pi \,\|\, g)$ governs IS efficiency, we place it in the broader framework of f-divergences that can be used to measure dissimilarity between two measures.

Definition 3.1 (f-divergence)**.** *Let $f : (0, \infty) \to \mathbb{R}$ be convex with $f(1) = 0$. The* f-divergence *between $P \ll Q$ with densities p, q on $\mathbb{R}^d$ is*

$$D_f(P \,\|\, Q) = \int_{\mathbb{R}^d} f\!\left(\frac{p(x)}{q(x)}\right) q(x)\, dx = \mathbb{E}_Q\left[f\!\left(\frac{p(Y)}{q(Y)}\right)\right]. \tag{3.1}$$

If $P \not\ll Q$, set $D_f(P \,\|\, Q) = +\infty$.

Proposition 3.2 (Basic properties)**.** *(i)* Non-negativity*: $D_f(P \,\|\, Q) \geq 0$.*

(ii) Definiteness*: if f is strictly convex at 1, then $D_f(P \,\|\, Q) = 0 \iff P = Q$.*

Both follow from Jensen: $D_f(P \,\|\, Q) \geq f(\mathbb{E}_Q[p/q]) = f(1) = 0$.

3.2 Special Cases

$f(t)$	Name	Formula
$t \log t$	KL divergence $D_{\mathrm{KL}}(P \,\|\, Q)$	$\displaystyle\int p \log \frac{p}{q}\, dx$
$-\log t$	Reverse KL $D_{\mathrm{KL}}(Q \,\|\, P)$	$\displaystyle\int q \log \frac{q}{p}\, dx$
$(t-1)^2$	χ^2-divergence $\chi^2(P \,\|\, Q)$	$\displaystyle\int \frac{(p-q)^2}{q}\, dx = \int \frac{p^2}{q}\, dx - 1$
$\frac{1}{2}\lvert t-1\rvert$	Total variation $D_{\mathrm{TV}}(P, Q)$	$\displaystyle\frac{1}{2}\int \lvert p-q\rvert\, dx$

3.3 Comparison and High-Dimensional Behavior

Proposition 3.3 (Ordering)**.** *For any $P \ll Q$,*

$$4\, D_{\mathrm{TV}}(P, Q)^2 \leq D_{\mathrm{KL}}(P \,\|\, Q) \leq \log\!\big(1 + \chi^2(P \,\|\, Q)\big) \leq \chi^2(P \,\|\, Q)\,. \tag{3.2}$$

The first inequality is Pinsker's. The second follows from the log-sum inequality (Exercise 4(d)). The third uses $\log(1+t) \leq t$. In particular, χ^2 is strictly stronger than KL, which is stronger than TV.

For product measures $\pi = \pi_1^{\otimes d}$, $g = g_1^{\otimes d}$:

$D_{\mathrm{KL}}(\pi \,\|\, g) = d \cdot D_{\mathrm{KL}}(\pi_1 \,\|\, g_1)$ (linear), $\qquad \chi^2(\pi \,\|\, g) = (1+\chi^2(\pi_1 \,\|\, g_1))^d - 1$ (exponential).

For shifted Gaussians with $\|m_1 - m_2\|^2 = d$:

$$D_{\mathrm{TV}}(\pi, g) = O(1), \qquad D_{\mathrm{KL}}(\pi \,\|\, g) = \tfrac{d}{2}, \qquad \chi^2(\pi \,\|\, g) = e^d - 1.$$

TV and KL give a benign picture; χ^2 reveals the true exponential difficulty for IS.

3.4 Preview: Density Evolution under ODE/SDE Dynamics

Returning to the dynamical approach (2.9)–(2.10), a key question is: how does the density ρ_t of X_t evolve?

Theorem 3.4 (Continuity and Fokker–Planck equations)**.** *Let ρ_t denote the density of X_t.*

(i) ***ODE*** *$dX_t = b_t(X_t)\,dt$: ρ_t satisfies the* continuity equation

$$\partial_t \rho_t + \nabla \cdot (\rho_t\, b_t) = 0. \tag{3.3}$$

(ii) ***SDE*** *$dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t$: ρ_t satisfies the* Fokker–Planck equation

$$\partial_t \rho_t + \nabla \cdot (\rho_t\, b_t) = \tfrac{1}{2}\sigma_t^2\, \Delta \rho_t. \tag{3.4}$$

Proof sketch of (i). For $\phi \in C_c^\infty(\mathbb{R}^d)$,

$$\begin{aligned}\int \phi\, \partial_t \rho_t\, dx &= \tfrac{d}{dt}\mathbb{E}[\phi(X_t)] = \mathbb{E}[\nabla\phi(X_t)\cdot b_t(X_t)] \\ &= \int \nabla\phi \cdot b_t\, \rho_t\, dx = -\int \phi\, \nabla\cdot(\rho_t b_t)\, dx.\end{aligned}$$

Since ϕ is arbitrary, $\partial_t \rho_t = -\nabla\cdot(\rho_t b_t)$. □

Remark 3.5. Equation (3.3) expresses conservation of mass: ρ_t is transported by b_t with no creation or destruction. Equation (3.4) adds diffusion $\frac{\sigma_t^2}{2}\Delta\rho_t$ from the Brownian noise.

The rest of the course constructs b_t, σ_t so that $\rho_t \to \pi$ efficiently in high dimensions. We will discuss how these lead to dynamical algorithms and how these algorithms achieve accuracy in some f-divergence with certain complexity depending on dimensions.

Week 1 Exercises

1. (CLT for RS and AIS.) Prove Theorem 2.5 by the following steps.

(a) Write $I^{\mathrm{RS}}_{\pi,N}(h) = A_N/B_N$ where

$$A_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{U_i \le r_i\}\, h(Y_i), \qquad B_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{U_i \le r_i\}.$$

Verify the summands are i.i.d. under g. Compute $\mathbb{E}[A_N]$, $\mathbb{E}[B_N]$, and the 2×2 covariance matrix Σ of (A_N, B_N). Apply the multivariate CLT and the delta method with $\varphi(a,b) = a/b$ to derive G^2_{RS} in (2.6).

(b) Repeat for $I^{\mathrm{AIS}}_{\pi,N}(h)$: identify the i.i.d. summands, compute their joint covariance, and apply the delta method to derive G^2_{IS} in (2.7).

(c) Prove $G^2_{\mathrm{IS}} \le G^2_{\mathrm{RS}}$ using Cauchy–Schwarz. Characterize when equality holds.

2. (Non-asymptotic bound for AIS.) Prove Theorem 2.8: for $\|h\|_\infty \le 1$,

$$\mathbb{E}\Big[\big(I^{\mathrm{AIS}}_{\pi,N}(h) - \mathbb{E}_\pi[h]\big)^2\Big] \le \frac{4}{N}\big(\chi^2(\pi \,\|\, g) + 1\big).$$

Hint. Write $I^{\mathrm{AIS}}_{\pi,N}(h) = \tilde{A}_N/\tilde{B}_N$ and note $I^{\mathrm{AIS}}_{\pi,N}(h) - \mathbb{E}_\pi[h] = (\tilde{A}_N - \mathbb{E}_\pi[h]\cdot\tilde{B}_N)/\tilde{B}_N$. Argue $\tilde{B}_N > 0$ a.s., bound the numerator's second moment using $\|h\|_\infty \le 1$, and show $\mathbb{E}_g[\tilde{w}_i^2]/Z^2 = \chi^2(\pi \,\|\, g) + 1$. See [61].

3. (χ^2-divergence in high dimensions.)

(a) Let $\pi = \pi_1^{\otimes d}$ and $g = g_1^{\otimes d}$. Show $\chi^2(\pi \,\|\, g) = (1 + \chi^2(\pi_1 \,\|\, g_1))^d - 1$. *Hint.* Use $\int p_1^{\otimes d}(x)^2/g_1^{\otimes d}(x)\,dx = (\int p_1^2/g_1\,dx_1)^d$. Conclude χ^2 is exponential in d while $D_{\mathrm{KL}}(\pi \,\|\, g) = d \cdot D_{\mathrm{KL}}(\pi_1 \,\|\, g_1)$ is linear.

(b) Let $\pi = \mathcal{N}(m_1, I_d)$ and $g = \mathcal{N}(m_2, I_d)$. By completing the square, show $\chi^2(\pi \,\|\, g) = e^{\|m_1 - m_2\|^2} - 1$. For $m_1 - m_2 = (1, \dots, 1)^\top$, how many AIS samples are needed to achieve MSE $\le \varepsilon^2$?

(c) For the same Gaussian pair, compute $D_{\mathrm{KL}}(\pi \,\|\, g)$ and $D_{\mathrm{TV}}(\pi, g)$ as functions of $\delta = \|m_1 - m_2\|$. For $\delta^2 = d$, compare all three divergences as $d \to \infty$ and explain why $D_{\mathrm{KL}}(\pi \,\|\, g) = O(d)$ is misleadingly optimistic about IS difficulty.

4. (f-divergence properties.)

(a) Prove $D_f(P \,\|\, Q) \ge 0$ using Jensen's inequality. State precisely where convexity and $f(1) = 0$ are used.

(b) Prove that if f is strictly convex at 1, then $D_f(P \,\|\, Q) = 0 \implies P = Q$ as measures.

(c) Verify each row of the table in Section 3 by substituting the given f into Theorem 3.1.

(d) Prove $D_{\mathrm{KL}}(P \,\|\, Q) \le \log(1 + \chi^2(P \,\|\, Q))$. *Hint.* Write $p/q = 1 + (p-q)/q$ and use $\log t \le t - 1$. Deduce $D_{\mathrm{KL}}(P \,\|\, Q) \le \chi^2(P \,\|\, Q)$.

5. (Optimal IS proposal.) Fix target π and h with $\mathbb{E}_\pi[h] = \mu_h$.

(a) Show G^2_{IS} is minimized over densities g with $\mathrm{supp}(\pi) \subseteq \mathrm{supp}(g)$ by $g^*(x) \propto \pi(x)|h(x) - \mu_h|$. *Hint.* Apply Cauchy–Schwarz: $\int \frac{(\pi|h-\mu_h|)^2}{g}\,dx \cdot \int g\,dx \ge (\int \pi|h - \mu_h|\,dx)^2$.

(b) Compute $G^2_{\mathrm{IS}}(g^*)$ in closed form.

(c) Explain why g^* is not directly usable in practice.

(d) When $h \ge 0$, find a proposal $g^\dagger$ achieving $G^2_{\mathrm{IS}}(g^\dagger) = 0$. Why is $g^\dagger$ also impractical?

6. (Continuity and Fokker–Planck equations.)

(a) Give a careful proof of Theorem 3.4(i) using a test function $\phi \in C_c^\infty(\mathbb{R}^d)$ and integration by parts. State which property each step uses.

(b) Show that a stationary density ρ_∞ of (3.3) satisfies $\nabla \cdot (\rho_\infty b) = 0$. Can we find a nontrivial drift b such that $\pi \propto e^{-V}$ is stationary?

(c) Prove (3.4) using Itô's formula: $d\phi(X_t) = \nabla\phi \cdot b_t\,dt + \frac{\sigma_t^2}{2}\Delta\phi\,dt + \sigma_t \nabla\phi \cdot dW_t$.

(d) Verify that for the overdamped Langevin SDE $dX_t = -\nabla V(X_t)\,dt + \sqrt{2}\,dW_t$, the stationary distribution of (3.4) is $\pi \propto e^{-V}$.

4 Langevin Dynamics

4.1 The Overdamped Langevin SDE

Recall the fundamental question: given query access to ∇V, how do we construct a drift b_t and noise level σ_t in (2.10) so that $\rho_t \to \pi \propto e^{-V}$?

Intuition from optimization. If we simply want to find the mode $x^* = \arg\min V$, we run gradient descent $\dot{x}_t = -\nabla V(x_t)$, which drives $x_t \to x^*$ exponentially fast under strong convexity. But sampling from π requires exploring the entire distribution, not just its peak. The *overdamped Langevin SDE* adds Brownian noise to gradient descent:

$$dX_t = -\nabla V(X_t)\,dt + \sqrt{2}\,dW_t. \tag{4.1}$$

The drift $-\nabla V$ pushes X_t toward regions of high probability, while the noise $\sqrt{2}\,dW_t$ prevents collapse and drives exploration. The factor $\sqrt{2}$ is chosen so that $\pi \propto e^{-V}$ is exactly stationary, as we now verify.

4.2 Fokker–Planck Equation and Stationarity

Derivation of the Fokker–Planck equation. Let ρ_t denote the density of X_t under (4.1). For any test function $\phi \in C_c^\infty(\mathbb{R}^d)$, we compute $\frac{d}{dt}\mathbb{E}[\phi(X_t)]$ in two ways.

On one hand, differentiating under the integral:

$$\frac{d}{dt}\mathbb{E}[\phi(X_t)] = \int \phi(x)\,\partial_t \rho_t(x)\,dx.$$

On the other hand, Itô's formula applied to (4.1) gives

$$d\phi(X_t) = \underbrace{\nabla\phi(X_t)\cdot(-\nabla V(X_t))\,dt + \Delta\phi(X_t)\,dt}_{\text{bounded variation}} + \underbrace{\sqrt{2}\,\nabla\phi(X_t)\cdot dW_t}_{\text{martingale}}.$$

Taking expectations, the martingale term vanishes, so

$$\frac{d}{dt}\mathbb{E}[\phi(X_t)] = \int \big(-\nabla\phi\cdot\nabla V + \Delta\phi\big)\rho_t\,dx.$$

Integrating by parts (boundary terms vanish since $\phi \in C_c^\infty$):

$$\int (-\nabla\phi\cdot\nabla V)\,\rho_t\,dx = \int \phi\,\nabla\cdot(\rho_t\,\nabla V)\,dx, \qquad \int \Delta\phi\,\rho_t\,dx = \int \phi\,\Delta\rho_t\,dx.$$

Since ϕ is arbitrary, equating the two expressions gives the *Fokker–Planck equation* for Langevin dynamics:

$$\boxed{\partial_t \rho_t = \nabla\cdot(\rho_t\,\nabla V) + \Delta\rho_t.} \tag{4.2}$$

Compact form via the score. Using the *score trick* $\Delta\rho = \nabla\cdot(\nabla\rho) = \nabla\cdot(\rho\,\nabla\log\rho)$, equation (4.2) becomes

$$\partial_t\rho_t = \nabla\cdot\big(\rho_t\,(\nabla V + \nabla\log\rho_t)\big) = \nabla\cdot\Big(\rho_t\,\nabla\log\frac{\rho_t}{\pi}\Big)\,, \tag{4.3}$$

where the last step uses $\nabla V + \nabla\log\rho_t = -\nabla\log\pi + \nabla\log\rho_t = \nabla\log(\rho_t/\pi)$. This form immediately shows that $\rho_t = \pi$ is a stationary solution (the right-hand side vanishes), and that the dynamics transports mass along $-\nabla\log(\rho_t/\pi)$, pushing ρ_t toward π.

Proposition 4.1 (Stationarity of π)**.** *$\pi \propto e^{-V}$ is a stationary solution of (4.2). Equivalently, if $X_0 \sim \pi$ then $X_t \sim \pi$ for all $t \geq 0$.*

Remark 4.2. The compact form (4.3) reveals that Langevin dynamics is a *gradient flow* of $D_{\mathrm{KL}}(\rho_t\,\|\,\pi)$ in the space of probability measures equipped with the Wasserstein-2 metric (Jordan–Kinderlehrer–Otto, 1998) [41]. We will revisit this perspective when studying variational inference.

4.3 Mixing Time and the Wasserstein Metric

We now ask: starting from $\rho_0 \neq \pi$, how quickly does $\rho_t \to \pi$? To measure convergence we use the *Wasserstein-2 metric*, which is well-suited to the coupling arguments below and connects naturally to optimal transport.

Definition 4.3 (Wasserstein-2 metric)**.** *For probability measures μ, ν on $\mathbb{R}^d$,*

$$W_2(\mu,\nu) = \left(\inf_{\gamma\in\Gamma(\mu,\nu)}\int \|x-y\|_2^2\,d\gamma(x,y)\right)^{1/2} = \inf_{X\sim\mu,\,Y\sim\nu}\big(\mathbb{E}\|X-Y\|_2^2\big)^{1/2}, \tag{4.4}$$

where $\Gamma(\mu,\nu)$ denotes all couplings of μ and ν. The minimizing coupling is the optimal transport plan*.*

Example 4.4. *$W_2(\delta_x,\delta_y) = \|x-y\|_2$. For shifted Gaussians, $W_2(\mathcal{N}(m_1,I_d),\mathcal{N}(m_2,I_d)) = \|m_1-m_2\|_2$, growing like $\sqrt{d}$ for typical shifts, whereas $D_{\mathrm{KL}}(\mathcal{N}(m_1,I_d)\,\|\,\mathcal{N}(m_2,I_d)) = \frac{1}{2}\|m_1-m_2\|^2$ grows like d.*

The *mixing time* to tolerance ε in metric D is

$$t_{\mathrm{mix}}^{D}(\varepsilon;\rho_0) = \inf\big\{t\geq 0 : D(\rho_t,\pi)\leq\varepsilon\big\}. \tag{4.5}$$

4.4 Convergence under Strong Convexity

Definition 4.5 (Strong convexity)**.** *$V\in C^2(\mathbb{R}^d)$ is* α-strongly convex *if $\nabla^2 V(x)\succeq\alpha I_d$ for all $x\in\mathbb{R}^d$.*

Intuition from optimization. When V is α-strongly convex, the gradient flow ODE $\dot{x}_t = -\nabla V(x_t)$ satisfies

$$\frac{d}{dt}\|x_t - x^*\|_2^2 = 2(x_t - x^*) \cdot (-\nabla V(x_t)) \leq -2\alpha\|x_t - x^*\|_2^2,$$

using $(x-y)\cdot(\nabla V(x) - \nabla V(y)) \geq \alpha\|x-y\|_2^2$ and $\nabla V(x^*) = 0$, giving $\|x_t - x^*\|_2 \leq e^{-\alpha t}\|x_0 - x^*\|_2$. The Langevin SDE extends this contraction to the level of probability distributions via a *synchronous coupling*: run two copies of (4.1) with the same Brownian motion so that their noise cancels, reducing the problem to the deterministic ODE argument above.

Theorem 4.6 (W_2 contraction for Langevin)**.** *Assume V is α-strongly convex. Let ρ_t be the law of X_t solving (4.1) with $X_0 \sim \rho_0$. Then*

$$W_2(\rho_t, \pi) \leq e^{-\alpha t}\, W_2(\rho_0, \pi). \tag{4.6}$$

Proof. Run two copies of (4.1) with the same W_t: $X_0 \sim \rho_0$ and $Y_0 \sim \pi$, so $Y_t \sim \pi$ for all t by stationarity. Since both processes share the same noise, the difference $Z_t = X_t - Y_t$ satisfies the *deterministic* ODE

$$\dot{Z}_t = -\nabla V(X_t) + \nabla V(Y_t).$$

Then $\frac{d}{dt}\|Z_t\|_2^2 = 2(X_t - Y_t)\cdot(-\nabla V(X_t) + \nabla V(Y_t)) \leq -2\alpha\|Z_t\|_2^2$ by strong convexity, so Gronwall gives $\mathbb{E}\|X_t - Y_t\|_2^2 \leq e^{-2\alpha t}\mathbb{E}\|X_0 - Y_0\|_2^2$. Optimizing over the initial coupling (X_0, Y_0) with $X_0 \sim \rho_0$, $Y_0 \sim \pi$ yields (4.6). □

4.5 Initialization and Dimension Dependence

From (4.6), $W_2(\rho_t, \pi) \leq \varepsilon$ after $t \geq \frac{1}{\alpha}\log\frac{W_2(\rho_0,\pi)}{\varepsilon}$. It remains to bound $W_2(\rho_0, \pi)$ for a concrete initialization. For α-strongly convex V, a natural choice is $\rho_0 = \delta_{x^*}$.

Lemma 4.7 (Stein's identity)**.** *For $X \sim \pi \propto e^{-V}$ and smooth $f : \mathbb{R}^d \to \mathbb{R}^d$,*

$$\mathbb{E}_\pi[-\nabla V(X)\cdot f(X)] = -\mathbb{E}_\pi[\nabla \cdot f(X)]. \tag{4.7}$$

Proof. Integration by parts: $\int(-\nabla V)\cdot f\,\pi\,dx = \int \nabla\log\pi\cdot f\,\pi\,dx = \int \nabla\pi\cdot f\,dx = -\int \pi\,\nabla\cdot f\,dx$. □

Proposition 4.8 (Initial distance from the mode)**.** *If V is α-strongly convex, then*

$$W_2^2(\delta_{x^*}, \pi) \leq \mathbb{E}_\pi[\|X - x^*\|_2^2] \leq \frac{d}{\alpha}.$$

Proof. The first inequality uses (x^*, X) with $X \sim \pi$ as a coupling. For the second, apply (4.7) with $f(x) = x - x^*$ (so $\nabla\cdot f = d$): $\mathbb{E}_\pi[-\nabla V(X)\cdot(X - x^*)] = -d$. Since $\nabla V(x^*) = 0$, strong convexity gives $(\nabla V(X) - \nabla V(x^*))\cdot(X - x^*) \geq \alpha\|X - x^*\|_2^2$, so $\alpha\,\mathbb{E}_\pi\|X - x^*\|_2^2 \leq d$. □

Corollary 4.9 (Mixing time from mode initialization)**.** *Under α-strong convexity, starting from $\rho_0 = \delta_{x^*}$,*

$$t_{\mathrm{mix}}^{W_2}(\varepsilon; \delta_{x^*}) \le \frac{1}{\alpha} \log \frac{\sqrt{d/\alpha}}{\varepsilon}.$$

Remark 4.10 (Continuous time is not an algorithm)*.* The mixing time in Theorem 4.9 is for the *continuous-time* SDE (4.1), which requires simulating the stochastic process exactly and is not directly implementable. In practice one must discretize (4.1) in time with step size $h > 0$, yielding the *Unadjusted Langevin Algorithm* (ULA):

$$X_{k+1} = X_k - h\,\nabla V(X_k) + \sqrt{2h}\,\xi_k, \qquad \xi_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d).$$

The true measure of algorithmic complexity is therefore the number of gradient evaluations k needed to reach $W_2(\mathrm{law}(X_k), \pi) \le \varepsilon$. This discrete-time analysis, which gives a practically meaningful complexity bound, is the subject of the next lecture. But just from the structure of the bound, the complexity will not scale with dimension exponentially, when the target distribution is strongly log-concave, i.e., V is strongly convex.

5 Langevin Monte Carlo

5.1 From Continuous Time to Algorithms

The continuous-time Langevin SDE (4.1) establishes that $\rho_t \to \pi$ exponentially fast, but it is not directly implementable. The standard discretization with step size $h > 0$ gives the *Unadjusted Langevin Algorithm* (ULA):

$$X_{(k+1)h} = X_{kh} - h\,\nabla V(X_{kh}) + \sqrt{2h}\,\xi_k, \qquad \xi_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d), \tag{5.1}$$

where $\sqrt{2h}\,\xi_k = \sqrt{2}(W_{(k+1)h} - W_{kh})$ is the discretized Brownian increment. Denote $\rho_{kh} = \mathrm{law}(X_{kh})$.

Before analyzing ULA, we first revisit the simpler discrete-time setting of optimization, which provides the key structural insight.

5.2 Warm-up: Gradient Descent for Optimization

Under the assumption $\alpha I \preceq \nabla^2 V \preceq \beta I$, consider gradient descent (GD):

$$X_{(k+1)h} = X_{kh} - h\,\nabla V(X_{kh}). \tag{5.2}$$

To analyze convergence, write

$$\nabla V(X_{kh}) - \nabla V(x^*) = \underbrace{\left(\int_0^1 \nabla^2 V(sX_{kh} + (1-s)x^*)\,ds\right)}_{=:\,H_k}(X_{kh} - x^*),$$

where H_k satisfies $\alpha I \preceq H_k \preceq \beta I$. Since $\nabla V(x^*) = 0$, we have $\nabla V(X_{kh}) = H_k(X_{kh} - x^*)$, so

$$X_{(k+1)h} - x^* = (I - hH_k)(X_{kh} - x^*).$$

For $h \le 1/\beta$, $\|I - hH_k\|_2 \le 1 - h\alpha \le e^{-h\alpha}$, giving

$$\|X_{kh} - x^*\|_2 \le e^{-kh\alpha}\|X_0 - x^*\|_2. \tag{5.3}$$

To reach $\|X_{kh} - x^*\|_2 \le \varepsilon$, it suffices to take $k \ge \frac{1}{h\alpha}\log\frac{\|X_0 - x^*\|_2}{\varepsilon}$, which for $h = 1/\beta$ gives $O\Big(\frac{\beta}{\alpha}\log\frac{\|X_0 - x^*\|_2}{\varepsilon}\Big)$ iterations. Here $\kappa = \beta/\alpha$ is called the condition number.

5.3 ULA: Main Theorem

We now state the main convergence result for ULA. Throughout, we assume $\alpha I \preceq \nabla^2 V \preceq \beta I$ with $\alpha, \beta > 0$, so $\pi \propto e^{-V}$ is *strongly log-concave*.

Theorem 5.1 (ULA convergence)**.** *Under $\alpha I \preceq \nabla^2 V \preceq \beta I$, for step size $h \le 1/\beta$, the ULA iterates satisfy*

$$W_2(\rho_{kh}, \pi) \le e^{-kh\alpha}\, W_2(\rho_0, \pi) + O\left(\frac{\beta}{\alpha}\sqrt{hd}\right). \tag{5.4}$$

The bound (5.4) has two terms: an exponentially decaying *contraction* term (inherited from the continuous-time analysis), and a *discretization bias* term $O(\frac{\beta}{\alpha}\sqrt{hd})$ that is independent of k and persists even as $k \to \infty$. This confirms that ULA does not converge exactly to π, but to a neighborhood of size $O(\frac{\beta}{\alpha}\sqrt{hd})$ in W_2.

5.4 Proof of Theorem 5.1

Proof. The proof proceeds by introducing a "true" (continuous-time) reference process Y_t, then splitting the error at each step into a contraction term and a discretization error via the triangle inequality.

Reference process. For $t \in [kh, (k+1)h]$, define Y_t as the solution to the Langevin SDE (4.1) on this interval, sharing the same Brownian motion as ULA:

$$dY_t = -\nabla V(Y_t)\, dt + \sqrt{2}\, dW_t, \qquad kh \le t \le (k+1)h, \tag{5.5}$$

initialized so that (X_{kh}, Y_{kh}) is the *optimal coupling* of (ρ_{kh}, π):

$$W_2(\rho_{kh}, \pi) = \left(\mathbb{E}\|X_{kh} - Y_{kh}\|_2^2\right)^{1/2}.$$

Since π is stationary for (5.5), $Y_t \sim \pi$ for all t, and in particular $Y_{(k+1)h} \sim \pi$.

Triangle inequality decomposition. Writing $Y_{(k+1)h}$ explicitly:

$$Y_{(k+1)h} = Y_{kh} - \int_{kh}^{(k+1)h} \nabla V(Y_t)\, dt + \sqrt{2}(W_{(k+1)h} - W_{kh}).$$

Since $X_{(k+1)h}$ and $Y_{(k+1)h}$ share the same Brownian motion,

$$\begin{aligned} W_2(\rho_{(k+1)h}, \pi) &\le \left(\mathbb{E}\|X_{(k+1)h} - Y_{(k+1)h}\|_2^2\right)^{1/2} \\ &= \Big(\mathbb{E}\Big\|\underbrace{X_{kh} - h\nabla V(X_{kh}) - Y_{kh} + h\nabla V(Y_{kh})}_{①} - \underbrace{\int_{kh}^{(k+1)h} \big(\nabla V(Y_t) - \nabla V(Y_{kh})\big)\, dt}_{②}\Big\|_2^2\Big)^{1/2}. \end{aligned}$$

By the triangle inequality:

$$W_2(\rho_{(k+1)h}, \pi) \le \underbrace{\left(\mathbb{E}\|①\|_2^2\right)^{1/2}}_{\text{GD-like term}} + \underbrace{\left(\mathbb{E}\|②\|_2^2\right)^{1/2}}_{\text{discretization error}}. \tag{5.6}$$

Term ①: GD-like contraction. By the same linearization as in (5.3), $① = (I - hH_k)(X_{kh} - Y_{kh})$ where $H_k = \int_0^1 \nabla^2 V(sX_{kh} + (1-s)Y_{kh})\, ds$ satisfies $\alpha I \preceq H_k \preceq \beta I$. For $h \le 1/\beta$, $\|I - hH_k\|_2 \le e^{-h\alpha}$, so

$$\left(\mathbb{E}\|①\|_2^2\right)^{1/2} \le e^{-h\alpha}\, W_2(\rho_{kh}, \pi). \tag{5.7}$$

Term ②: discretization error. We first bound the expected squared gradient norm.

Lemma 5.2 (Expected squared gradient norm)**.** *For $Y \sim \pi \propto e^{-V}$ with $\alpha I \preceq \nabla^2 V \preceq \beta I$,*

$$\mathbb{E}_\pi \|\nabla V(Y)\|_2^2 \le \beta d.$$

Proof. Apply Stein's identity (4.7) with $f(y) = \nabla V(y)$ (so $\nabla \cdot f = \Delta V$):

$$-\mathbb{E}_\pi[\nabla V(Y) \cdot \nabla V(Y)] = -\mathbb{E}_\pi[\Delta V(Y)],$$

giving $\mathbb{E}_\pi\|\nabla V(Y)\|_2^2 = \mathbb{E}_\pi[\Delta V(Y)]$. Since $\nabla^2 V \preceq \beta I$, $\Delta V(y) = \operatorname{tr}(\nabla^2 V(y)) \le \beta d$, so $\mathbb{E}_\pi\|\nabla V(Y)\|_2^2 \le \beta d$. □

Lemma 5.3 (Discretization error)**.** *Under $\alpha I \preceq \nabla^2 V \preceq \beta I$ and $h \le 1/\beta$,*

$$\big(\mathbb{E}\|②\|_2^2\big)^{1/2} = O\Big(\beta\sqrt{h^3 d}\Big).$$

Proof. We have $② = \int_{kh}^{(k+1)h} \big(\nabla V(Y_t) - \nabla V(Y_{kh})\big) dt$. By Cauchy–Schwarz in time and $\|\nabla^2 V\|_2 \le \beta$:

$$\mathbb{E}\|②\|_2^2 \le h\beta^2 \int_{kh}^{(k+1)h} \mathbb{E}\|Y_t - Y_{kh}\|_2^2 \, dt.$$

Writing $Y_t - Y_{kh} = -\int_{kh}^t \nabla V(Y_s)\, ds + \sqrt{2}(W_t - W_{kh})$ and applying $(a+b)^2 \le 2a^2 + 2b^2$:

$$\mathbb{E}\|Y_t - Y_{kh}\|_2^2 \le 2\mathbb{E}\Big\|\int_{kh}^t \nabla V(Y_s)\, ds\Big\|_2^2 + 4(t-kh)d.$$

By Cauchy–Schwarz in time and Theorem 5.2: $\mathbb{E}\|\int_{kh}^t \nabla V(Y_s)\, ds\|_2^2 \le h^2 \beta d$. Therefore $\mathbb{E}\|Y_t - Y_{kh}\|_2^2 \le 6hd$ for $h \le 1/\beta$, giving $\mathbb{E}\|②\|_2^2 \le 6h^3\beta^2 d$. □

Combining the bounds. Substituting (5.7) and Lemma 5.3 into (5.6) and unrolling over k steps, using $1 - e^{-h\alpha} \ge h\alpha/2$ to sum the geometric series:

$$W_2(\rho_{kh}, \pi) \le e^{-kh\alpha}\, W_2(\rho_0, \pi) + O\left(\frac{\beta}{\alpha}\sqrt{hd}\right). \tag{5.8}$$

□

5.5 Algorithmic Complexity

To reach $W_2(\rho_{kh}, \pi) \le \varepsilon$, it suffices to require simultaneously that the contraction term satisfies $e^{-kh\alpha} W_2(\rho_0, \pi) \le \varepsilon/2$ and the bias term satisfies $O(\frac{\beta}{\alpha}\sqrt{hd}) \le \varepsilon/2$.

Corollary 5.4 (Iteration complexity of ULA)**.** *Under $\alpha I \preceq \nabla^2 V \preceq \beta I$, initialize at $\rho_0 = \delta_{x^*}$ so that $W_2(\rho_0, \pi) \le \sqrt{d/\alpha}$. Choose step size $h \asymp \alpha^2\varepsilon^2/(\beta^2 d)$. Then to achieve $W_2(\rho_{kh}, \pi) \le \varepsilon$, the required number of gradient evaluations is*

$$k = O\left(\frac{\kappa^2 d}{\varepsilon^2} \log \frac{\sqrt{d/\alpha}}{\varepsilon}\right), \tag{5.9}$$

where $\kappa = \beta/\alpha$ is the condition number. For fixed ε and α, the dimension dependence is $O(d \log d)$.

Proof. With $h \asymp \alpha^2\varepsilon^2/(\beta^2 d)$, the bias satisfies $\frac{\beta}{\alpha}\sqrt{hd} \asymp \varepsilon$. The contraction condition $e^{-kh\alpha} W_2(\rho_0, \pi) \le \varepsilon/2$ then requires $k \ge \frac{1}{h\alpha} \log \frac{W_2(\rho_0,\pi)}{\varepsilon} \asymp \frac{\kappa^2 d}{\varepsilon^2} \log \frac{\sqrt{d/\alpha}}{\varepsilon}$. □

Remark 5.5 (Dimensional scaling and references). The $O(d \log d)$ iteration complexity is a significant improvement over the exponential complexity of IS and RS. Other results measuring convergence in KL or χ^2 are given in Chewi's book [22]. The dimensional dependence can be improved to $\tilde{O}(\sqrt{d})$ by more refined analysis and assumptions; see Li–Zha–Tao (2022) [47]. We also note a dimension-free result if the quantity of interest only depends on low-dimensional coordinates [18].

Week 2 Exercises

1. **(Gaussian target: explicit computations.)** Throughout, let $\pi = \mathcal{N}(0, \Sigma)$ with $\Sigma \succ 0$, so $V(x) = \frac{1}{2} x^\top \Sigma^{-1} x$ and $\nabla V(x) = \Sigma^{-1} x$.

   (a) *(Continuous-time Langevin.)* Show that the Langevin SDE (4.1) becomes the *Ornstein–Uhlenbeck* (OU) process $dX_t = -\Sigma^{-1} X_t \, dt + \sqrt{2}\, dW_t$. If $X_0 \sim \mathcal{N}(m_0, S_0)$, show that $X_t \sim \mathcal{N}(m_t, S_t)$ for all $t \geq 0$, where

   $$m_t = e^{-\Sigma^{-1} t} m_0, \qquad S_t = e^{-\Sigma^{-1} t} S_0 \, e^{-\Sigma^{-1} t} + \Sigma\big(I - e^{-2\Sigma^{-1} t}\big).$$

   Verify that $m_t \to 0$ and $S_t \to \Sigma$ as $t \to \infty$.

   (b) *(Explicit W_2 convergence.)* For two Gaussians, $W_2^2\big(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)\big) = \|\mu_1 - \mu_2\|_2^2 + \|\Sigma_1^{1/2} - \Sigma_2^{1/2}\|_F^2$. Work in the eigenbasis of $\Sigma = \mathrm{diag}(\lambda_1, \dots, \lambda_d)$ with $0 < \lambda_1 \leq \cdots \leq \lambda_d$, and assume S_0 is diagonal with entries $s_1, \dots, s_d > 0$.

   (i) Show $\|m_t\|_2^2 = \sum_{i=1}^d e^{-2t/\lambda_i} m_{0,i}^2$, decaying at rate $2/\lambda_d = 2\lambda_{\min}(\Sigma^{-1})$.

   (ii) Show $S_{t,ii} = e^{-2t/\lambda_i} s_i + \lambda_i (1 - e^{-2t/\lambda_i})$, and for large t,

   $$\sqrt{S_{t,ii}} - \sqrt{\lambda_i} \approx \frac{s_i - \lambda_i}{2\sqrt{\lambda_i}} \, e^{-2t/\lambda_i},$$

   so $\|S_t^{1/2} - \Sigma^{1/2}\|_F^2$ decays at rate $4/\lambda_d$.

   (iii) Conclude $W_2(\rho_t, \pi) \leq C\, e^{-\alpha t}$ where $\alpha = 1/\lambda_d = \lambda_{\min}(\Sigma^{-1})$, confirming Theorem 4.6.

   Remark: in the non-diagonal case the dominant decay rate $e^{-\alpha t}$ persists, but the formula for $\|S_t^{1/2} - \Sigma^{1/2}\|_F^2$ is more involved.

   (c) *(ULA stationary distribution.)* Specialize to $\Sigma = I_d$ ($\alpha = \beta = 1$). For ULA with step size $h > 0$, $X_{(k+1)h} = (1-h) X_{kh} + \sqrt{2h}\, \xi_k$, show that $X_{kh} \sim \mathcal{N}(0, S_{kh})$ with $S_{kh} \to S_\infty = \frac{2}{2-h} I_d$ as $k \to \infty$. Hence ULA has a biased stationary distribution $\pi_h = \mathcal{N}\Big(0, \frac{2}{2-h} I_d\Big) \neq \pi$.

   (d) *(Bias and optimal step size.)* Show that

   $$W_2(\pi_h, \pi) = \sqrt{d} \left| \sqrt{\frac{2}{2-h}} - 1 \right| \asymp \frac{\sqrt{d}}{4} h \quad \text{as } h \to 0.$$

   Conclude that $h \asymp \varepsilon / \sqrt{d}$ suffices to achieve bias $\leq \varepsilon$. Compare to the general bound $h \asymp \varepsilon^2 / (\beta^2 d)$ from Theorem 5.4 (with $\beta = 1$) and explain the improvement.

2. **(Convergence under convexity without strong convexity.)** Assume $V \in C^2(\mathbb{R}^d)$ is convex ($\nabla^2 V \succeq 0$) with minimizer $x^* = \arg\min V$, but not α-strongly convex. Recall the gradient inequality

   $$V(y) \geq V(x) + \nabla V(x) \cdot (y - x) \quad \text{for all } x, y \in \mathbb{R}^d. \tag{5.10}$$

(a) *(Gradient flow: $O(1/T)$ rate.)* Consider $\dot{x}_t = -\nabla V(x_t)$.

(i) Show $\frac{d}{dt}V(x_t) = -\|\nabla V(x_t)\|_2^2 \le 0$.

(ii) Using (5.10) with $y = x^*$, show $V(x_t) - V(x^*) \le -\frac{1}{2}\frac{d}{dt}\|x_t - x^*\|_2^2$.

(iii) Integrate and use monotonicity of $V(x_t)$ to deduce

$$V(x_T) - V(x^*) \le \frac{\|x_0 - x^*\|_2^2}{2T}.$$

(b) *(Discrete GD: $O(1/K)$ rate.)* Consider GD (5.2) with $h \le 1/\beta$.

(i) Using the descent lemma, show $V(X_{(k+1)h}) - V(X_{kh}) \le -\frac{h}{2}\|\nabla V(X_{kh})\|_2^2$, so $V(X_{kh})$ is monotone decreasing.

(ii) Using (5.10), show $V(X_{kh}) - V(x^*) \le \frac{1}{2h}\big(\|X_{kh} - x^*\|_2^2 - \|X_{(k+1)h} - x^*\|_2^2\big)$. Sum over $k = 0, \dots, K-1$, use telescoping and monotone decrease of $V(X_{kh})$ to obtain

$$V(X_{Kh}) - V(x^*) \le \frac{\|X_0 - x^*\|_2^2}{2Kh}.$$

This $O(1/K)$ bound holds for the last iterate X_{Kh}.

(c) *(Extension to the Langevin setting.)* Optional. This relies on the interpretation that Langevin can be viewed as a gradient flow of KL under the Wasserstein geometry which we haven't discussed in the lecture.

(i) *(KL decay identity.)* Let $\mathcal{I}(\rho\|\pi) = \int \|\nabla \log(\rho/\pi)\|_2^2 \, \rho \, dx$ be the relative Fisher information. Using (4.3), prove

$$\frac{d}{dt} D_{\mathrm{KL}}(\rho_t \,\|\, \pi) = -\mathcal{I}(\rho_t\|\pi).$$

This is the exact analogue of $\frac{d}{dt}V(x_t) = -\|\nabla V(x_t)\|_2^2$ from (a)(i).

(ii) *($O(1/T)$ convergence via the HWI inequality.)* Under convexity of V, the *HWI inequality* (Otto–Villani) states $D_{\mathrm{KL}}(\rho \,\|\, \pi) \le W_2(\rho, \pi)\sqrt{\mathcal{I}(\rho\|\pi)}$. Use the following steps to deduce $D_{\mathrm{KL}}(\rho_T \,\|\, \pi) \le W_2(\rho_0, \pi)^2/T$:

- Use the synchronous coupling with $\alpha = 0$ to show $W_2(\rho_t, \pi) \le W_2(\rho_0, \pi) =: C$ for all $t \ge 0$.
- Let $f(t) = D_{\mathrm{KL}}(\rho_t \,\|\, \pi)$. Apply HWI to get $f(t) \le C\sqrt{\mathcal{I}(\rho_t\|\pi)}$, hence $\mathcal{I}(\rho_t\|\pi) \ge f(t)^2/C^2$ (squaring and rearranging). Combined with (i): $-f'(t) \ge f(t)^2/C^2$.
- Rewrite as $\frac{d}{dt}(1/f(t)) \ge 1/C^2$, integrate from 0 to T, and conclude $D_{\mathrm{KL}}(\rho_T \,\|\, \pi) \le W_2(\rho_0, \pi)^2/T$.

Compare with (a)(iii): the arguments are structurally identical, with $(V - V^*, \|\nabla V\|^2, \|x - x^*\|)$ replaced by $(D_{\mathrm{KL}}(\rho \,\|\, \pi), \mathcal{I}, W_2)$.

6 Score-Based Diffusion Models

6.1 Motivation and Setting

The setting of this section differs from the previous ones. We do *not* have access to a model V for the target π, but instead have data $X^{(1)}, \dots, X^{(N)} \sim \pi$ and wish to generate new samples from π. This is the *generative modeling* problem.

Popular approaches all try to find a transport from a simple reference $\gamma = \mathcal{N}(0, I_d)$ to the target π: variational autoencoders (VAE), generative adversarial networks (GAN), normalizing flows (NF), and diffusion models (DM). This section focuses on score based diffusion models, which approach this through a learned stochastic dynamics involving scores.

Recall from Section 4 that Langevin dynamics can be written as

$$dX_t = \nabla \log \pi(X_t)\, dt + \sqrt{2}\, dW_t,$$

so if we can learn *the score* $\nabla \log \pi$ from data, we can sample by running Langevin.

6.2 Score Matching

Warm-up: learning the score from data. Given a neural network $s(\cdot, \theta)$, we wish to find θ so that $s(x, \theta) \approx \nabla \log \pi(x)$. The natural loss is

$$L(\theta) = \int \|s(x,\theta) - \nabla \log \pi(x)\|_2^2 \, \pi(x)\, dx,$$

but this requires knowing $\nabla \log \pi$, which is unavailable. Integration by parts transforms this into a tractable form.

Proposition 6.1 (Score matching identity)**.** *Up to a constant C independent of θ,*

$$L(\theta) = \mathbb{E}_\pi\big[\|s(X,\theta)\|_2^2 + 2\,\nabla \cdot s(X,\theta)\big] + C. \tag{6.1}$$

Proof. Expand $\|s - \nabla \log \pi\|^2 = \|s\|^2 - 2s \cdot \nabla \log \pi + \|\nabla \log \pi\|^2$. The last term is constant in θ. For the cross term,

$$\begin{aligned} \int -2s(x,\theta) \cdot \nabla \log \pi(x)\, \pi(x)\, dx &= -2 \int s(x,\theta) \cdot \nabla \pi(x)\, dx \\ &= 2 \int (\nabla \cdot s(x,\theta))\, \pi(x)\, dx, \end{aligned}$$

where the last step is integration by parts (boundary terms vanish). Adding $\mathbb{E}_\pi[\|s\|^2]$ gives (6.1). □

The loss (6.1) depends only on expectations under π, so it can be empiricalized using data and minimized with stochastic optimization algorithms such as SGD/ADAM.

Issues. Plain score matching has two significant drawbacks:

① *The divergence term $\nabla \cdot s(X, \theta)$ is expensive to evaluate* for high-dimensional neural networks. (Variants such as sliced score matching reduce this cost.)

② *The loss only guarantees $L^2(\pi)$-accuracy*: since it averages against π, the score estimate may be inaccurate in low-probability regions where $\pi(x)$ is small. This is problematic.

Moreover, even with the exact score, running Langevin to sample from a multimodal π ($\pi \propto e^{-V}$ with non-convex V) can be very slow: the chain must traverse low-probability barriers between modes.

6.3 Multiscale Approach and the Noising Process

A simple trick: add noise. To address issue ②, replace π by the smoothed target $\pi * \mathcal{N}(0, \sigma^2 I)$. Adding noise to data yields samples from this smoothed distribution. One then:

- estimates the score of $\pi * \mathcal{N}(0, \sigma^2 I)$ first (more well-behaved globally, as the low-probability region is smoothed and becomes less low-probability);
- runs Langevin with the learned score to get approximate samples, then refines using the original score. One can imagine the convergence will be faster using this annealing strategy.

This multiscale idea is the intuitive foundation of score based diffusion models [66, 67].

The OU noising process (Variance Preserving design). Rather than a single noise level, we consider a continuous-time noising process. The standard choice is the *Ornstein–Uhlenbeck* (OU) process:

$$dX_t = -X_t\, dt + \sqrt{2}\, dW_t, \qquad X_0 \sim \pi, \tag{6.2}$$

which is Langevin dynamics (4.1) with $V(x) = \frac{1}{2}\|x\|^2$, so $\mathcal{N}(0, I_d)$ is stationary. As $t \to \infty$, the distribution of X_t converges to $\mathcal{N}(0, I_d)$ regardless of X_0.

Proposition 6.2 (Marginal of the OU process)**.** *Let $X_0 \sim \pi$ and $Z \sim \mathcal{N}(0, I_d)$ be independent. Then*

$$X_t \overset{d}{=} e^{-t} X_0 + \sqrt{1 - e^{-2t}}\, Z \sim \mathcal{N}\big(e^{-t} X_0,\, (1 - e^{-2t}) I_d\big) \quad \textit{given } X_0. \tag{6.3}$$

In particular, the conditional distribution is $p_t(\cdot \mid x_0) = \mathcal{N}(e^{-t} x_0,\, (1 - e^{-2t}) I_d)$.

Proof. Apply Itô's formula to $e^t X_t$:

$$d(e^t X_t) = e^t X_t\, dt + e^t\, dX_t = e^t X_t\, dt + e^t(-X_t\, dt + \sqrt{2}\, dW_t) = \sqrt{2}\, e^t\, dW_t.$$

Integrating from 0 to t: $e^t X_t - X_0 = \int_0^t \sqrt{2}\, e^s\, dW_s \sim \mathcal{N}\big(0,\, 2\int_0^t e^{2s}\, ds\big) = \mathcal{N}(0,\, e^{2t} - 1)$. Therefore $X_t \overset{d}{=} e^{-t} X_0 + \sqrt{1 - e^{-2t}}\, Z$. □

6.4 Time Reversal and the Reverse SDE

The key idea. If we can *reverse* the noising process (6.2), we can generate samples using it. The following theorem identifies the reverse dynamics.

Theorem 6.3 (Time reversal [5], Anderson 1982)**.** *Let X_t solve (6.2) for $t \in [0, T]$ with $X_0 \sim \pi$, and let p_t denote the density of X_t. Define the time-reversal $Y_t := X_{T-t}$ for $t \in [0, T]$. Then Y_t solves the* reverse SDE

$$dY_t = \big(Y_t + 2\nabla \log p_{T-t}(Y_t)\big)\, dt + \sqrt{2}\, dW_t, \tag{6.4}$$

where W_t is a Brownian motion.

Heuristic demonstration via Fokker–Planck. Let $q_t(x) = p_{T-t}(x)$ be the density of Y_t. The OU process (6.2) satisfies the Fokker–Planck equation

$$\partial_t p_t = \nabla \cdot (p_t\, x) + \Delta p_t.$$

Since $\partial_t q_t(x) = -\partial_t p_{T-t}(x)$, we have

$$\partial_t q_t = -\nabla \cdot (q_t\, x) - \Delta q_t.$$

Using the score trick $\Delta q = \nabla \cdot (q\, \nabla \log q)$:

$$\begin{aligned} \partial_t q_t &= -\nabla \cdot (q_t\, x) - \nabla \cdot (q_t\, \nabla \log q_t) \\ &= -\nabla \cdot \big(q_t(x + 2\nabla \log q_t)\big) + \Delta q_t. \end{aligned}$$

This leads to

$$\partial_t q_t + \nabla \cdot \big(q_t(x + 2\nabla \log q_t)\big) = \Delta q_t,$$

which is the Fokker–Planck equation for the SDE (6.4) (with $q_t = p_{T-t}$). □

Practice: the approximate reverse process. In practice the score $\nabla \log p_{T-t}$ is unknown and replaced by a learned approximation $\hat{S}_{T-t}(\cdot, \theta)$. The *approximate reverse process* is

$$d\hat{Y}_t = \big(\hat{Y}_t + 2\hat{S}_{T-t}(\hat{Y}_t, \theta)\big)\, dt + \sqrt{2}\, dW_t, \qquad \hat{Y}_0 \sim \mathcal{N}(0, I_d). \tag{6.5}$$

Let $\hat{q}_t$ denote the law of $\hat{Y}_t$. The goal is $\hat{q}_T \approx q_T = p_0 = \pi$, which holds (intuitively) when three conditions are met:

(1) T is large enough so that $p_T \approx \mathcal{N}(0, I_d)$;

(2) $\hat{S}_t \approx \nabla \log p_t$ accurately enough;

(3) the time discretization of (6.5) is fine enough.

6.5 Denoising Score Matching

It remains to learn $\nabla \log p_t$ tractably from data.

Marginalizing the score. Since $p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0$, differentiating gives

$$\nabla_x p_t(x) = \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0.$$

Dividing by $p_t(x)$ and using the Bayes formula for conditional distributions:

$$\nabla \log p_t(x) = \mathbb{E}\big[\nabla_{X_t} \log p_t(X_t \mid X_0) \mid X_t = x\big]. \tag{6.6}$$

Since $p_t(\cdot \mid x_0) = \mathcal{N}(e^{-t}x_0,\, (1 - e^{-2t})I_d)$, the conditional score is explicit:

$$\nabla_x \log p_t(x \mid x_0) = -\frac{x - e^{-t}x_0}{1 - e^{-2t}}. \tag{6.7}$$

In particular,

$$\nabla_x \log p_t(x) = \frac{e^{-t}\mathbb{E}[X_0 \mid X_t = x] - x}{1 - e^{-2t}}. \tag{6.8}$$

Here $\mathbb{E}[X_0 \mid X_t]$ is the optimal estimator (denoiser) of the clean data X_0 given the noisy observation X_t. Hence *learning the score is equivalent to learning the denoiser* $\mathbb{E}[X_0 \mid X_t]$. This above is also referred to as one form of Tweedie's formula.

Denoising score matching loss. By (6.6), the marginal score is the best $L^2(p_t)$-estimator given X_t. This motivates learning $S_t(x, \theta) \approx \nabla \log p_t(x)$ by minimizing the *denoising score matching* (DSM) loss:

$$L(\theta) = \int_0^T \mathbb{E}_{X_0 \sim p_0,\, X_t \mid X_0} \Big[\big\| S_t(X_t, \theta) - \nabla_{X_t} \log p_t(X_t \mid X_0) \big\|_2^2 \Big]\, dt. \tag{6.9}$$

By (6.6), $L(\theta)$ equals the original score matching objective integrated over time:

$$\int_0^T \mathbb{E}_{X_t \sim p_t} \Big[\big\| S_t(X_t, \theta) - \nabla_{X_t} \log p_t(X_t) \big\|_2^2 \Big]\, dt, \tag{6.10}$$

up to a constant that is independent of θ.

Substituting the explicit formula (6.7) into (6.9):

$$L(\theta) = \int_0^T \mathbb{E}\left[\left\| S_t(X_t, \theta) - \frac{e^{-t}X_0 - X_t}{1 - e^{-2t}} \right\|_2^2 \right] dt. \tag{6.11}$$

The loss (6.11) is fully tractable: $X_0 \sim p_0$ is sampled from data; $X_t \mid X_0 \sim \mathcal{N}(e^{-t}X_0, (1 - e^{-2t})I_d)$ is sampled via (6.3); and $\nabla_x \log p_t(X_t \mid X_0)$ is given by (6.7). There is no divergence term and no intractable expectation. In practice one samples a batch $X_0 \sim p_0$, a random time $t \in [0, T]$, and $X_t \mid X_0$, evaluates the empirical loss, and performs a step of SGD/ADAM. One often includes a time-dependent weight $w(t)$ to balance the loss at different noise levels.

6.6 Error Analysis

We now quantify how well $\hat{q}_T$ approximates $q_T = p_0 = \pi$ in terms of the score approximation error. Recall that $q_t = p_{T-t}$ denotes the law of the true time-reversal Y_t, and $\hat{q}_t$ denotes the law of the approximate process $\hat{Y}_t$ from (6.5). The drifts are

$$b_t(y) = y + 2\nabla \log p_{T-t}(y), \qquad \hat{b}_t(y) = y + 2S_{T-t}(y, \theta), \tag{6.12}$$

so $q_0 = p_T$, $\hat{q}_0 = \mathcal{N}(0, I_d)$, $q_T = p_0 = \pi$, and $\hat{q}_T \approx \pi$ is the goal.

Proposition 6.4 (W_2 error bound)**.** *Assume $\hat{b}_t$ is Lipschitz in x with constant $\hat{L}$, uniformly in t. Then*

$$W_2(q_T, \hat{q}_T)^2 \le e^{(2\hat{L}+1)T}\left(W_2(q_0, \hat{q}_0)^2 + \int_0^T \mathbb{E}_{q_t}\left[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2\right] dt\right). \tag{6.13}$$

The integral on the right is exactly the score matching loss.

Proof. Couple (6.4) and (6.5) with the same W_t. Then

$$\frac{d}{dt}\|Y_t - \hat{Y}_t\|_2^2 = 2(Y_t - \hat{Y}_t) \cdot \big(b_t(Y_t) - \hat{b}_t(\hat{Y}_t)\big).$$

Add and subtract $\hat{b}_t(Y_t)$:

$$= 2(Y_t - \hat{Y}_t) \cdot \big(b_t(Y_t) - \hat{b}_t(Y_t)\big) + 2(Y_t - \hat{Y}_t) \cdot \big(\hat{b}_t(Y_t) - \hat{b}_t(\hat{Y}_t)\big).$$

For the first term, Young's inequality $2ab \le a^2 + b^2$ gives

$$2(Y_t - \hat{Y}_t) \cdot \big(b_t(Y_t) - \hat{b}_t(Y_t)\big) \le \|Y_t - \hat{Y}_t\|_2^2 + \|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2.$$

For the second, the Lipschitz condition on $\hat{b}_t$ gives

$$2(Y_t - \hat{Y}_t) \cdot \big(\hat{b}_t(Y_t) - \hat{b}_t(\hat{Y}_t)\big) \le 2\hat{L}\|Y_t - \hat{Y}_t\|_2^2.$$

Combining:

$$\frac{d}{dt}\|Y_t - \hat{Y}_t\|_2^2 \le (2\hat{L} + 1)\|Y_t - \hat{Y}_t\|_2^2 + \|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2.$$

Taking expectations and applying Gronwall's inequality:

$$\mathbb{E}\|Y_T - \hat{Y}_T\|_2^2 \le e^{(2\hat{L}+1)T}\left(\mathbb{E}\|Y_0 - \hat{Y}_0\|_2^2 + \int_0^T \mathbb{E}\big[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2\big]\, dt\right).$$

Since $\mathbb{E}\|Y_T - \hat{Y}_T\|_2^2 \ge W_2(q_T, \hat{q}_T)^2$ and minimizing $\mathbb{E}\|Y_0 - \hat{Y}_0\|_2^2$ over all couplings of $(q_0, \hat{q}_0)$ gives $W_2(q_0, \hat{q}_0)^2$, we obtain (6.13). □

The above W_2 upper bound using coupling arguments leads to exponential dependence on T.

Proposition 6.5 (KL error bound)**.** *With the same setup,*

$$D_{\mathrm{KL}}(q_T \,\|\, \hat{q}_T) \le D_{\mathrm{KL}}(q_0 \,\|\, \hat{q}_0) + \int_0^T \frac{1}{4}\mathbb{E}_{q_t}\big[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2\big]\, dt. \tag{6.14}$$

Proof. This can be easily derived using path-KL and Girsanov formula:

$$D_{\mathrm{KL}}(q_T \,\|\, \hat{q}_T) \le D_{\mathrm{KL}}\big(q_{[0,T]} \,\|\, \hat{q}_{[0,T]}\big) = D_{\mathrm{KL}}(q_0 \,\|\, \hat{q}_0) + \int_0^T \frac{1}{4}\mathbb{E}_{q_t}\big[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2\big]\, dt.$$

We also provide an explicit proof based on PDE calculations. Both q_t and $\hat{q}_t$ satisfy Fokker–Planck equations

$$\partial_t q_t + \nabla\cdot(q_t b_t) = \Delta q_t, \qquad \partial_t \hat{q}_t + \nabla\cdot(\hat{q}_t \hat{b}_t) = \Delta \hat{q}_t.$$

We differentiate $D_{\mathrm{KL}}(q_t \,\|\, \hat{q}_t) = \int q_t \log(q_t/\hat{q}_t)\, dx$:

$$\frac{d}{dt} D_{\mathrm{KL}}(q_t \,\|\, \hat{q}_t) = \underbrace{\int \partial_t q_t \, \log \frac{q_t}{\hat{q}_t}\, dx}_{(A)} - \underbrace{\int \frac{q_t}{\hat{q}_t}\, \partial_t \hat{q}_t \, dx}_{(B)}, \tag{6.15}$$

where the middle term $\int \partial_t q_t \, dx = 0$ was dropped.

Term (A). Substituting the FP equation for q_t and integrating by parts:

$$\begin{aligned}(A) &= \int \Big(-\nabla\cdot(q_t b_t) + \Delta q_t\Big) \log \frac{q_t}{\hat{q}_t}\, dx \\ &= \int q_t\, b_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}\, dx - \int \nabla q_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}\, dx.\end{aligned}$$

Using $\nabla q_t = q_t \nabla \log q_t$ and splitting $\nabla \log q_t = \nabla \log(q_t/\hat{q}_t) + \nabla \log \hat{q}_t$:

$$-\int \nabla q_t \cdot \nabla \log \frac{q_t}{\hat{q}_t} = -\int q_t \left\| \nabla \log \frac{q_t}{\hat{q}_t} \right\|_2^2 dx - \int q_t\, \nabla \log \hat{q}_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}\, dx.$$

Term (B). Substituting the FP equation for $\hat{q}_t$ and integrating by parts:

$$\begin{aligned}(B) &= \int \frac{q_t}{\hat{q}_t}\Big(-\nabla\cdot(\hat{q}_t \hat{b}_t) + \Delta \hat{q}_t\Big)\, dx \\ &= \int q_t\, \hat{b}_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}\, dx - \int q_t\, \nabla \log \hat{q}_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}\, dx,\end{aligned}$$

using $-\int \nabla\cdot(\hat{q}_t \hat{b}_t)\frac{q_t}{\hat{q}_t} = \int q_t \hat{b}_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}$ and $\int \Delta \hat{q}_t \frac{q_t}{\hat{q}_t} = \int q_t \nabla \log \hat{q}_t \cdot \nabla \log \frac{q_t}{\hat{q}_t}$.

Combining. Substituting into (6.15), the $\nabla \log \hat{q}_t$ terms cancel, leaving

$$\frac{d}{dt} D_{\mathrm{KL}}(q_t \,\|\, \hat{q}_t) = \int q_t\, \nabla \log \frac{q_t}{\hat{q}_t} \cdot (b_t - \hat{b}_t)\, dx - \int q_t \left\| \nabla \log \frac{q_t}{\hat{q}_t} \right\|_2^2 dx. \tag{6.16}$$

Applying Young's inequality $ab \le \frac{\sigma^2}{2}a^2 + \frac{1}{2\sigma^2}b^2$ with $\sigma^2 = 2$ to the first integral:

$$\int q_t \, \nabla \log \frac{q_t}{\hat{q}_t} \cdot (b_t - \hat{b}_t) \le \int q_t \left\| \nabla \log \frac{q_t}{\hat{q}_t} \right\|_2^2 dx + \frac{1}{4} \int q_t \|b_t - \hat{b}_t\|_2^2 \, dx.$$

The Fisher information terms cancel, giving $\frac{d}{dt} D_{\mathrm{KL}}(q_t \,\|\, \hat{q}_t) \le \frac{1}{4} \mathbb{E}_{q_t} \|b_t - \hat{b}_t\|_2^2$. Integrating from 0 to T gives (6.14). □

Remark 6.6. Substituting the drifts (6.12), $b_t(y) - \hat{b}_t(y) = 2(\nabla \log p_{T-t}(y) - S_{T-t}(y, \theta))$, so the integrands in (6.13) and (6.14) are proportional to $\|\nabla \log p_{T-t}(y) - S_{T-t}(y, \theta)\|_2^2$, which corresponds exactly to the loss (6.10). Thus *minimizing the training loss directly controls the quality of the generated samples* in both W_2 and KL. For time discretizations, see for example [16, 69].

Week 3 Exercises

1. **(Probability flow ODE.)** The reverse SDE (6.4) has noise, which is useful for sampling but makes likelihood computation difficult. There is a deterministic counterpart, the *probability flow ODE*,

$$\frac{dY_t}{dt} = Y_t + \nabla \log p_{T-t}(Y_t), \qquad Y_0 \sim p_T. \tag{6.17}$$

It is a purely deterministic flow. The key claim is that it shares the same marginal densities as the reverse SDE.

   (a) *(Same marginals as the reverse SDE.)* Let q_t^{ODE} denote the law of Y_t solving (6.17). Show that $q_t^{\mathrm{ODE}} = p_{T-t}$ for all t by verifying that p_{T-t} satisfies the continuity equation for (6.17):

$$\partial_t p_{T-t} + \nabla \cdot \big(p_{T-t}(y + \nabla \log p_{T-t})\big) = 0.$$

   Hint. Differentiate p_{T-t} in t using the Fokker–Planck equation for the forward OU process (6.2), and use the score trick $\Delta p = \nabla \cdot (p \nabla \log p)$.

   (b) *(Comparison with SDE.)* The reverse SDE (6.4) has drift $y + 2\nabla \log p_{T-t}(y)$, while the probability flow ODE (6.17) has drift $y + \nabla \log p_{T-t}(y)$. Both yield $q_T = p_0 = \pi$ when initialized at $q_0 = p_T$.

      (i) Explain why the ODE formulation allows exact likelihood computation via the instantaneous change-of-variables formula $\frac{d}{dt} \log q_t^{\mathrm{ODE}}(Y_t) = -\nabla \cdot (Y_t + \nabla \log p_{T-t}(Y_t))$, while the SDE formulation does not.

      (ii) Explain why the ODE formulation may allow faster generation.

2. **(Bounding the initialization error.)** The error bounds (6.13) and (6.14) each contain an initialization term measuring how well $\hat{q}_0 = \mathcal{N}(0, I_d)$ approximates $q_0 = p_T$.

   (a) *(W_2 initialization bound.)* The OU forward process (6.2) is Langevin dynamics (4.1) with $V(x) = \frac{1}{2}\|x\|^2$, so $\alpha = 1$ and stationary distribution $\mathcal{N}(0, I_d)$. Apply Theorem 4.6 directly to obtain

$$W_2(p_T, \mathcal{N}(0, I_d)) \le e^{-T}\, W_2(\pi, \mathcal{N}(0, I_d)).$$

   (b) *(KL initialization bound.)* For the KL bound (6.14), we need $D_{\mathrm{KL}}(q_0 \,\|\, \hat{q}_0) = D_{\mathrm{KL}}(p_T \,\|\, \mathcal{N}(0, I_d))$. Since the OU process (6.2) is Langevin with stationary distribution $\gamma = \mathcal{N}(0, I_d)$, the KL decay identity from Week 2 Exercise 2(c)(i) gives

$$\frac{d}{dt} D_{\mathrm{KL}}(p_t \,\|\, \gamma) = -\mathcal{I}(p_t \| \gamma), \qquad \mathcal{I}(p_t \| \gamma) = \int \|\nabla \log(p_t/\gamma)\|_2^2 \, p_t \, dx.$$

(i) For $\gamma = \mathcal{N}(0, I_d)$, show that $\nabla \log \gamma(x) = -x$, so

$$\mathcal{I}(p_t \| \gamma) = \mathbb{E}_{p_t} \| \nabla \log p_t(X) + X \|_2^2.$$

(ii) We can show (accept without proof) that

$$D_{\mathrm{KL}}(p_t \,\|\, \gamma) \le \tfrac{1}{2}\, \mathcal{I}(p_t \| \gamma).$$

This is the *log-Sobolev inequality* for $\mathcal{N}(0, I_d)$. Combine with the decay identity to show $\frac{d}{dt} D_{\mathrm{KL}}(p_t \,\|\, \gamma) \le -2 D_{\mathrm{KL}}(p_t \,\|\, \gamma)$, and conclude by Gronwall that

$$D_{\mathrm{KL}}(p_T \,\|\, \mathcal{N}(0, I_d)) \le e^{-2T}\, D_{\mathrm{KL}}(\pi \,\|\, \mathcal{N}(0, I_d))\,.$$

(c) *(Complete W_2 error bound.)* Combine part (a) with (6.13). Let $\varepsilon_{\mathrm{score}}^2 = \int_0^T \mathbb{E}_{q_t}[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2]\, dt$. Show that

$$W_2(\pi, \hat{q}_T) \le e^{(\hat{L} + \frac{1}{2})T} \big(e^{-T}\, W_2(\pi, \mathcal{N}(0, I_d)) + \varepsilon_{\mathrm{score}} \big).$$

What happens to this upper bound as $T \to \infty$?

(d) *(Complete KL error bound.)* Similarly combine part (b) with (6.14) to obtain

$$D_{\mathrm{KL}}(\pi \,\|\, \hat{q}_T) \le e^{-2T}\, D_{\mathrm{KL}}(\pi \,\|\, \mathcal{N}(0, I_d)) + \int_0^T \frac{1}{4} \mathbb{E}_{q_t}[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2]\, dt.$$

Note that unlike the W_2 bound, this KL bound has *no exponentially growing prefactor* in T.

3. (Gaussian mixture: explicit computations.) Let $d = 1$ and $\pi = \frac{1}{2}\mathcal{N}(\mu, 1) + \frac{1}{2}\mathcal{N}(-\mu, 1)$ for $\mu > 0$.

(a) *(Marginals of the OU process.)* Show that

$$p_t = \tfrac{1}{2}\mathcal{N}(e^{-t}\mu,\, 1) + \tfrac{1}{2}\mathcal{N}(-e^{-t}\mu,\, 1).$$

(b) *(Explicit score.)* Write $w_t(x) = \phi(x;\, e^{-t}\mu, 1) \,/\, [\phi(x;\, e^{-t}\mu, 1) + \phi(x; -e^{-t}\mu, 1)]$ for the mixture weight, where $\phi(x;\, m, \sigma^2)$ is the $\mathcal{N}(m, \sigma^2)$ density. Show that

$$\nabla \log p_t(x) = -\big(x - e^{-t}\mu\, (2 w_t(x) - 1) \big),$$

and verify that as $t \to \infty$, $w_t(x) \to \frac{1}{2}$ and $\nabla \log p_t(x) \to -x$ (score of $\mathcal{N}(0, 1)$).

7 Flow Matching, Rectified Flow, Stochastic Interpolants

7.1 Motivation: Beyond Gaussian Noise Source and Infinite Time

The score-based diffusion model of Section 6 has two structural limitations:

① *Finite-time truncation.* The forward OU process (6.2) reaches $\mathcal{N}(0, I_d)$ only as $T \to \infty$; in practice one truncates to $[0, T]$, introducing initialization errors.

② *Restricted transport.* The noising process always moves data toward Gaussian noise. One can therefore only learn to transport between a data distribution and $\mathcal{N}(0, I_d)$, not between two arbitrary distributions π_0 and π_1.

In many applications, one wants a direct transport between two non-Gaussian distributions — for instance, π_0 a low-resolution image distribution and π_1 a high-resolution image distribution. Flow matching [49], as well as concurrent work, stochastic interpolants [2, 1] and rectified flows [52], addresses both issues by replacing the noising process with a general interpolation over a finite time interval.

7.2 Setup and Interpolation Process

Let $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ be samples from two target distributions (which may be accessed only through data), and let $Z \sim \mathcal{N}(0, I_d)$ be independent auxiliary noise. Define an *interpolation process*

$$I_t = I_t(X_0, X_1, Z), \qquad t \in [0, 1], \tag{7.1}$$

satisfying the boundary conditions $I_0 = X_0$ and $I_1 = X_1$. Let μ_t denote the law of I_t (with density also written μ_t by abuse of notation), so $\mu_0 = \pi_0$ and $\mu_1 = \pi_1$. The goal is to construct a vector field b_t such that the ODE $dY_t = b_t(Y_t)\, dt$ transports π_0 to π_1 by matching the marginals of the interpolation.

Example 7.1 (Linear interpolant)**.** *The simplest choice is the linear interpolant*

$$I_t = \alpha_t X_0 + \beta_t X_1, \qquad \dot{I}_t = \dot{\alpha}_t X_0 + \dot{\beta}_t X_1, \tag{7.2}$$

where $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = 0$, so that $I_0 = X_0$ and $I_1 = X_1$. The canonical choice $\alpha_t = 1 - t$, $\beta_t = t$ gives $I_t = (1-t)X_0 + tX_1$ with $\dot{I}_t = X_1 - X_0$.

Remark 7.2 (Connection to diffusion models). Consider an infinite time interval and $X_1 = Z \sim \mathcal{N}(0, I_d)$ and the interpolant coefficients $\alpha_t = e^{-t}$, $\beta_t = \sqrt{1 - e^{-2t}}$. By Theorem 6.2, the OU marginal satisfies $p_t(\cdot \mid x_0) = \mathcal{N}(e^{-t}x_0, (1 - e^{-2t})I_d)$, so

$$I_t \overset{d}{=} e^{-t}X_0 + \sqrt{1 - e^{-2t}}\, Z \;\sim\; p_t,$$

where $p_t = \mathrm{Law}(X_t)$ is the OU forward process marginal. Thus $\mu_t = p_t$: the interpolant marginals coincide with the diffusion model forward marginals. See exercise at the end of this section.

Remark 7.3. In the above two examples, $I_t = I_t(X_0, X_1)$ without dependence on the latent random variable Z. This means that conditional on X_0, X_1, I_t is a point mass. One can introduce additional stochasticity so that $I_t|X_0, X_1$ has some variability and there is a conditional density $p_t(\cdot|X_0, X_1)$. Using the latent Z can achieve this. In the original paper on flow matching, the conditional distribution is directly modeled, while in stochastic interpolants, the interpolant process is the primary modeling object.

7.3 The Matching Theorem

Theorem 7.4. *Let I_t be an interpolation process (7.1) with marginal law μ_t, and define the* conditional velocity field

$$b_t(x) = \mathbb{E}\big[\dot{I}_t \mid I_t = x\big], \qquad \dot{I}_t := \tfrac{d}{dt} I_t. \tag{7.3}$$

Let Y_t solve the ODE

$$dY_t = b_t(Y_t)\, dt, \qquad Y_0 \sim \pi_0. \tag{7.4}$$

Then $\mathrm{Law}(Y_t) = \mu_t$ *for all $t \in [0,1]$. In particular, $Y_1 \sim \pi_1$.*

Proof. Assume μ_t has a smooth density (also denoted μ_t); otherwise one uses a weak formulation. For any test function $\phi \in C_c^\infty(\mathbb{R}^d)$, differentiate $\int \mu_t\, \phi\, dx = \mathbb{E}[\phi(I_t)]$:

$$\int \partial_t \mu_t\, \phi\, dx = \frac{d}{dt}\mathbb{E}[\phi(I_t)] = \mathbb{E}[\nabla\phi(I_t) \cdot \dot{I}_t].$$

By the tower property of conditional expectation,

$$\begin{aligned}\mathbb{E}[\nabla\phi(I_t) \cdot \dot{I}_t] &= \mathbb{E}\big[\nabla\phi(I_t) \cdot \mathbb{E}[\dot{I}_t \mid I_t]\big] = \int \nabla\phi(x) \cdot b_t(x)\, \mu_t(x)\, dx \\ &= -\int \phi(x)\, \nabla \cdot (\mu_t\, b_t)(x)\, dx,\end{aligned}$$

where the last equality is integration by parts. Since ϕ is arbitrary:

$$\partial_t \mu_t + \nabla \cdot (\mu_t\, b_t) = 0. \tag{7.5}$$

This is the continuity equation (3.3) for the ODE (7.4). Since $\mathrm{Law}(Y_0) = \pi_0 = \mu_0$ and both $\mathrm{Law}(Y_t)$ and μ_t satisfy (7.5) with the same initial condition, they coincide for all t. □

7.4 Algorithm: Training by Matching the Velocity

Theorem 7.4 reduces transporting π_0 to π_1 to learning $b_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x]$. Since the conditional expectation is the L^2-projection, b_t is the minimizer of the *flow matching loss*:

$$L(\theta) = \int_0^1 \mathbb{E}\Big[\big\|\hat{b}_t(I_t, \theta) - \dot{I}_t\big\|_2^2\Big]\, dt. \tag{7.6}$$

The loss (7.6) is fully tractable: at each step, sample $X_0 \sim \pi_0$, $X_1 \sim \pi_1$, and $Z \sim \mathcal{N}(0, I_d)$ from data; evaluate I_t and $\dot{I}_t$ in closed form from (7.1); and optimize with SGD/ADAM. Once trained, generate new samples from π_1 by integrating the ODE (7.4) from $Y_0 \sim \pi_0$.

Error bound. Applying the same coupling argument as Theorem 6.4, if $\hat{b}_t$ is $\hat{L}$-Lipschitz in x uniformly in t, then

$$W_2^2(\hat{\mu}_1, \mu_1) \le e^{(2\hat{L}+1)} \int_0^1 \mathbb{E}\big[\|\hat{b}_t(I_t, \theta) - b_t(I_t)\|_2^2\big]\, dt, \tag{7.7}$$

where $\hat{\mu}_1$ is the law of $\hat{Y}_1$ from the approximate ODE $d\hat{Y}_t = \hat{b}_t(\hat{Y}_t, \theta)\, dt$. Note the absence of the initialization error term $W_2(q_0, \hat{q}_0)^2$: with exact boundary conditions $I_0 = Z \sim \mathcal{N}(0, I_d) = \hat{Y}_0$, the initialization error is zero.

Remark 7.5 (Summary of advantages over diffusion models). For the same interpolant choice as in Theorem 7.2, flow matching and diffusion models share the same forward marginals $\mu_t = p_t$. Flow matching's advantages are:

- *Finite horizon:* the time interval is $[0, 1]$, with exact boundary conditions and no truncation error.
- *General transport:* π_0 and π_1 can be any two distributions, not just data and Gaussian noise.
- *Simpler framework:* the loss (7.6) regresses onto $\dot{I}_t$, which is known in closed form. In comparison, denoising score matching (6.9) regresses onto the conditional score $\nabla_{X_t} \log p_t(X_t \mid X_0)$ whose form requires some calculations to obtain.

7.5 Connection to Score Based Diffusion Models

In this subsection we specialize to the interpolant $I_t = \alpha_t Z + \beta_t X_1$ which goes from Z at time $t = 0$ to X_1 at $t = 1$, and derive explicit formulas connecting b_t, the score $\nabla \log \mu_t$, and the diffusion model.

Noise predictor and denoiser. Taking conditional expectations in $I_t = \alpha_t Z + \beta_t X_1$ gives:

$$x = \alpha_t \underbrace{\mathbb{E}[Z \mid I_t = x]}_{\text{noise predictor}} + \beta_t \underbrace{\mathbb{E}[X_1 \mid I_t = x]}_{\text{denoiser}}, \qquad b_t(x) = \dot{\alpha}_t\, \mathbb{E}[Z \mid I_t = x] + \dot{\beta}_t\, \mathbb{E}[X_1 \mid I_t = x]. \tag{7.8}$$

Thus b_t can equivalently be expressed through the noise predictor $\mathbb{E}[Z \mid I_t = x]$ or the denoiser $\mathbb{E}[X_1 \mid I_t = x]$, analogously to (6.8) in the diffusion model.

Score of the interpolant marginals. The following lemma is the key tool.

Lemma 7.6 (Score via conditional expectation)**.** *Let X, Y be jointly distributed random variables with p the density of X. Then*

$$\nabla \log p(x) = \mathbb{E}\big[\nabla_X \log p(X \mid Y) \mid X = x\big]. \tag{7.9}$$

Proof. Differentiating $p(x) = \int p(x \mid y)\, q(y)\, dy$: $\nabla_x p(x) = \int \nabla_x p(x \mid y)\, q(y)\, dy$. Dividing by $p(x)$ and recognizing the posterior:

$$\nabla \log p(x) = \int \nabla_x \log p(x \mid y)\, \frac{p(x \mid y) q(y)}{p(x)}\, dy = \mathbb{E}[\nabla_X \log p(X \mid Y) \mid X = x].$$

□

Proposition 7.7 (Score and noise predictor)**.** *For the interpolant $I_t = \alpha_t Z + \beta_t X_1$,*

$$\nabla \log \mu_t(x) = -\frac{1}{\alpha_t}\, \mathbb{E}[Z \mid I_t = x]. \tag{7.10}$$

Proof. Apply Theorem 7.6 with $X = I_t$, $Y = X_1$. Given $X_1 = x_1$, $I_t \mid X_1 = x_1 \sim \mathcal{N}(\beta_t x_1,\, \alpha_t^2 I_d)$, so

$$\nabla_x \log p_{t|1}(x \mid x_1) = -\frac{x - \beta_t x_1}{\alpha_t^2}.$$

By Theorem 7.6:

$$\nabla \log \mu_t(x) = \mathbb{E}\left[-\frac{I_t - \beta_t X_1}{\alpha_t^2} \,\middle|\, I_t = x \right] = -\frac{1}{\alpha_t^2}\, \mathbb{E}[I_t - \beta_t X_1 \mid I_t = x] = -\frac{1}{\alpha_t^2}\, \mathbb{E}[\alpha_t Z \mid I_t = x].$$

□

Solving (7.8) and (7.10) for $\mathbb{E}[Z \mid I_t = x]$ and substituting into the expression for b_t:

$$b_t(x) = \alpha_t^2 \left(\frac{\dot\beta_t}{\beta_t} - \frac{\dot\alpha_t}{\alpha_t} \right) \nabla \log \mu_t(x) + \frac{\dot\beta_t}{\beta_t}\, x. \tag{7.11}$$

This expresses the flow matching velocity as an affine function of the score $\nabla \log \mu_t$, mirroring (4.3) in the Langevin setting.

A family of SDEs with the same marginals. By the score identity $\nabla \cdot (\mu_t \nabla \log \mu_t) = \Delta \mu_t$, adding any diffusion $g_t \geq 0$ to the drift of (7.4) preserves the marginals.

Proposition 7.8. *For any $g_t \geq 0$, the SDE*

$$dY_t = \big(b_t(Y_t) + g_t \nabla \log \mu_t(Y_t) \big)\, dt + \sqrt{2 g_t}\, dW_t, \qquad Y_0 \sim \mathcal{N}(0, I_d), \tag{7.12}$$

satisfies $\mathrm{Law}(Y_t) = \mu_t$ *for all* $t \in [0, 1]$.

Proof. The Fokker–Planck equation for (7.12) is

$$\partial_t \mu_t + \nabla \cdot \big(\mu_t (b_t + g_t \nabla \log \mu_t) \big) = g_t \Delta \mu_t.$$

Since $\nabla \cdot (\mu_t \nabla \log \mu_t) = \Delta \mu_t$, the right-hand side equals $g_t \nabla \cdot (\mu_t \nabla \log \mu_t)$, which cancels the added term, reducing to the continuity equation (7.5) that μ_t satisfies. □

The ODE (7.4) corresponds to $g_t = 0$. Equation (7.11) shows that (7.12) is always an SDE whose drift is an affine function of the score $\nabla \log \mu_t$.

Remark 7.9 (Stochastic interpolants: score formula). For the stochastic interpolant $I_t = \alpha_t X_0 + \beta_t X_1 + \sigma_t Z$ with $\sigma_0 = \sigma_1 = 0$, Theorem 7.6 applied with $Y = (X_0, X_1)$ gives $\nabla \log \mu_t(x) = -\mathbb{E}[Z/\sigma_t \mid I_t = x]$, and $b_t(x) = \mathbb{E}[\dot\alpha_t X_0 + \dot\beta_t X_1 + \dot\sigma_t Z \mid I_t = x]$. The family (7.12) applies unchanged.

7.6 Rectified Flow and Optimal Transport

Straight trajectories and numerical efficiency. For the linear interpolant $I_t = \alpha_t X_0 + \beta_t X_1$ with the rectified flow choice $\alpha_t = 1 - t$, $\beta_t = t$, the conditional velocity is $b_t(x) = \mathbb{E}[X_1 - X_0 \mid I_t = x]$. If trajectories of the ODE (7.4) were perfectly straight — i.e., $b_t(Y_t) = Y_1 - Y_0$ along each trajectory — then the ODE could be solved exactly with a single Euler step. Trajectory crossing prevents this in general: when (X_0, X_1) are drawn independently from $\pi_0 \times \pi_1$, different pairs (X_0, X_1) and (X_0', X_1') can produce paths I_t and I_t' that intersect at some time t, forcing b_t at the crossing point to average over multiple directions.

The reflow procedure. Given a trained velocity field $\hat{b}_t$, one can generate a new *straightened* coupling by simulating the learned ODE: starting from $X_0 \sim \pi_0$, run

$$\frac{dY_t}{dt} = \hat{b}_t(Y_t), \qquad Y_0 = X_0,$$

and set $X_1' := Y_1$. By construction, (X_0, X_1') is a new coupling of π_0 and π_1 (since $Y_1 \sim \pi_1$ when the ODE is exact), and the straight-line paths $(1-t)X_0 + tX_1'$ are precisely the ODE trajectories. These cannot cross, so training a new flow on this coupling produces a velocity field that is closer to a pure translation $b_t(x) \approx X_1' - X_0$ at each point. Iterating this *reflow* [52] procedure

$$(X_0^{(k+1)}, X_1^{(k+1)}) \;:=\; \text{ODE coupling from } \hat{b}_t^{(k)}$$

progressively straightens the trajectories.

Connection to optimal transport. The *Monge optimal transport* (OT) problem between π_0 and π_1 seeks a transport map $T : \mathbb{R}^d \to \mathbb{R}^d$ with $T_{\#}\pi_0 = \pi_1$ minimizing the quadratic cost:

$$\min_{T:\, T_{\#}\pi_0 = \pi_1} \mathbb{E}_{X_0 \sim \pi_0}[\|X_0 - T(X_0)\|_2^2]. \tag{7.13}$$

Under the OT coupling (X_0, X_1^*) where $X_1^* = T(X_0)$, the straight-line paths $(1-t)X_0 + tT(X_0)$ never cross (a consequence of cyclical monotonicity of the OT map), so the velocity $b_t(x) = T(X_0) - X_0$ is a pure translation and the flow ODE can be solved in one step.

The reflow procedure approximates this: each iteration reduces trajectory crossings, and in the limit the coupled pairs $(X_0^{(\infty)}, X_1^{(\infty)})$ converge to the OT coupling. The rectified flow therefore provides a practical algorithm in high dimensions for approximating the OT map without solving (7.13) directly. For related diffusion methods for approximating the entropy regularized OT, namely Schrödinger's bridge, see [65].

Week 4 Exercises

1. (W_2 error bound for flow matching.) Prove the error bound (7.7) by adapting the coupling argument of Theorem 6.4.

(a) *(Setup.)* Let Y_t solve the true ODE $dY_t = b_t(Y_t)\,dt$, $Y_0 \sim \pi_0$, and $\hat{Y}_t$ solve the approximate ODE $d\hat{Y}_t = \hat{b}_t(\hat{Y}_t, \theta)\,dt$, $\hat{Y}_0 \sim \pi_0$. Since both are initialized at the same π_0, one can take $Y_0 = \hat{Y}_0$ (i.e. couple them at initialization). Compute $\frac{d}{dt}\|Y_t - \hat{Y}_t\|_2^2$ and apply the same add-and-subtract and Young/Lipschitz argument as in Theorem 6.4 to obtain

$$\frac{d}{dt}\|Y_t - \hat{Y}_t\|_2^2 \le (2\hat{L}+1)\|Y_t - \hat{Y}_t\|_2^2 + \|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2.$$

(b) *(Gronwall and initialization.)* Apply Gronwall to the inequality in (a) and take expectations to show

$$\mathbb{E}\|Y_1 - \hat{Y}_1\|_2^2 \le e^{(2\hat{L}+1)} \int_0^1 \mathbb{E}\big[\|b_t(Y_t) - \hat{b}_t(Y_t)\|_2^2\big]\,dt.$$

Explain why there is no initialization error term $W_2(\mu_0, \hat{\mu}_0)^2$ here, in contrast to (6.13).

The diffusion model bound (6.13) has a prefactor $e^{(2\hat{L}+1)T}$ that grows with the time horizon $T \to \infty$. The flow matching bound (7.7) has prefactor $e^{(2\hat{L}+1)}$ (with $T = 1$ fixed). Nevertheless, the Lipschitz constant $\hat{L}$ can be large in practice.

2. (Infinite-time interpolant and score-based diffusion models.) This exercise establishes the precise relationship between the flow matching framework and the diffusion model of Section 6.

In Section 7 the interpolant is $I_t = \alpha_t Z + \beta_t X_1$ with $Z \sim \mathcal{N}(0, I_d)$ (source/noise) and $X_1 \sim \pi_1$ (target). For the diffusion model, the source is data $X_0 \sim \pi_0$ and the target is Gaussian noise $Z' \sim \mathcal{N}(0, I_d)$ — the roles are swapped. So set $\pi_0 = \pi$ (data), $\pi_1 = \mathcal{N}(0, I_d)$, and consider the interpolant with the data as source:

$$I_t = \alpha_t X_0 + \beta_t Z', \qquad X_0 \sim \pi_0,\ Z' \sim \mathcal{N}(0, I_d) \text{ independent}, \tag{7.14}$$

with $\alpha_t = e^{-t}$, $\beta_t = \sqrt{1 - e^{-2t}}$, $\alpha_0 = 1$, $\beta_0 = 0$, $\alpha_\infty = 0$, $\beta_\infty = 1$. This matches Theorem 6.2: $I_t \overset{d}{=} X_t$ where X_t solves the forward OU (6.2), so $\mu_t = p_t$.

(a) *(Score in the swapped convention.)* In the convention of (7.14), X_0 plays the role of Z and Z' plays the role of X_1. Apply Theorem 7.6 with $X = I_t$, $Y = X_0$: given $X_0 = x_0$, $I_t \mid X_0 \sim \mathcal{N}(\alpha_t x_0, \beta_t^2 I_d)$, so $\nabla_x \log p_{t|0}(x \mid x_0) = -(x - \alpha_t x_0)/\beta_t^2$. Show that

$$\nabla \log \mu_t(x) = \nabla \log p_t(x) = -\frac{1}{\beta_t}\,\mathbb{E}[Z' \mid I_t = x]. \tag{7.15}$$

This is the analogue of (7.10) with $Z \leftrightarrow X_0$ and $X_1 \leftrightarrow Z'$: the "noise predictor" is now $\mathbb{E}[Z' \mid I_t = x]$.

(b) *(Velocity field b_t and the OU drift.)* The conditional velocity is $b_t(x) = \dot{\alpha}_t \mathbb{E}[X_0 \mid I_t = x] + \dot{\beta}_t \mathbb{E}[Z' \mid I_t = x]$. Using (7.15) and $x = \alpha_t \mathbb{E}[X_0 \mid I_t = x] + \beta_t \mathbb{E}[Z' \mid I_t = x]$, eliminate $\mathbb{E}[X_0 \mid I_t = x]$ and $\mathbb{E}[Z' \mid I_t = x]$ to show

$$b_t(x) = \frac{\dot{\alpha}_t}{\alpha_t}\, x + \beta_t \left(\frac{\dot{\alpha}_t \beta_t}{\alpha_t} - \dot{\beta}_t \right) \nabla \log p_t(x). \tag{7.16}$$

Substituting $\alpha_t = e^{-t}$, $\beta_t = \sqrt{1 - e^{-2t}}$, show that $\dot{\alpha}_t/\alpha_t = -1$ and $\dot{\alpha}_t \beta_t/\alpha_t - \dot{\beta}_t = -\beta_t - e^{-2t}/\beta_t = -(1 - e^{-2t} + e^{-2t})/\beta_t = -1/\beta_t$, giving

$$b_t(x) = -x - \nabla \log p_t(x).$$

(c) *(Recovering the forward OU and reverse SDE.)* Using Theorem 7.8 with $b_t(x) = -x - \nabla \log p_t(x)$:

(i) Show that $g_t = 1$ gives drift $b_t + \nabla \log \mu_t = -x$ and SDE $dY_t = -Y_t\, dt + \sqrt{2}\, dW_t$, which is exactly the forward OU process (6.2). This confirms the family (7.12) contains the forward OU as the $g_t = 1$ member.

(ii) Apply Anderson's time-reversal (Theorem 6.3) to the forward OU process to obtain the reverse SDE (6.4). Since any member of (7.12) shares the marginals p_t, their time-reversals all yield the same marginal law p_{T-t} for $Y_t = X_{T-t}$. In particular, note that the flow matching ODE (7.4) with $b_t(x) = -x - \nabla \log p_t(x)$ has the same marginals p_t as the forward OU SDE. Its time-reversal $\tilde{Y}_s = Y_{T-s}$ satisfies $d\tilde{Y}_s = -b_{T-s}(\tilde{Y}_s)\, ds = (X_{T-s} + \nabla \log p_{T-s}(\tilde{Y}_s))\, ds$, i.e. $d\tilde{Y}_s = (\tilde{Y}_s + \nabla \log p_{T-s}(\tilde{Y}_s))\, ds$. This matches the probability flow ODE (6.17).

As a summary, the forward OU SDE, the flow matching ODE, and the SDE family (7.12) all share marginals p_t; Anderson's time-reversal of any of them with diffusion $\sqrt{2}$ gives the reverse SDE (6.4), while reversing the ODE ($g_t = 0$) gives the probability flow ODE (6.17).

8 Conditional Sampling with Generative Priors

8.1 Motivation and Setting

The preceding sections developed generative models that learn to transport noise to a data distribution π. Given unlabeled data $x^{(1)}, \dots, x^{(N)} \sim \pi$, we trained SDEs/ODEs of the form

$$dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t, \qquad X_0 \sim \mathcal{N}(0, I_d), \quad X_1 \sim \pi,$$

where for example (i) $b_t(x) = x + 2\nabla \log p_t(x)$, $\sigma_t = \sqrt{2}$ (reverse SDE), or (ii) $b_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x]$, $\sigma_t = 0$ (flow matching ODE). In both cases the generative model samples *unconditionally* from π. Note that in case (ii), we can make it SDE by adding score terms in the drift, see Proposition 7.8.

In practice, one often wants to *control* the generated samples. Two canonical settings arise:

① **Bayesian inverse problem.** One has unlabeled data $x^{(1)}, \dots, x^{(N)} \sim \pi$ and a *forward model* $y = F(x) + \sigma\varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$, producing a noisy observation $y \in \mathbb{R}^m$. The goal is to sample the *posterior*

$$P(x \mid y) \propto \pi(x)\,\exp\!\left(-\frac{\|y - F(x)\|_2^2}{2\sigma^2}\right).$$

The likelihood $P(y \mid x) \propto \exp(-\|y - F(x)\|^2/(2\sigma^2))$ is known but $\pi(x)$ is known only through samples.

② **Reward tilting / fine-tuning / inference-time scaling.** One wants to sample from $\pi(x)\exp(R(x; y))$ for a *reward function* $R : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$, where y is a conditioning variable. Intuition: generate samples that sit in the region where the reward R is large while staying close to the prior π. We can define a likelihood $P(y|x) \propto \exp(R(x; y))$.

Both settings reduce to the same problem: given a generative model for π and a likelihood $P(y \mid x)$, sample from $P(x \mid y) \propto \pi(x)\,P(y \mid x)$ for a fixed observation y. We present two alternative approaches with different features: *conditional flow matching* (Setting 1, requires paired training data and retraining) and *guidance* (Setting 2, uses a pre-trained unconditional model with a tilting term at inference time).

8.2 Setting 1: Amortized Conditional Flow Matching

Key idea. Suppose we can generate *paired* samples by first drawing $X_1 \sim \pi$ and then drawing $Y \mid X_1 \sim P(y \mid X_1)$. We use these pairs to train a new conditional velocity field $b_t(x, y)$ that transports noise to $P(\cdot \mid y)$.

Setup. Let $X_0 \sim \mathcal{N}(0, I_d)$ and $X_1 \sim \pi$ be independent, and draw $Y \mid X_1 \sim P(y \mid X_1)$. Consider the linear interpolant $I_t = \alpha_t X_0 + \beta_t X_1$ with $I_0 = X_0$, $I_1 = X_1$. Define the *conditional velocity field*

$$b_t(x, y) := \mathbb{E}\big[\dot{I}_t \mid I_t = x,\, Y = y\big], \qquad \dot{I}_t = \dot{\alpha}_t X_0 + \dot{\beta}_t X_1. \tag{8.1}$$

Let $p_t(x \mid y)$ denote the conditional density of I_t given $Y = y$, so $p_0(\cdot \mid y) = \mathcal{N}(0, I_d)$ for all y (since $I_0 = X_0 \perp Y$) and $p_1(\cdot \mid y) \propto \pi(x)\, P(y \mid x)$.

Proposition 8.1 (Conditional flow matching)**.** *Let $\tilde{X}_t$ solve the ODE*

$$d\tilde{X}_t = b_t(\tilde{X}_t, y)\, dt, \qquad \tilde{X}_0 \sim \mathcal{N}(0, I_d), \tag{8.2}$$

for a fixed y. Then $\mathrm{Law}(\tilde{X}_t) = p_t(\cdot \mid y)$ for all $t \in [0, 1]$. In particular, $\tilde{X}_1 \sim P(\cdot \mid y) \propto \pi(x)\, P(y \mid x)$.

Proof. Fix y and let $\phi \in C_c^\infty(\mathbb{R}^d)$. Differentiating $\int p_t(x \mid y)\, \phi(x)\, dx = \mathbb{E}[\phi(I_t) \mid Y = y]$ in t:

$$\int \partial_t p_t(x \mid y)\, \phi(x)\, dx = \frac{d}{dt}\mathbb{E}[\phi(I_t) \mid Y = y] = \mathbb{E}\big[\nabla\phi(I_t) \cdot \dot{I}_t \mid Y = y\big].$$

By the tower property, conditioning on (I_t, Y) inside the expectation conditioned on $Y = y$:

$$\begin{aligned}
\mathbb{E}[\nabla\phi(I_t) \cdot \dot{I}_t \mid Y = y] &= \mathbb{E}\big[\nabla\phi(I_t) \cdot \mathbb{E}[\dot{I}_t \mid I_t, Y] \mid Y = y\big] \\
&= \int \nabla_x \phi(x) \cdot b_t(x, y)\, p_t(x \mid y)\, dx \\
&= -\int \phi(x)\, \nabla_x \cdot \big(p_t(x \mid y)\, b_t(x, y)\big)\, dx.
\end{aligned}$$

Since ϕ is arbitrary, $p_t(\cdot \mid y)$ satisfies

$$\partial_t p_t(x \mid y) + \nabla_x \cdot \big(p_t(x \mid y)\, b_t(x, y)\big) = 0. \tag{8.3}$$

This is the continuity equation for (8.2) with the same initial condition $p_0(\cdot \mid y) = \mathcal{N}(0, I_d)$, so $\mathrm{Law}(\tilde{X}_t) = p_t(\cdot \mid y)$ for all t. □

Training. The conditional velocity $b_t(x, y)$ minimizes the loss

$$L(\theta) = \int_0^1 \mathbb{E}\Big[\big\|\hat{b}_t(I_t, Y, \theta) - \dot{I}_t\big\|_2^2\Big]\, dt, \tag{8.4}$$

where the expectation is over $X_0 \sim \mathcal{N}(0, I_d)$, $X_1 \sim \pi$, $Y \mid X_1 \sim P(\cdot \mid X_1)$, and $I_t = \alpha_t X_0 + \beta_t X_1$. Training proceeds by first drawing $I_t \mid X_0, X_1$, then $Y \mid X_1$, and regressing $\hat{b}_t(I_t, Y, \theta)$ onto $\dot{I}_t$.

Remark 8.2 (Amortized inference). The trained network $\hat{b}_t(x, y, \theta)$ can be evaluated at any y to generate approximate samples from $P(\cdot \mid y)$ by integrating (8.2). However, the loss (8.4) only guarantees accuracy of $P(x \mid y)$ *averaged over* $Y \sim P(y)$. For a specific fixed observation y, errors may be larger. This is *amortized inference*: one trained model serves all y simultaneously, at the cost of per-instance accuracy.

Remark 8.3 (Classifier-free guidance). In modern generative modeling practice, conditional flow matching is often combined with the unconditional model into a single network $\hat{b}_t(x, y, \theta)$ trained jointly on both conditional and unconditional samples (the latter obtained by replacing Y with a null token $\emptyset$). At inference time, one uses the extrapolated drift

$$(1 + w)\,\hat{b}_t(x, y, \theta) - w\,\hat{b}_t(x, \emptyset, \theta), \qquad w \geq 0,$$

which heuristically amplifies the conditional signal. This is *classifier-free guidance* (Ho–Salimans 2022 [38]) and is a variant of Setting 1 that empirically gives sharper conditional samples for a fixed y at the cost of a bias controlled by the guidance strength w.

In the above setting, the information of $P(y|x)$ is not used directly; it is used based on the simulated many samples Y. This is also referred to as *simulation-based inference*.

8.3 Setting 2: Guidance without Re-training

Setup. Setting 2 takes a pre-trained unconditional generative model for π and steers it toward $P(\cdot \mid y)$ at inference time for a specific fixed observation y, without any retraining.

We work in the flow matching setting of Section 7. By (7.11), the unconditional velocity field satisfies

$$b_t(x) = \frac{\dot{\beta}_t}{\beta_t}\, x + \alpha_t^2 \left(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) \nabla_x \log p_t(x), \tag{8.5}$$

i.e. b_t depends on x through the score $\nabla_x \log p_t(x)$ and the linear term $\frac{\dot{\beta}_t}{\beta_t}\, x$. Combining (8.5) with the SDE family (7.12), the unconditional generative SDE has drift

$$b_t(x) + g_t \nabla_x \log p_t(x) = \frac{\dot{\beta}_t}{\beta_t}\, x + \underbrace{\left(\alpha_t^2 \left(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) + g_t \right)}_{=:\, c_t} \nabla_x \log p_t(x). \tag{8.6}$$

The drift is an affine function of the score with coefficient c_t.

From unconditional to conditional via Bayes. To sample from $p_t(\cdot \mid y)$ instead of $p_t(\cdot)$, we replace $\nabla_x \log p_t(x)$ in (8.6) by $\nabla_x \log p_t(x \mid y)$. Bayes' rule gives

$$\nabla_x \log p_t(x \mid y) = \underbrace{\nabla_x \log p_t(x)}_{\text{prior score (known)}} + \underbrace{\nabla_x \log p_t(y \mid x)}_{\text{likelihood score}}. \tag{8.7}$$

The prior score is available from the trained model. The likelihood score is the difficulty.

The marginal likelihood. The likelihood of observation y given $I_t = x$ is

$$p_t(y \mid x) = \int p_1(y \mid x_1)\, p_t(x_1 \mid x)\, dx_1, \tag{8.8}$$

where $p_1(y \mid x_1) = P(y \mid x_1)$ is the data likelihood and $p_t(x_1 \mid x)$ is the denoising distribution at level t. Computing (8.8) is generally intractable.

Guidance approximation. Approximate $p_t(\cdot \mid x)$ by a point mass at its conditional mean:

$$p_t(y \mid x) \;\approx\; p_1\big(y \mid \mathbb{E}[X_1 \mid I_t = x]\big). \tag{8.9}$$

Here $\mathbb{E}[X_1 \mid I_t = x]$ is the *denoiser* of the prior, which is available from the trained model via Theorem 7.7 (in particular, expressible through the prior score $\nabla_x \log p_t(x)$). Define the *guidance function*

$$f_t(x, y) := \nabla_x \log p_1\big(y \mid \mathbb{E}[X_1 \mid I_t = x]\big). \tag{8.10}$$

Then (8.7) is approximated by $\nabla_x \log p_t(x \mid y) \approx \nabla_x \log p_t(x) + f_t(x, y)$.

Guided generation. Substituting this approximation into (8.6) gives the *guided drift*:

$$\frac{\dot\beta_t}{\beta_t}\, x + c_t\big(\nabla_x \log p_t(x) + f_t(x, y)\big). \tag{8.11}$$

The guided process is

$$d\tilde X_t = \left[\frac{\dot\beta_t}{\beta_t}\, \tilde X_t + c_t\big(\nabla_x \log p_t(\tilde X_t) + f_t(\tilde X_t, y)\big)\right] dt + \sqrt{2g_t}\, dW_t, \quad \tilde X_0 \sim \mathcal{N}(0, I_d). \tag{8.12}$$

When the approximation (8.9) is exact, Theorem 7.8 (applied to the conditional marginals $p_t(\cdot \mid y)$) guarantees $\mathrm{Law}(\tilde X_t) = p_t(\cdot \mid y)$ and $\tilde X_1 \sim P(\cdot \mid y)$.

Remark 8.4 (Diffusion model guidance). For the score-based diffusion model of Section 6 (the reverse SDE (6.4) with explicit drift $y + 2\nabla_x \log p_{T-t}(y)$), the guided version simply replaces the score by the guided score:

$$d\tilde Y_t = \big(\tilde Y_t + 2(\nabla_x \log p_{T-t}(\tilde Y_t) + f_{T-t}(\tilde Y_t, y))\big)\, dt + \sqrt{2}\, dW_t.$$

This is like *classifier guidance* (Dhariwal–Nichol 2021 [25]): the guidance term $f_t(x, y)$ approximates $\nabla_x \log p_t(y \mid x)$ via (8.9) (DPS [23]) or via a separately trained classifier $\hat p_t(y \mid x)$ [25].

Remark 8.5 (Bias of the guidance approximation). The approximation (8.9) introduces a bias: it is exact only when the denoising distribution $p_t(\cdot \mid x)$ is a point mass, which holds at $t = 1$ but not at intermediate t. The bias is most severe at small t, when $p_t(\cdot \mid x)$ is spread over multiple modes and the conditional mean $\mathbb{E}[X_1 \mid I_t = x]$ is a poor summary.

8.4 Discussions and Summary

Several lines of work aim to reduce or remove this bias:

- *Sequential Monte Carlo (SMC), annealed importance sampling (AIS) and twisted particle filtering:* run multiple guided trajectories (particles) carrying importance weights, with periodic resampling to focus computation on high-weight particles.

The "twist" refers to the choice of guided drift that makes the particle proposals closer to the target conditional distribution; (8.10) is one such twist. With proper weighting, SMC yields asymptotically exact samples from $P(\cdot \mid y)$ as the number of particles grows. *This is the dynamical analogue of importance sampling from Section 2*: the guided SDE plays the role of the proposal g, and the importance weights play the role of π/g, but now along an entire path rather than at a single point. The dynamical structure is what makes the proposal close to the target, mitigating the curse of dimensionality that crippled static IS (recall Theorem 2.10).

- *Stochastic optimal control:* formulate conditional sampling as an optimal control problem whose solution is the exact conditional drift, with the guidance term (8.10) arising as a first-order approximation to the optimal control.

These methods trade additional computation at inference time for reduced bias in the conditional samples.

Remark 8.6 (Comparison of the two settings). We summarize as follows.

- *Setting 1 (conditional flow matching):* requires paired training data (X_1, Y) and retraining the model; generalizes over all y in an amortized way; classifier-free guidance is a popular variant that gives sharper conditional samples.
- *Setting 2 (guidance):* uses only a pre-trained unconditional model and a differentiable likelihood $P(y \mid x)$; requires no retraining; works for a specific y at inference time; the basic form has bias controlled by (8.9), which can be reduced by SMC or control-theoretic corrections.

Neither setting strictly dominates the other: Setting 1 invests upfront training cost for fast conditional sampling; Setting 2 invests inference-time cost for flexibility and biased but cheap conditional sampling.

Week 5 Exercises

1. **(Error bound for conditional flow matching.)** This exercise derives an error bound analogous to (7.7) for the conditional flow matching of Section 8.2, making precise the amortization claim that the trained model only guarantees accuracy of $P(\cdot \mid y)$ *averaged over* Y.

   Setup. Recall the conditional velocity field $b_t(x, y) = \mathbb{E}[\dot{I}_t \mid I_t = x, Y = y]$ from (8.1). Let $\hat{b}_t(x, y, \theta)$ be a trained network (not necessarily the minimizer of (8.4)). For each fixed y, let $\tilde{X}_t^y$ solve the true ODE $d\tilde{X}_t^y = b_t(\tilde{X}_t^y, y)\, dt$ with $\tilde{X}_0^y \sim \mathcal{N}(0, I_d)$, and let $\hat{X}_t^y$ solve the approximate ODE $d\hat{X}_t^y = \hat{b}_t(\hat{X}_t^y, y, \theta)\, dt$ with $\hat{X}_0^y \sim \mathcal{N}(0, I_d)$. By Theorem 8.1, $\tilde{X}_1^y \sim P(\cdot \mid y)$. Denote $\hat{P}(\cdot \mid y) = \mathrm{Law}(\hat{X}_1^y)$.

   (a) *(Pointwise bound for fixed y.)* For a fixed observation y, assume $\hat{b}_t(\cdot, y, \theta)$ is $\hat{L}$-Lipschitz in x uniformly in t. Adapt the coupling argument of Theorem 6.4 (with $Y_0 = \hat{Y}_0$ exact, since both ODEs start from $\mathcal{N}(0, I_d)$) to show

   $$W_2^2\big(\hat{P}(\cdot \mid y),\, P(\cdot \mid y)\big) \le e^{(2\hat{L}+1)} \int_0^1 \mathbb{E}\Big[\big\|\hat{b}_t(I_t, y, \theta) - b_t(I_t, y)\big\|_2^2 \,\Big|\, Y = y\Big]\, dt.$$

   (b) *(Bound averaged over Y.)* Take expectation over $Y \sim P(y) = \int \pi(x_1) P(y \mid x_1)\, dx_1$ in part (a) to obtain

   $$\mathbb{E}_Y\big[W_2^2\big(\hat{P}(\cdot \mid Y),\, P(\cdot \mid Y)\big)\big] \le e^{(2\hat{L}+1)} \int_0^1 \mathbb{E}\Big[\big\|\hat{b}_t(I_t, Y, \theta) - b_t(I_t, Y)\big\|_2^2\Big]\, dt. \tag{8.13}$$

   Show that the right-hand side equals $L(\theta) - L^*$, where $L(\theta)$ is the conditional flow matching loss (8.4) and $L^* = \int_0^1 \mathbb{E}\|\dot{I}_t - b_t(I_t, Y)\|^2\, dt$ is the irreducible component of the loss independent of θ. *Hint.* Use the tower property and the fact that $b_t(I_t, Y) = \mathbb{E}[\dot{I}_t \mid I_t, Y]$ is the L^2-projection, so $\mathbb{E}\|\hat{b}_t(I_t, Y, \theta) - \dot{I}_t\|^2 = \mathbb{E}\|\hat{b}_t(I_t, Y, \theta) - b_t(I_t, Y)\|^2 + \mathbb{E}\|b_t(I_t, Y) - \dot{I}_t\|^2$. Conclude that minimizing (8.4) directly controls the Y-*averaged* W_2^2 error between the learned and true posteriors.

   (c) *(Why pointwise accuracy is not guaranteed.)* Explain why the bound (8.13) does *not* imply pointwise accuracy $W_2^2(\hat{P}(\cdot \mid y), P(\cdot \mid y)) \le \varepsilon$ for a specific y. Construct a simple example where the integrated CFM loss is small but the pointwise error is large for some y in a low-probability region of $P(y)$. *Hint.* Take a discrete $Y \in \{y_1, y_2\}$ with $P(Y = y_1) = 1 - \delta$ and $P(Y = y_2) = \delta$ for small δ. If the model has small error for y_1 and large error Δ for y_2, the averaged error is $(1 - \delta) \cdot \text{small} + \delta \cdot \Delta^2$, which is small in expectation even when Δ is large. This is the precise sense in which conditional flow matching is *amortized*: rare observations may receive poor conditional samples.

2. **(Unification of conditional flow matching and guidance.)** This exercise shows that, at the level of exact solutions, Settings 1 and 2 of Section 8 are solving the

same equation: both produce a drift that is the unconditional flow matching formula (8.5) with the unconditional score $\nabla_x \log p_t(x)$ replaced by the conditional score $\nabla_x \log p_t(x \mid y)$. The two settings differ only in *how* they obtain the conditional score.

Throughout, use the linear interpolant $I_t = \alpha_t X_0 + \beta_t X_1$ with $X_0 \sim \mathcal{N}(0, I_d)$, $X_1 \sim \pi$, $Y \mid X_1 \sim P(\cdot \mid X_1)$. Let $p_t(x \mid y)$ denote the conditional density of I_t given $Y = y$.

(a) *(Conditional analogue of Theorem 7.7.)* Conditional on $Y = y$ and $X_1 = x_1$, the interpolant $I_t \mid X_1 = x_1, Y = y$ has the same distribution as $I_t \mid X_1 = x_1$, namely $\mathcal{N}(\beta_t x_1, \alpha_t^2 I_d)$ (since $X_0 \perp Y$ given X_1). Apply Theorem 7.6 with $X = I_t \mid Y = y$ and $W = X_1 \mid Y = y$ to show that the conditional score satisfies

$$\nabla_x \log p_t(x \mid y) = -\frac{1}{\alpha_t^2} \mathbb{E}[\alpha_t X_0 \mid I_t = x, Y = y]. \tag{8.14}$$

(b) *(Conditional velocity in terms of the conditional score.)* Following the derivation of (7.11), starting from $b_t(x, y) = \mathbb{E}[\dot{I}_t \mid I_t = x, Y = y] = \dot{\alpha}_t \mathbb{E}[X_0 \mid I_t = x, Y = y] + \dot{\beta}_t \mathbb{E}[X_1 \mid I_t = x, Y = y]$, and using (8.14) together with $x = \alpha_t \mathbb{E}[X_0 \mid I_t = x, Y = y] + \beta_t \mathbb{E}[X_1 \mid I_t = x, Y = y]$, derive

$$b_t(x, y) = \frac{\dot{\beta}_t}{\beta_t} x + \alpha_t^2 \left(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) \nabla_x \log p_t(x \mid y). \tag{8.15}$$

This is exactly (8.5) with the unconditional score $\nabla_x \log p_t(x)$ replaced by the conditional score $\nabla_x \log p_t(x \mid y)$.

(c) *(Reconciling Settings 1 and 2.)* Combine (8.15) with the SDE family Theorem 7.8 (applied to the conditional marginals $p_t(\cdot \mid y)$): for any $g_t \geq 0$, the SDE

$$d\tilde{X}_t = \big(b_t(\tilde{X}_t, y) + g_t \nabla_x \log p_t(\tilde{X}_t \mid y)\big)\, dt + \sqrt{2 g_t}\, dW_t, \qquad \tilde{X}_0 \sim \mathcal{N}(0, I_d),$$

has $\mathrm{Law}(\tilde{X}_t) = p_t(\cdot \mid y)$.

(i) *(Setting 1 / CFM.)* Setting $g_t = 0$ recovers the conditional flow matching ODE (8.2). The conditional score $\nabla_x \log p_t(x \mid y)$ enters *implicitly* through $b_t(x, y)$, which is regressed onto $\dot{I}_t$ in the loss (8.4).

(ii) *(Setting 2 / guidance.)* Apply Bayes' rule (8.7) to the conditional score in (8.15): $\nabla_x \log p_t(x \mid y) = \nabla_x \log p_t(x) + \nabla_x \log p_t(y \mid x)$. Substituting and rearranging:

$$b_t(x, y) = \underbrace{b_t(x)}_{\text{unconditional drift}} + \alpha_t^2 \left(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) \underbrace{\nabla_x \log p_t(y \mid x)}_{\text{likelihood score}}.$$

Compare with (8.11): this is exactly the guided drift, with the likelihood score appearing as the additive "guidance" correction to the unconditional drift.

Conclusion. Settings 1 and 2 produce identical *exact* drifts; they differ only in implementation:

- Setting 1 estimates the full conditional score $\nabla_x \log p_t(x \mid y)$ directly via regression onto $\dot{I}_t$ from paired data (X_1, Y).
- Setting 2 decomposes the conditional score by Bayes into the (known) prior score plus the likelihood score, and approximates the latter using (8.9).

The two approaches incur different errors: Setting 1 has amortization error (Exercise 1); Setting 2 has the bias of (8.9). SMC-style methods correct the latter using the dynamical-IS perspective discussed in Section 8.3.

9 Stochastic Optimal Control for Conditional Sampling

9.1 Motivation

In Section 8.3 we approximated the conditional score $\nabla_x \log p_t(x \mid y)$ by Bayes' rule plus the guidance approximation (8.9), which introduces a bias. We now present a more principled approach via *stochastic optimal control* (SOC), which yields the *exact* conditional drift as the solution of an optimization problem (under certain assumptions) and in such a setting, (8.10) can be understood as a approximation of the control problem.

Setup. We have a pre-trained unconditional generative SDE

$$dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t, \qquad X_0 \sim \pi_0 = \mathcal{N}(0, I_d), \tag{9.1}$$

with $\mathrm{Law}(X_1) = \pi_1$ (the data distribution π). Examples include the reverse SDE (6.4) (after time-reversal) and the SDE family (7.12). The *real goal* is to sample from the tilted distribution

$$\pi_1^R(x) \propto \pi_1(x)\,\exp\big(R(x)\big), \tag{9.2}$$

where $R : \mathbb{R}^d \to \mathbb{R}$ is a reward (or the negative log-likelihood $\log P(y \mid x)$ in Bayesian inference for a fixed observation y). By itself, (9.1) samples π_1, not π_1^R.

Steering via a control. Add a control term $u_t : \mathbb{R}^d \to \mathbb{R}^d$ to the drift and consider the *controlled SDE*:

$$dX_t^u = \big(b_t(X_t^u) + \sigma_t\,u_t(X_t^u)\big)\,dt + \sigma_t\,dW_t, \qquad X_0^u \sim \pi_0, \tag{9.3}$$

parameterized so that taking $u_t \equiv 0$ recovers (9.1). The question becomes: what control u_t steers (9.3) so that $\mathrm{Law}(X_1^u) = \pi_1^R$? The guidance correction $f_t(x, y)$ in (8.10) is one approximation answer; SOC provides a principled framework.

9.2 The Stochastic Optimal Control Problem

Cost functional. Consider the SOC problem

$$\min_{u_t} \int_0^1 \tfrac{1}{2}\,\mathbb{E}\big[\|u_t(X_t^u)\|_2^2\big]\,dt \;-\; \mathbb{E}[R(X_1^u)] \quad \text{s.t. (9.3)}, \tag{9.4}$$

which combines a *running cost* (the L^2 control energy) with a *terminal reward* $R(X_1^u)$.

Why this cost? The running cost penalizes deviation from the uncontrolled process, keeping X_t^u close in distribution to X_t; the terminal cost rewards trajectories ending in high-R regions. We will see in Section 9.5 that $\frac{1}{2}\int_0^1 \mathbb{E}\|u_t\|_2^2\,dt$ is exactly $D_{\mathrm{KL}}(\mathbb{P}^u \,\|\, \mathbb{P})$, the path-space KL divergence between the controlled and uncontrolled measures. The SOC problem (9.4) is therefore a KL-regularized maximization of $\mathbb{E}[R]$.

9.3 Value Function and Dynamic Programming

Value function. Define the cost-to-go starting from state x at time t:

$$V_t(x) := \min_u \mathbb{E}\left[\tfrac{1}{2}\int_t^1 \|u_s(X_s^u)\|_2^2\, ds - R(X_1^u)\,\middle|\, X_t^u = x\right], \tag{9.5}$$

so that $V_1(x) = -R(x)$ is the terminal condition.

Proposition 9.1 (Dynamic programming)**.** *For $\Delta t > 0$,*

$$V_t(x) = \min_{u_t} \mathbb{E}\left[\tfrac{1}{2}\int_t^{t+\Delta t} \|u_s(X_s^u)\|_2^2\, ds + V_{t+\Delta t}(X_{t+\Delta t}^u)\,\middle|\, X_t^u = x\right]. \tag{9.6}$$

The proof is the standard splitting of the cost integral at $t+\Delta t$ and using the Markov property of the controlled SDE.

9.4 The Hamilton–Jacobi–Bellman Equation

Heuristic derivation. Send $\Delta t \to 0$ in (9.6). Expanding $V_{t+\Delta t}(X_{t+\Delta t}^u)$ via Itô's formula along the controlled SDE (9.3):

$$V_{t+\Delta t}(X_{t+\Delta t}^u) \approx V_t(x) + \partial_t V_t(x)\,\Delta t + \nabla V_t(x)\cdot \Delta x + \tfrac{1}{2}\sigma_t^2\,\Delta V_t(x)\,\Delta t + O\big((\Delta t)^2\big),$$

where $\Delta x = (b_t(x) + \sigma_t u_t(x))\Delta t + \sigma_t\,\Delta W_t$ and $\mathbb{E}[\Delta x] = (b_t + \sigma_t u_t)\Delta t$. Substituting into (9.6) and dividing by Δt:

$$0 = \min_{u_t}\big[\tfrac{1}{2}\|u_t(x)\|_2^2 + \partial_t V_t(x) + \nabla V_t(x)\cdot\big(b_t(x) + \sigma_t u_t(x)\big) + \tfrac{1}{2}\sigma_t^2\,\Delta V_t(x)\big]. \tag{9.7}$$

The minimization over u_t is pointwise and quadratic, with minimizer

$$u_t^*(x) = -\sigma_t\,\nabla V_t(x). \tag{9.8}$$

Substituting (9.8) back into (9.7) eliminates the minimization and gives the *Hamilton–Jacobi–Bellman* (HJB) equation.

Proposition 9.2 (Optimal control and HJB)**.** *The optimal control is $u_t^*(x) = -\sigma_t\,\nabla V_t(x)$, and the value function V_t satisfies the HJB equation*

$$\partial_t V_t(x) + b_t(x)\cdot\nabla V_t(x) - \tfrac{1}{2}\sigma_t^2\|\nabla V_t(x)\|_2^2 + \tfrac{1}{2}\sigma_t^2\,\Delta V_t(x) = 0, \qquad V_1(x) = -R(x). \tag{9.9}$$

Remark 9.3. The HJB equation is a *nonlinear* PDE due to the $\|\nabla V_t\|^2$ term, reflecting the optimization over u. The Cole–Hopf transformation $V_t(x) = -\log \Phi_t(x)$ linearizes it: Φ_t then satisfies a linear (backward) heat equation, the Feynman–Kac representation of which connects Φ_t to the unconditional process (9.1). This linearization is the foundation of the SMC and twisted particle methods discussed in Section 8.3.

9.5 Characterizing the Distributions via Girsanov's Theorem

The remaining question is: what is the law of $X_1^{u^*}$ under the optimal control? We answer it via the Girsanov change of measure.

Path measures. Let $\mathbb{P}$ denote the path measure of the uncontrolled process $\{X_s : 0 \le s \le 1\}$ from (9.1), and $\mathbb{P}^u$ the path measure of the controlled process $\{X_s^u : 0 \le s \le 1\}$ from (9.3).

Discrete-time intuition. Discretize: $X_{t_{n+1}} = X_{t_n} + b_{t_n}\Delta t + \sigma_{t_n}(W_{t_{n+1}} - W_{t_n})$ with grid $0 = t_0 < \cdots < t_N = 1$. The Markov factorization gives

$$\mathbb{P}(X) \approx p(X_{t_0}) \prod_k p(X_{t_k} \mid X_{t_{k-1}}), \qquad \mathbb{P}^u(X) \approx p(X_{t_0}) \prod_k p^u(X_{t_k} \mid X_{t_{k-1}}),$$

where the conditional densities are Gaussian:

$$p(X_{t_k} \mid X_{t_{k-1}}) = \frac{1}{(2\pi\sigma_{t_{k-1}}^2 \Delta t)^{d/2}} \exp\left(-\frac{\|X_{t_k} - X_{t_{k-1}} - b_{t_{k-1}}\Delta t\|^2}{2\sigma_{t_{k-1}}^2 \Delta t} \right),$$
$$p^u(X_{t_k} \mid X_{t_{k-1}}) = \frac{1}{(2\pi\sigma_{t_{k-1}}^2 \Delta t)^{d/2}} \exp\left(-\frac{\|X_{t_k} - X_{t_{k-1}} - (b_{t_{k-1}} + \sigma_{t_{k-1}} u_{t_{k-1}})\Delta t\|^2}{2\sigma_{t_{k-1}}^2 \Delta t} \right).$$

Computing the log-ratio and summing:

$$\log \frac{\mathbb{P}^u}{\mathbb{P}}(X) \approx \sum_k \Big[-\tfrac{1}{2}\|u_{t_{k-1}}\|_2^2 \, \Delta t + \tfrac{1}{\sigma_{t_{k-1}}}(X_{t_k} - X_{t_{k-1}} - b_{t_{k-1}}\Delta t) \cdot u_{t_{k-1}} \Big].$$

Continuous-time limit. Sending $\Delta t \to 0$ formally, the second sum in the discrete log-ratio becomes a path integral involving dX_t:

$$\log \frac{d\mathbb{P}^u}{d\mathbb{P}}(X) = -\tfrac{1}{2} \int_0^1 \|u_t(X_t)\|_2^2 \, dt + \int_0^1 \frac{1}{\sigma_t} \, (dX_t - b_t(X_t)\, dt) \cdot u_t(X_t). \tag{9.10}$$

This is the Radon–Nikodym density of $\mathbb{P}^u$ with respect to $\mathbb{P}$, evaluated on a path $X_{[0,1]}$. The expression (9.10) makes no reference to *which* measure the path X is sampled from; it is a function of the path. The interpretation is determined by the measure under which we take expectations.

Evaluating along the controlled process. We pick $X = X^u$ as the controlled process with law $\mathbb{P}^u$, satisfying (9.3). Then $dX_t^u - b_t(X_t^u)\, dt = \sigma_t u_t(X_t^u)\, dt + \sigma_t \, dW_t$, where W_t is a Brownian motion under $\mathbb{P}^u$. Substituting into (9.10):

$$\frac{1}{\sigma_t}(dX_t^u - b_t(X_t^u)\, dt) \cdot u_t(X_t^u) = \|u_t(X_t^u)\|_2^2 \, dt + u_t(X_t^u) \cdot dW_t,$$

so

$$\log \frac{d\mathbb{P}^u}{d\mathbb{P}}(X^u) = \tfrac{1}{2}\int_0^1 \|u_t(X_t^u)\|_2^2\, dt + \int_0^1 u_t(X_t^u)\cdot dW_t. \tag{9.11}$$

Taking expectation under $\mathbb{P}^u$, the stochastic integral $\int_0^1 u_t \cdot dW_t$ has mean zero (Itô integral against a $\mathbb{P}^u$-Brownian motion):

Theorem 9.4 (Girsanov / KL of path measures)**.** *Under suitable integrability conditions (e.g., Novikov), the Radon–Nikodym density of $\mathbb{P}^u$ with respect to $\mathbb{P}$ is given by (9.10), and*

$$D_{\mathrm{KL}}(\mathbb{P}^u \,\|\, \mathbb{P}) = \mathbb{E}^{\mathbb{P}^u}\left[\log \frac{d\mathbb{P}^u}{d\mathbb{P}}(X^u)\right] = \tfrac{1}{2}\int_0^1 \mathbb{E}^{\mathbb{P}^u}\left[\|u_t(X_t^u)\|_2^2\right] dt. \tag{9.12}$$

Remark 9.5 (Evaluating along the uncontrolled process). Alternatively, picking X as the uncontrolled process under $\mathbb{P}$ (so $dX_t - b_t(X_t)\,dt = \sigma_t\, dW_t$) gives

$$\log \frac{d\mathbb{P}^u}{d\mathbb{P}}(X) = -\tfrac{1}{2}\int_0^1 \|u_t(X_t)\|_2^2\, dt + \int_0^1 u_t(X_t)\cdot dW_t,$$

with sign opposite to (9.11). This form is useful for importance-sampling computations: to estimate $\mathbb{E}^{\mathbb{P}^u}[\Phi(X)]$ by sampling from $\mathbb{P}$, reweight by $d\mathbb{P}^u/d\mathbb{P}$ evaluated along uncontrolled paths.

9.6 Connection to Sampling: KL-Control Duality

KL-control duality. Combining the SOC problem (9.4) with the path-KL formula (9.12):

$$\min_u \left[D_{\mathrm{KL}}(\mathbb{P}^u \,\|\, \mathbb{P}) - \mathbb{E}^{\mathbb{P}^u}[R(X_1^u)]\right] = \min_u D_{\mathrm{KL}}\left(\mathbb{P}^u \,\|\, \mathbb{P}^R\right) + \text{const}, \tag{9.13}$$

where $\mathbb{P}^R$ is the path measure characterized by the Radon–Nikodym density

$$\frac{d\mathbb{P}^R}{d\mathbb{P}}(X_{[0,1]}) = \frac{e^{R(X_1)}}{\mathbb{E}_{\mathbb{P}}[e^{R(X_1)}]}. \tag{9.14}$$

The terminal marginal of $\mathbb{P}^R$ is precisely the desired tilted target:

$$\mathrm{Law}_{\mathbb{P}^R}(X_1) \propto \pi_1(x)\, e^{R(x)} = \pi_1^R(x). \tag{9.15}$$

Thus, SOC can be thought of as KL optimization in the path space, namely variational inference in the path space.

Subtlety: the initial-distribution constraint. However, the controlled SDE (9.3) *constrains* $X_0^u \sim \pi_0 = \mathcal{N}(0, I_d)$, and the SOC problem (9.4) therefore minimizes (9.13) *only over* path measures $\mathbb{P}^u$ with this fixed initial marginal. The path measure $\mathbb{P}^R$ in (9.14) does *not* have $X_0 \sim \mathcal{N}(0, I_d)$ in general: the tilting by $e^{R(X_1)}$ couples X_1 to X_0 through the joint $\mathbb{P}$, so its X_0-marginal is

$$\mathrm{Law}_{\mathbb{P}^R}(X_0) \propto \pi_0(x_0)\, \mathbb{E}_{\mathbb{P}}[e^{R(X_1)} \mid X_0 = x_0],$$

which differs from π_0 unless $\mathbb{E}_{\mathbb{P}}[e^{R(X_1)} \mid X_0]$ is constant in X_0 — typically not the case, since X_0 is informative about X_1 under the generative dynamics.

Consequence. The constrained SOC problem therefore does *not* produce $\mathbb{P}^u = \mathbb{P}^R$, and the law of $X_1^{u^*}$ under the constrained optimal control is in general *not exactly* π_1^R. This work of Domingo-Enrich, Drozdzal, Karrer & Chen (2024) [28] show that exact sampling from π_1^R requires a special noise schedule ("memoryless" noise) so that X_0 becomes informationally decoupled from X_1 along the controlled path; only then does the constrained optimum coincide with $\mathbb{P}^R$ and yield $\mathrm{Law}(X_1^{u^*}) = \pi_1^R$. Without this modification, the SOC solution provides a high-quality but slightly biased approximation to π_1^R.

Remark 9.6 (Memoryless noise schedule). The fix is to choose σ_t large enough so that the controlled process forgets X_0 before reaching $t = 1$. Concretely, when the unconditional SDE (9.1) is constructed so that $\mathbb{E}_{\mathbb{P}}[e^{R(X_1)} \mid X_0 = x_0]$ is independent of x_0 (the memoryless condition), the constrained KL-minimization attains $\mathbb{P}^u = \mathbb{P}^R$ and the SOC solution samples exactly from π_1^R. See [28] for the precise formulation.

We note that for score-based diffusion models based on time reversal of Langevin, the SDE automatically uses the memoryless schedule due to the ergodicity of the forward noising diffusion, which implies that the noise and target are independently coupled.

Connection to guidance. The optimal drift in (9.3) under u^* is

$$b_t(x) + \sigma_t u_t^*(x) = b_t(x) - \sigma_t^2 \, \nabla V_t(x),$$

which has the same affine-in-correction structure as the guided drift (8.11): the guidance term $f_t(x, y)$ plays the role of $-\sigma_t \nabla V_t(x)$ (up to the SDE-family prefactor), and the guidance approximation (8.9) corresponds to taking

$$V_t(x) \approx -\log p_1\big(y \mid \mathbb{E}[X_1 \mid I_t = x]\big).$$

9.7 Pathwise Gradient and the Adjoint Equation

There are two natural strategies for solving the SOC problem (9.4):

(i) *Solve the optimality condition directly.* The HJB equation (9.9) is the optimality condition for the SOC problem; given V_t, the optimal control is recovered as $u_t^*(x) = -\sigma_t \nabla V_t(x)$. This requires solving a nonlinear PDE in $\mathbb{R}^d$.

(ii) *Optimize the objective directly.* Parameterize the control as u_θ and minimize the SOC cost $J(\theta)$ by stochastic gradient descent. This requires computing $\nabla_\theta J(\theta)$ along simulated trajectories.

This subsection develops the second route. The key tool is the *adjoint equation*, which collapses backpropagation through a simulated trajectory into a single backward ODE and yields gradients of J with respect to either u (functional gradient) or θ (parameter gradient).

Roadmap.

- Section 9.7.1: derive the adjoint recursion in discrete time; this is just reverse-mode autodiff through Euler–Maruyama, transparent and intuitive.
- Section 9.7.2: take the formal continuum limit to obtain the pathwise adjoint ODE.
- Sections 9.7.3 and 9.7.4: use the adjoint to express the functional gradient $\delta J/\delta u$ and the parameter gradient $\nabla_\theta J$.
- Section 9.7.5: discuss *adjoint matching* [28], a recent regression-based simplification that uses a *lean* adjoint (dropping all u-dependent terms from the adjoint ODE to simplify it) and converts the SOC problem into a least-squares loss.

Generalized cost. Throughout this subsection we work with the slightly more general SOC cost

$$J(u) := \mathbb{E}\left[\int_0^1 \big(\tfrac{1}{2}\|u_t(X_t^u)\|_2^2 + f_t(X_t^u)\big)dt + g(X_1^u)\right], \tag{9.16}$$

allowing a state-dependent running cost f_t and terminal cost g. The setup of (9.4) is the special case $f \equiv 0$ and $g = -R$.

Notation. We use lowercase x for a generic state argument (so $u_t(x)$ is the control as a function of x and t) and capital X_t for the random state along a trajectory. Gradients $\nabla u_t(x)$, $\nabla b_t(x)$, $\nabla f_t(x)$, etc. denote derivatives with respect to x; when evaluated at the current trajectory we write $\nabla u_t(X_t)$ etc.

9.7.1 Discrete Pathwise Adjoint

Discretize (9.3) on the grid $0 = t_0 < t_1 < \cdots < t_N = 1$ with $\Delta t = 1/N$:

$$X_{k+1} = X_k + \big(b_k(X_k) + \sigma_k\, u_k(X_k)\big)\Delta t + \sigma_k\sqrt{\Delta t}\,\xi_k, \qquad \xi_k \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d), \tag{9.17}$$

with feedback control $u_k(\cdot) = u(\cdot, t_k)$ and shorthand $b_k = b_{t_k}$, $\sigma_k = \sigma_{t_k}$, $f_k = f_{t_k}$.

Pathwise adjoint. Freeze the noise sequence $\{\xi_k\}_{k=0}^{N-1}$. Along this fixed path, the trajectory $\{X_j\}_{j\geq k}$ is a deterministic function of the state X_k at step k via the recursion (9.17). Hence the future cost

$$\mathcal{C}_k(x) := \sum_{j=k}^{N-1} \big(\tfrac{1}{2}\|u_j(X_j(x))\|^2 + f_j(X_j(x))\big)\Delta t + g(X_N(x))$$

is a deterministic function of the initial state x at step k, where $X_j(x)$ denotes the trajectory started from $X_k = x$ along the fixed noise sequence. Define the *pathwise adjoint* as its gradient evaluated at the actual state:

$$a_k := \nabla \mathcal{C}_k(X_k). \tag{9.18}$$

Equivalently, a_k is the sensitivity of the path-wise tail cost to a perturbation of X_k.

Backward recursion. Introduce the per-step running cost $\ell_t(x) := \frac{1}{2}\|u_t(x)\|^2 + f_t(x)$. Splitting off the k-th term and applying the chain rule,

$$\mathcal{C}_k(x) = \ell_k(x)\,\Delta t + \mathcal{C}_{k+1}(F_k(x)),$$

where $F_k(x) := x + (b_k(x) + \sigma_k u_k(x))\Delta t + \sigma_k\sqrt{\Delta t}\,\xi_k$ is the one-step Euler map. Differentiating in x and evaluating at X_k:

$$a_k = \nabla\ell_k(X_k)\,\Delta t + \nabla F_k(X_k)^\top\, a_{k+1}. \tag{9.19}$$

The one-step Jacobian, from F_k, is

$$\nabla F_k(X_k) = I + \big(\nabla b_k(X_k) + \sigma_k\,\nabla u_k(X_k)\big)\Delta t, \tag{9.20}$$

where the noise term $\sigma_k\sqrt{\Delta t}\,\xi_k$ contributes *nothing* because σ_k is state-independent. Substituting into (9.19):

$$\boxed{a_k = \nabla\ell_k(X_k)\,\Delta t + \big(I + (\nabla b_k(X_k) + \sigma_k\nabla u_k(X_k))\Delta t\big)^\top a_{k+1}, \qquad a_N = \nabla g(X_N).} \tag{9.21}$$

This recursion is precisely reverse-mode automatic differentiation through the Euler–Maruyama scheme (9.17).

9.7.2 Continuous Pathwise Adjoint ODE

Rearranging (9.21) as

$$\frac{a_{k+1} - a_k}{\Delta t} = -\big[(\nabla b_k(X_k) + \sigma_k\nabla u_k(X_k))^\top a_{k+1} + \nabla\ell_k(X_k)\big] + O(\Delta t),$$

and sending $\Delta t \to 0$:

Proposition 9.7 (Pathwise adjoint ODE)**.** *Along each Brownian path of (9.3), the pathwise adjoint*

$$a_t := \nabla_x\left[\int_t^1 \ell_s(X_s(x))\,ds + g(X_1(x))\right]\Big|_{x=X_t}, \qquad \ell_s(x) := \tfrac{1}{2}\|u_s(x)\|^2 + f_s(x),$$

where $X_s(x)$ denotes the trajectory started from $X_t = x$ along the fixed Brownian path, satisfies the linear backward ODE

$$\frac{da_t}{dt} = -\big[(\nabla b_t(X_t) + \sigma_t\nabla u_t(X_t))^\top a_t + \nabla\ell_t(X_t)\big], \qquad a_1 = \nabla g(X_1). \tag{9.22}$$

Remark 9.8 (Pathwise vs. adapted: BSDE form). By construction, a_t is $\mathcal{F}_1$-measurable but *not* $\mathcal{F}_t$-adapted: it depends on the entire future Brownian path through the trajectory $\{X_s\}_{s\in[t,1]}$. Its $\mathcal{F}_t$-projection

$$\hat{a}_t := \mathbb{E}[a_t \mid \mathcal{F}_t] = \nabla_x J(u; X_t, t), \tag{9.23}$$

where $J(u;x,t)$ is the cost functional from (9.5) with the min removed, is adapted (Markov property + interchange of ∇ and $\mathbb{E}$). Applying Itô's formula to $\nabla_x J(u;X_t,t)$ along the controlled SDE, using the linear Feynman–Kac PDE for J, yields the *backward stochastic differential equation* (BSDE)

$$d\hat{a}_t = -\big[(\nabla b_t(X_t) + \sigma_t \nabla u_t(X_t))^\top \hat{a}_t + \nabla \ell_t(X_t)\big]\,dt + Z_t\,dW_t, \qquad \hat{a}_1 = \nabla g(X_1), \quad (9.24)$$

where the martingale coefficient Z_t is determined by the martingale representation theorem. The drift agrees with the pathwise ODE (9.22); the $Z_t\,dW_t$ piece is the price of forcing adaptedness. The pathwise ODE (9.22) is what one *implements* via backpropagation through a single simulated trajectory; the BSDE (9.24) is the form for theoretical analysis (FBSDE methods).

9.7.3 Functional Gradient

We use the pathwise adjoint to compute the gradient of the SOC cost (9.16) with respect to u as a function. Perturb $u \to u + \varepsilon\,\delta u$ and let X_t^ε denote the controlled trajectory under the perturbed control. The first-order trajectory variation $\delta X_t := \partial_\varepsilon X_t^\varepsilon|_{\varepsilon=0}$ satisfies the linearized SDE

$$d(\delta X_t) = \big(\nabla b_t(X_t) + \sigma_t \nabla u_t(X_t)\big)\delta X_t\,dt + \sigma_t\,\delta u_t(X_t)\,dt, \qquad \delta X_0 = 0, \quad (9.25)$$

with no Itô term, since σ_t is state-independent. Differentiating (9.16) at $\varepsilon = 0$ (recall $\ell_t(x) = \frac{1}{2}\|u_t(x)\|^2 + f_t(x)$):

$$\left.\frac{dJ}{d\varepsilon}\right|_0 = \mathbb{E}\left[\int_0^1 \langle u_t(X_t), \delta u_t(X_t)\rangle\,dt + \int_0^1 \nabla \ell_t(X_t)^\top \delta X_t\,dt + \nabla g(X_1)^\top \delta X_1\right]. \quad (9.26)$$

The first integral is the *direct* variation through u_t at fixed trajectory; the remaining two are the *indirect* variation through δX_t.

Adjoint identity. The dynamics (9.22) are designed so that the indirect contribution collapses. Compute, using (9.22) and (9.25):

$$\begin{aligned}
\frac{d}{dt}(a_t^\top \delta X_t) &= \dot{a}_t^\top \delta X_t + a_t^\top\,\dot{\delta X}_t \\
&= -\big[(\nabla b_t(X_t) + \sigma_t \nabla u_t(X_t))^\top a_t + \nabla \ell_t(X_t)\big]^\top \delta X_t \\
&\quad + a_t^\top\big[(\nabla b_t(X_t) + \sigma_t \nabla u_t(X_t))\delta X_t + \sigma_t\,\delta u_t(X_t)\big] \\
&= -\nabla \ell_t(X_t)^\top \delta X_t + a_t^\top \sigma_t\,\delta u_t(X_t),
\end{aligned}$$

where the $(\nabla b + \sigma \nabla u)$-terms cancel between the adjoint drift and the linearized state. Integrating from 0 to 1 with $\delta X_0 = 0$ and $a_1 = \nabla g(X_1)$:

$$\nabla g(X_1)^\top \delta X_1 + \int_0^1 \nabla \ell_t(X_t)^\top \delta X_t\,dt = \int_0^1 \langle \sigma_t^\top a_t,\, \delta u_t(X_t)\rangle\,dt. \quad (9.27)$$

The left side is the indirect contribution in (9.26). Substituting:

Proposition 9.9 (Functional gradient)**.** *The first variation of J along an admissible perturbation δu is*

$$\left.\frac{dJ}{d\varepsilon}\right|_0 = \mathbb{E}\left[\int_0^1 \big\langle u_t(X_t) + \sigma_t^\top a_t,\ \delta u_t(X_t)\big\rangle\, dt\right], \tag{9.28}$$

where a_t is the pathwise adjoint (9.22). Equivalently, in terms of the law ρ_t^u of X_t^u,

$$\frac{\delta J}{\delta u}(x,t) = \rho_t^u(x)\,\big[u_t(x) + \sigma_t^\top \mathbb{E}[a_t \mid X_t = x]\big].$$

Remark 9.10 (Recovering HJB optimality). The stationarity condition $\delta J/\delta u = 0$ reads

$$u_t^*(x) = -\sigma_t^\top \mathbb{E}[a_t \mid X_t = x] = -\sigma_t^\top \nabla_x J(u^*; x, t) = -\sigma_t^\top \nabla V_t(x),$$

using (9.23) and $\nabla J(u^*;\cdot) = \nabla V$ at the optimum. This recovers the HJB optimality formula (9.8) as a function-space stationarity condition, complementing the dynamic programming derivation of Theorem 9.2.

9.7.4 Parameter Gradient

For numerical optimization, parameterize the control by a neural network $u_t(x,\theta)$ with parameters $\theta \in \mathbb{R}^p$. The SOC cost becomes $J(\theta) := J(u_t(\cdot,\theta))$, and we want $\nabla_\theta J$. The derivation parallels the functional-gradient case.

Linearized trajectory. Perturb $\theta \to \theta + \varepsilon\,\delta\theta$ and let X_t^ε be the trajectory under the perturbed control $u_t(\cdot, \theta + \varepsilon\delta\theta)$. Linearizing (9.3) in ε, the trajectory variation $\delta X_t := \partial_\varepsilon X_t^\varepsilon|_{\varepsilon=0}$ satisfies

$$d(\delta X_t) = \big(\nabla b_t(X_t) + \sigma_t \nabla_x u_t(X_t,\theta)\big)\delta X_t\, dt + \sigma_t\, \partial_\theta u_t(X_t,\theta)\,\delta\theta\, dt, \qquad \delta X_0 = 0, \tag{9.29}$$

where $\nabla_x u_t(x,\theta) \in \mathbb{R}^{d\times d}$ is the state Jacobian of $u_t(\cdot,\theta)$ at fixed θ, and $\partial_\theta u_t(x,\theta) \in \mathbb{R}^{d\times p}$ is the parameter Jacobian at fixed (x,t).

Cost variation. The control cost $\frac{1}{2}\|u_t(X_t^\varepsilon,\theta)\|^2$ depends on ε both through X_t^ε *and* through $u_t(\cdot,\theta)$ explicitly. By the chain rule, evaluating at X_t,

$$\left.\frac{d}{d\varepsilon}\right|_0 \big[\tfrac{1}{2}\|u_t(X_t^\varepsilon,\theta)\|^2\big] = u_t(X_t,\theta)^\top\big[\nabla_x u_t(X_t,\theta)\,\delta X_t + \partial_\theta u_t(X_t,\theta)\,\delta\theta\big],$$

where the first piece equals $\nabla_x\big(\frac{1}{2}\|u_t(x,\theta)\|^2\big)\big|_{x=X_t}^\top \delta X_t$. With $\ell_t(x) := \frac{1}{2}\|u_t(x,\theta)\|^2 + f_t(x)$,

$$\left.\frac{dJ}{d\varepsilon}\right|_0 = \underbrace{\mathbb{E}\left[\int_0^1 \nabla\ell_t(X_t)^\top \delta X_t\, dt + \nabla g(X_1)^\top \delta X_1\right]}_{\text{indirect (through trajectory)}} + \underbrace{\mathbb{E}\left[\int_0^1 u_t(X_t,\theta)^\top \partial_\theta u_t(X_t,\theta)\,\delta\theta\, dt\right]}_{\text{direct (through } u_t\text{)}}. \tag{9.30}$$

The indirect term can be handled by the same identity as in Theorem 9.9, with δX_t now driven by $\delta\theta$ via (9.29) instead of by δu via (9.25). Compute, using (9.22) and (9.29):

$$\frac{d}{dt}(a_t^\top \delta X_t) = -\nabla \ell_t(X_t)^\top \delta X_t + a_t^\top \sigma_t \, \partial_\theta u_t(X_t, \theta)\, \delta\theta,$$

once again with the $(\nabla b_t + \sigma_t \nabla_x u_t)$-terms cancelling. Integrating with $\delta X_0 = 0$ and $a_1 = \nabla g(X_1)$:

$$\nabla g(X_1)^\top \delta X_1 + \int_0^1 \nabla \ell_t(X_t)^\top \delta X_t \, dt = \int_0^1 a_t^\top \sigma_t \, \partial_\theta u_t(X_t, \theta)\, \delta\theta \, dt,$$

which is exactly the indirect term in (9.30). Substituting and reading off the coefficient of $\delta\theta$:

Proposition 9.11 (Parameter gradient)**.** *Let $u_t(x, \theta)$ be a control parameterized smoothly by $\theta \in \mathbb{R}^p$. Then*

$$\boxed{\nabla_\theta J(\theta) = \mathbb{E}\left[\int_0^1 \partial_\theta u_t(X_t, \theta)^\top \big(u_t(X_t, \theta) + \sigma_t^\top a_t\big) dt\right],} \tag{9.31}$$

where X_t is the trajectory generated under the current control $u_t(\cdot, \theta)$ and a_t is the pathwise adjoint (9.22) along that trajectory.

Direct vs. indirect contributions. Splitting (9.31):

$$\nabla_\theta J(\theta) = \tfrac{1}{2}\,\mathbb{E}\left[\int_0^1 \frac{\partial}{\partial\theta} \|u_t(X_t, \theta)\|^2 \, dt\right] + \mathbb{E}\left[\int_0^1 \partial_\theta u_t(X_t, \theta)^\top \sigma_t^\top a_t \, dt\right],$$

where the partial $\partial/\partial\theta$ in the first term holds X_t fixed. The first piece is the direct dependence on θ through the running cost $\frac{1}{2}\|u_t\|^2$; the second is the indirect dependence through the trajectory. The adjoint state a_t collapses the entire trajectory backpropagation into a single time-integrated coupling $\sigma_t^\top a_t$ at each t, so the cost of (9.31) is one forward simulation plus one backward ODE per gradient evaluation, independent of the parameter dimension p.

9.7.5 Adjoint Matching as Regression

From parameter gradient to regression loss. The functional form of (9.31) suggests training $u_t(\cdot, \theta)$ by minimizing the quadratic regression loss

$$\mathcal{L}_{\mathrm{AM}}(\theta) := \tfrac{1}{2}\,\mathbb{E}\left[\int_0^1 \big\| u_t(\bar{X}_t, \theta) + \sigma_t^\top \bar{a}_t \big\|^2 dt\right], \tag{9.32}$$

where the bars on $\bar{X}_t$ and $\bar{a}_t$ indicate that, although both are simulated using the current value of θ on each iteration, they are treated as *constants* in θ when differentiating: any implicit dependence through the trajectory and adjoint is dropped, and only the

explicit θ-dependence in $u_t(\bar{X}_t, \theta)$ contributes to $\nabla_\theta \mathcal{L}_{\mathrm{AM}}$. In an autodiff framework this is implemented by a `stop_gradient` (or `.detach()`) call applied to $\bar{X}_t$ and $\bar{a}_t$.

Differentiating (9.32) under this convention gives

$$\nabla_\theta \mathcal{L}_{\mathrm{AM}}(\theta) = \mathbb{E}\left[\int_0^1 \partial_\theta u_t(\bar{X}_t, \theta)^\top \big(u_t(\bar{X}_t, \theta) + \sigma_t^\top \bar{a}_t\big) dt\right] = \nabla_\theta J(\theta),$$

which is *exactly* (9.31): the barred quantities $\bar{X}_t$ and $\bar{a}_t$ have the same numerical values as the θ-dependent X_t and a_t (they are the same simulated trajectory and adjoint, just flagged as constants for autodiff), so the integrands agree pointwise. The reason stopping gradient through $\bar{X}_t$ and $\bar{a}_t$ is correct, rather than throwing away information, is that the chain-rule contribution from the implicit θ-dependence of the trajectory has *already been absorbed* into the $\sigma_t^\top a_t$ term by the adjoint identity (9.27). Differentiating through $\bar{X}_t$ again would double-count that contribution.

(9.32) therefore provides a *plain least-squares implementation* of the parameter gradient: at each step, simulate the forward and backward passes to obtain $(\bar{X}_t, \bar{a}_t)$, treat them as data, and update θ by stochastic gradient descent on the regression of $u_t(\bar{X}_t, \theta)$ onto the target $-\sigma_t^\top \bar{a}_t$.

Lean adjoint. Computing a_t via (9.22) requires gradients of the control through both (i) the drift Jacobian $\sigma_t \nabla_x u_t(\bar{X}_t, \theta)$ and (ii) the running cost $\nabla\big(\frac{1}{2}\|u_t(x,\theta)\|^2\big)|_{x=\bar{X}_t} = \nabla_x u_t(\bar{X}_t, \theta)^\top u_t(\bar{X}_t, \theta)$. Both involve the state Jacobian of $u_t(\cdot, \theta)$, a second-order quantity when $u_t(\cdot, \theta)$ is a neural network of x. The *lean adjoint* [28] $\tilde{a}_t$ drops all u-dependent contributions to the adjoint dynamics — both the $\sigma_t \nabla_x u_t(\bar{X}_t, \theta)$ term in the linearized drift and the $\nabla\big(\frac{1}{2}\|u_t(\cdot,\theta)\|^2\big)$ term in the running cost — while still running along the (frozen) controlled trajectory $\bar{X}_t$:

$$\frac{d\tilde{a}_t}{dt} = -\big[\nabla b_t(\bar{X}_t)^\top \tilde{a}_t + \nabla f_t(\bar{X}_t)\big], \qquad \tilde{a}_1 = \nabla g(\bar{X}_1). \tag{9.33}$$

The lean dynamics depend on the control only through the trajectory $\bar{X}_t$; the right-hand side itself contains no u. Replacing $\bar{a}_t$ with $\tilde{a}_t$ in (9.32) preserves u^* as a critical point [28]. The savings are substantial: each adjoint step requires only ∇b_t (a fixed term from the base SDE).

Week 6 Exercises

1. **(HJB via Lagrange multipliers.)** Derive the HJB equation (9.9) by treating the Fokker–Planck equation as a constraint.

   Reformulate the SOC problem (9.4) as

   $$\min_{u,\rho^u} \int_0^1 \int \tfrac{1}{2}\|u_t\|_2^2 \, \rho_t^u \, dx \, dt - \int R(x) \, \rho_1^u(x) \, dx$$

   subject to the FP equation $\partial_t \rho_t^u + \nabla \cdot (\rho_t^u (b_t + \sigma_t u_t)) = \frac{1}{2}\sigma_t^2 \Delta \rho_t^u$ with $\rho_0^u = \pi_0$. Introduce a Lagrange multiplier $\lambda_t(x)$ for this constraint and form the Lagrangian $\mathcal{L}[u, \rho^u, \lambda]$.

   By integration by parts in t and x, write $\mathcal{L}$ in the form

   $$\mathcal{L} = \int_0^1 \int \rho_t^u \, H_t(x, u_t, \lambda_t) \, dx \, dt + \text{boundary terms},$$

   where H_t is a pointwise function. Show that:

   (a) Stationarity in u_t gives $u_t^*(x) = \sigma_t \nabla \lambda_t(x)$.

   (b) Stationarity in ρ_t^u for $t \in (0, 1)$, after substituting u_t^*, gives a backward PDE for λ_t.

   (c) The boundary term in $t = 1$ gives the terminal condition $\lambda_1 = R$.

   Setting $V_t = -\lambda_t$, verify that the equations from (a)–(c) are exactly $u_t^* = -\sigma_t \nabla V_t$ and the HJB equation (9.9). This identifies the value function as the Lagrange multiplier of the FP constraint (up to sign).

2. **(Connection to entropy-regularized RL.)** Show that the SOC problem (9.4) is the continuous-time limit of an entropy-regularized RL problem.

   Background on RL. Reinforcement learning studies an agent interacting with an environment over discrete time steps $n = 0, 1, \ldots, N$. At step n the agent observes a *state* s_n, samples an *action* $a_n \sim \pi_n(\cdot \mid s_n)$ from a *policy* π_n, and the environment transitions to a new state s_{n+1} (in continuous-time control problems, the transition kernel is determined by the dynamics of the system and is not chosen by the agent). The agent receives a per-step reward $r(s_n, a_n)$ and a terminal reward $r_T(s_N)$. The goal is to choose $\pi = (\pi_0, \ldots, \pi_{N-1})$ to maximize $\mathbb{E}_\pi[\sum_n r(s_n, a_n) + r_T(s_N)]$.

   In *entropy-regularized* (or KL-regularized) RL, one fixes a *reference policy* $\pi_n^{\mathrm{ref}}(\cdot \mid s)$ and penalizes deviation from it via a KL term:

   $$\max_\pi \; \mathbb{E}_\pi \left[r_T(s_N) + \sum_{n=0}^{N-1} \Big(r(s_n, a_n) - D_{\mathrm{KL}}\Big(\pi_n(\cdot \mid s_n) \,\|\, \pi_n^{\mathrm{ref}}(\cdot \mid s_n) \Big) \Big) \right]. \tag{9.34}$$

This formulation is standard in robotics (where π^{ref} is a safe baseline) and in fine-tuning of language models, where π^{ref} is a pretrained model and r_T is a learned reward (the standard RLHF setup).

Discretization of the SOC problem. Discretize the controlled SDE (9.3) on a grid $0 = t_0 < \cdots < t_N = 1$ with step Δt. At each step, the controlled and uncontrolled transition kernels are Gaussian:

$$q^u(x' \mid x) = \mathcal{N}(x';\, x + (b_t + \sigma_t u_t)\Delta t,\, \sigma_t^2 \Delta t\, I_d), \quad q(x' \mid x) = \mathcal{N}(x';\, x + b_t \Delta t,\, \sigma_t^2 \Delta t\, I_d).$$

View q^u as a policy and q as a reference policy.

(a) Show that $D_{\text{KL}}(q^u(\cdot \mid x) \,\|\, q(\cdot \mid x)) = \frac{1}{2}\|u_t(x)\|_2^2\, \Delta t$.

(b) Conclude that the SOC running cost $\int_0^1 \frac{1}{2}\mathbb{E}\|u_t\|_2^2\, dt$ is the continuous-time limit of $\sum_n \mathbb{E}[D_{\text{KL}}(q^u(\cdot \mid X_{t_n}) \,\|\, q(\cdot \mid X_{t_n}))]$ — a sum of per-step KL penalties between the policy and reference, i.e. the entropy-regularization term in (9.34).

(c) Compare the SOC objective (9.4) (a *minimization*) with the RL objective (9.34) (a *maximization*). Multiply the SOC objective by -1 to convert it into a maximization:

$$\max_u \; \mathbb{E}\big[R(X_1^u)\big] - \int_0^1 \tfrac{1}{2}\mathbb{E}\|u_t\|_2^2\, dt.$$

Using (b), the running cost is the continuous-time limit of $\sum_n \mathbb{E}[D_{\text{KL}}(q^u \,\|\, q)]$. Match this term-by-term against (9.34) to obtain the dictionary

$$s_n \leftrightarrow X_{t_n}, \quad \pi_n \leftrightarrow q^u, \quad \pi_n^{\text{ref}} \leftrightarrow q, \quad r_T \leftrightarrow R, \quad r \leftrightarrow 0.$$

The per-step reward r vanishes because the SOC running cost contains only the KL term and no reward term inside the time integral; the entire reward signal is concentrated at the terminal time $t = 1$ via $R(X_1^u)$.

The conclusion is that fine-tuning a generative model to a tilted target $\pi_1 \exp(R)$ via SOC is structurally the same problem as fine-tuning a reference policy with KL-regularized RL (the standard RLHF setup), and methods developed for one translate directly to the other.

3. **(REINFORCE / score-function gradient.)** The pathwise adjoint of Section 9.7 requires backpropagating through the simulated SDE, which in particular needs ∇g and ∇f_t — the gradients of the terminal and running cost in x. In many applications (e.g. reward fine-tuning where g is a non-differentiable preference score, or a black-box simulator output) those gradients are unavailable. The *score-function* or *REINFORCE* estimator computes $\nabla_\theta J$ using only function evaluations of the cost, by moving the gradient onto the sampling density.

(a) REINFORCE in basic RL. For a single-step decision problem with action $a \sim \pi_\theta(\cdot)$ and reward $r(a)$, derive the score-function identity

$$\nabla_\theta \, \mathbb{E}_{a\sim\pi_\theta}[r(a)] = \mathbb{E}_{a\sim\pi_\theta}\big[r(a)\, \nabla_\theta \log \pi_\theta(a)\big], \tag{9.35}$$

by writing the expectation as an integral and differentiating under the integral sign. Observe that r enters only through evaluations, not through its gradient. For a multi-step MDP with trajectory $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$, joint density $p_\theta(\tau) = p(s_0) \prod_n \pi_\theta(a_n \mid s_n)\, p(s_{n+1} \mid s_n, a_n)$, and total reward $R(\tau) = \sum_n r(s_n, a_n)$, show that

$$\nabla_\theta \, \mathbb{E}_{\tau\sim p_\theta}[R(\tau)] = \mathbb{E}_{\tau\sim p_\theta}\Big[R(\tau) \sum_n \nabla_\theta \log \pi_\theta(a_n \mid s_n)\Big],$$

where the transition density $p(s_{n+1} \mid s_n, a_n)$ contributes nothing because it does not depend on θ.

(b) Discrete-time SOC via the score trick. Apply REINFORCE to the discretized SOC problem on the grid $0 = t_0 < \cdots < t_N = 1$ with the Euler–Maruyama scheme (9.17). Show that the joint trajectory density is

$$p_\theta(X_{0:N}) = \pi_0(X_0) \prod_{k=0}^{N-1} \mathcal{N}\big(X_{k+1};\, X_k + (b_k(X_k) + \sigma_k u_{t_k}(X_k, \theta))\Delta t,\, \sigma_k^2 \Delta t\, I\big),$$

and compute that the per-step score is

$$\nabla_\theta \log p_\theta(X_{k+1} \,|\, X_k) = \partial_\theta u_{t_k}(X_k, \theta)^\top \, \frac{X_{k+1} - X_k - (b_k(X_k) + \sigma_k u_{t_k}(X_k, \theta))\Delta t}{\sigma_k} \tag{9.36}$$

which can be written as $\partial_\theta u_{t_k}(X_k, \theta)^\top \sqrt{\Delta t}\, \xi_k$.

Hence, separating the direct dependence of the cost on θ from the indirect dependence through the law of $X_{0:N}$,

$$\nabla_\theta J(\theta) = \mathbb{E}\Bigg[\sum_{k=0}^{N-1} \partial_\theta u_{t_k}(X_k, \theta)^\top u_{t_k}(X_k, \theta)\, \Delta t + \Phi(X_{0:N}) \sum_{k=0}^{N-1} \partial_\theta u_{t_k}(X_k, \theta)^\top \sqrt{\Delta t}\, \xi_k\Bigg], \tag{9.37}$$

where $\Phi(X_{0:N}) := \sum_k \big(\frac{1}{2}\|u_k\|^2 + f_k\big)\Delta t + g(X_N)$ is the path cost. Note that Φ enters only through evaluations, never through gradients.

(c) Continuous limit. Send $\Delta t \to 0$ in (9.37). Show that the score becomes a stochastic integral,

$$\sum_{k=0}^{N-1} \partial_\theta u_{t_k}(X_k, \theta)^\top \sqrt{\Delta t}\, \xi_k \;\longrightarrow\; \int_0^1 \partial_\theta u_t(X_t, \theta)^\top \, dW_t,$$

yielding the *continuous-time REINFORCE estimator*

$$\boxed{\nabla_\theta J(\theta) = \mathbb{E}\left[\int_0^1 \partial_\theta u_t(X_t, \theta)^\top u_t(X_t, \theta)\, dt + \Phi(X) \int_0^1 \partial_\theta u_t(X_t, \theta)^\top \, dW_t\right],} \tag{9.38}$$

where $\Phi(X) = \int_0^1 \ell_t(X_t)\,dt + g(X_1)$. Compare with the pathwise adjoint formula (9.31): the direct term agrees, while the $\sigma_t^\top a_t$ piece, which captures the trajectory-wise sensitivity through backward simulation, is here replaced by the return $\Phi(X)$ multiplied by the (forward-only) score $\int \partial_\theta u_t(X_t,\theta)^\top dW_t$. In particular, computing (9.38) requires no backward pass and no gradient of g or f_t; only their values along the trajectory are needed.

(d) Girsanov derivation. Re-derive (9.38) directly in continuous time using Girsanov's theorem. Let $\mathbb{P}$ be the law of the uncontrolled base process $dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t^{\mathbb{P}}$ and let $\mathbb{P}^\theta$ be the law of the controlled process. The score for the parametric family $\{\mathbb{P}^\theta\}$ relative to the fixed reference $\mathbb{P}$ is

$$\nabla_\theta \log p_\theta(X) = \nabla_\theta \log \frac{d\mathbb{P}^\theta}{d\mathbb{P}}(X),$$

which is a path functional, well-defined for any path X.

We compute this score for a *generic* path X, then specialize to $X = X^\theta$. From (9.10), the log Radon–Nikodym density expressed as a path functional is

$$\log \frac{d\mathbb{P}^\theta}{d\mathbb{P}}(X) = \int_0^1 u_t(X_t,\theta)^\top \sigma_t^{-1}(dX_t - b_t(X_t)\,dt) - \tfrac{1}{2}\int_0^1 \|u_t(X_t,\theta)\|^2\,dt,$$

where $\sigma_t^{-1}(dX_t - b_t\,dt)$ denotes the path-data expression for the driving noise (and reduces to $dW_t^{\mathbb{P}}$ under $\mathbb{P}$ or to $u_t(X_t,\theta)\,dt + dW_t^{\mathbb{P}^\theta}$ under $\mathbb{P}^\theta$). Holding the path X fixed and differentiating in θ:

$$\nabla_\theta \log \frac{d\mathbb{P}^\theta}{d\mathbb{P}}(X) = \int_0^1 \partial_\theta u_t(X_t,\theta)^\top \sigma_t^{-1}(dX_t - b_t\,dt) - \int_0^1 \partial_\theta u_t(X_t,\theta)^\top u_t(X_t,\theta)\,dt.$$

Now specialize to the controlled path: $X_t = X_t^\theta$ satisfies $dX_t^\theta - b_t\,dt = \sigma_t u_t(X_t^\theta,\theta)\,dt + \sigma_t\,dW_t^{\mathbb{P}^\theta}$, so $\sigma_t^{-1}(dX_t^\theta - b_t\,dt) = u_t(X_t^\theta,\theta)\,dt + dW_t^{\mathbb{P}^\theta}$. Substituting, the cross-terms $\int \partial_\theta u_t(X_t^\theta,\theta)^\top u_t(X_t^\theta,\theta)\,dt$ cancel and

$$\nabla_\theta \log \frac{d\mathbb{P}^\theta}{d\mathbb{P}}\bigg|_{X^\theta} = \int_0^1 \partial_\theta u_t(X_t^\theta,\theta)^\top\,dW_t^{\mathbb{P}^\theta}. \tag{9.39}$$

This is a martingale, hence has zero mean under $\mathbb{P}^\theta$, as a score should.

The standard score-function identity then yields, for any path functional Φ_θ depending on θ both directly (through the integrand) and indirectly (through the law of X),

$$\nabla_\theta \mathbb{E}^{\mathbb{P}^\theta}[\Phi_\theta] = \mathbb{E}^{\mathbb{P}^\theta}\big[\nabla_\theta \Phi_\theta\big] + \mathbb{E}^{\mathbb{P}^\theta}\big[\Phi_\theta \cdot \nabla_\theta \log p_\theta(X^\theta)\big].$$

Applying this with $\Phi_\theta = \int_0^1 \ell_t(X_t)\,dt + g(X_1)$, using $\nabla_\theta \Phi_\theta = \int_0^1 \partial_\theta u_t(X_t,\theta)^\top u_t(X_t,\theta)\,dt$ (only $\frac{1}{2}\|u_t(X_t,\theta)\|^2$ contributes a direct gradient), and substituting (9.39) recovers (9.38).

(e) Variance. Argue heuristically why the REINFORCE estimator (9.38) typically has higher variance than the pathwise adjoint estimator (9.31): in REINFORCE the *entire* return $\Phi(X)$ multiplies a stochastic integral, whereas in the pathwise adjoint the local sensitivity $\sigma_t^\top a_t$ isolates the contribution of each time slice. The trade-off between score-function and pathwise gradient estimators is the central design choice in policy-gradient reinforcement learning.

10 Sampling as Optimization in Probability Space

The SOC formulation of Section 9 cast generative sampling as an optimization problem on a space of *path measures*: minimize a KL divergence subject to dynamical constraints. The same idea applies more broadly. Most modern sampling algorithms can be understood as instances of optimization on the space $\mathcal{P}(\mathbb{R}^d)$ of probability measures, with the target π playing the role of the minimizer of an energy functional. The following dictionary will guide the narrative in this section:

Optimization on $\mathbb{R}^d$	**Sampling on $\mathcal{P}(\mathbb{R}^d)$**
gradient descent	Langevin dynamics
heavy-ball / Nesterov momentum	underdamped Langevin
preconditioned methods (Newton, natural gradient)	preconditioned samplers (Riemannian Langevin/HMC, ensemble)

This section walks through each row, with concrete examples to build intuition. We do not aim for full proofs — many of the rate results require nontrivial technical work — but we will state the key formulas, work through a few revealing special cases (mostly involving Gaussians, where everything is computable), and explain why each algorithm earns its place in the table.

10.1 Langevin as KL Minimization

We begin on the optimization side, recall the gradient-flow viewpoint together with its proximal and preconditioned variants, and then transport the entire picture to probability space via the Wasserstein metric. The end result is a clean identification of Langevin dynamics as Wasserstein gradient flow of KL divergence, with the log-Sobolev inequality playing the role of the Polyak–Łojasiewicz condition.

10.1.1 Optimization on $\mathbb{R}^d$ and gradient flows

Let $V : \mathbb{R}^d \to \mathbb{R}$ be smooth and (for concreteness) α-strongly convex, with unique minimizer x^*. The simplest discrete iteration,

$$x_{k+1} = x_k - h\,\nabla V(x_k), \tag{10.1}$$

is the *forward-Euler* discretization of the gradient-flow ODE

$$\dot{x}_t = -\nabla V(x_t). \tag{10.2}$$

Under α-strong convexity, $\frac{d}{dt}\|x_t - x^*\|^2 = 2(x_t - x^*)\cdot(-\nabla V(x_t)) \le -2\alpha\|x_t - x^*\|^2$, so by Grönwall $\|x_t - x^*\|^2 \le e^{-2\alpha t}\|x_0 - x^*\|^2$.

A second route to the same flow is the *proximal* or backward-Euler scheme

$$x_{k+1} \;=\; \arg\min_{y\in\mathbb{R}^d}\Big[V(y) + \tfrac{1}{2h}\,\|y - x_k\|^2\Big]. \tag{10.3}$$

The first-order condition gives the implicit update $x_{k+1} = x_k - h\,\nabla V(x_{k+1})$, which is backward Euler for (10.2). Both (10.1) and (10.3) limit to the gradient flow (10.2) as $h \to 0$. The difference matters when $h > 0$: forward Euler is explicit (cheap per step) but requires $h \lesssim 1/L$ for stability when $\nabla^2 V \preceq LI$, while the proximal scheme is unconditionally stable in h at the cost of solving an optimization problem each step.

Preconditioning via a non-Euclidean metric. The Euclidean norm in (10.3) is a choice. Replacing it by $\|y - x_k\|_M^2 := (y - x_k)^\top M (y - x_k)$ for some $M \succ 0$ produces the *preconditioned* proximal scheme

$$x_{k+1} \;=\; \arg\min_y \Big[V(y) + \tfrac{1}{2h} \|y - x_k\|_M^2 \Big], \tag{10.4}$$

with first-order condition $\nabla V(x_{k+1}) + \frac{1}{h} M(x_{k+1} - x_k) = 0$, i.e. $x_{k+1} = x_k - h\, M^{-1} \nabla V(x_{k+1})$. The continuous-time limit is the preconditioned flow

$$\dot{x}_t = -M^{-1} \nabla V(x_t). \tag{10.5}$$

Choosing $M = \nabla^2 V(x_k)$ locally recovers Newton's method; matching M to the global curvature of V removes anisotropy. The point worth recording is that the *metric in the proximal step determines the gradient flow*: a different metric gives a different flow, with the same equilibrium x^* but different trajectories and convergence constants.

10.1.2 Transport to probability space

We now transport this picture to sampling. The state is no longer a point $x \in \mathbb{R}^d$ but a probability measure $\mu \in \mathcal{P}(\mathbb{R}^d)$; the role of the objective V is played by the KL divergence to the target,

$$\mathcal{F}(\mu) := D_{\mathrm{KL}}(\mu \,\|\, \pi)\,, \qquad \arg\min_\mu \mathcal{F}(\mu) = \pi, \tag{10.6}$$

attained uniquely at π where $\mathcal{F}(\pi) = 0$. The minimum encodes all information about the target.

For the proximal scheme to make sense on $\mathcal{P}(\mathbb{R}^d)$ we need a metric. The natural choice is the Wasserstein-2 distance W_2 (Theorem 4.3). With this choice, (10.3) becomes the *JKO scheme* [41]:

$$\mu_{k+1} = \arg\min_\mu \Big[D_{\mathrm{KL}}(\mu \,\|\, \pi) + \tfrac{1}{2h} W_2^2(\mu, \mu_k) \Big]. \tag{10.7}$$

This is the probability-space analogue of (10.3), with $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$ in place of V and $W_2^2/2h$ in place of $\|\cdot - x_k\|^2/2h$.

Solving JKO infinitesimally. The aim is to identify a continuous-time flow as the $h \to 0$ limit of (10.7). We approximate both terms in (10.7) to first order in h around μ_k.

The minimizer μ_{k+1} in (10.7) is close to μ_k for small h, so we parametrize it as the pushforward of μ_k along an infinitesimal displacement field hv for some velocity $v : \mathbb{R}^d \to \mathbb{R}^d$:

$$\mu_{k+1} = (I + hv)_{\#}\mu_k. \tag{10.8}$$

The pair (μ_k, v) then sweeps out a curve $\mu_t^{(\mathrm{lin})} = (I + tv)_{\#}\mu_k$ for $t \in [0, h]$ joining μ_k to μ_{k+1}, satisfying the continuity equation $\partial_t \mu_t^{(\mathrm{lin})} + \nabla\cdot(\mu_t^{(\mathrm{lin})} v) = 0$ at $t = 0$.

Approximating W_2^2. The cost of moving every particle of μ_k along the displacement hv is $\mathbb{E}_{\mu_k}[\|hv(X)\|^2] = h^2 \int \|v\|^2 \, d\mu_k$. Since W_2^2 is the minimum transport cost between μ_k and μ_{k+1}, and the displacement (10.8) provides one admissible transport,

$$W_2^2(\mu_{k+1}, \mu_k) \;\le\; h^2 \int \|v(x)\|^2 \, d\mu_k(x); \tag{10.9}$$

infinitesimally this inequality is an equality (the displacement is optimal to leading order), and we will treat it as such. The right-hand side is a quadratic form in v, the squared $L^2(\mu_k)$ *norm*; intuitively the kinetic energy needed to displace μ_k to μ_{k+1}.

Approximating $D_{\mathrm{KL}}(\mu \,\|\, \pi)$. Differentiating $\mu_t^{(\mathrm{lin})}$ at $t = 0$:

$$D_{\mathrm{KL}}(\mu_{k+1} \,\|\, \pi) \approx D_{\mathrm{KL}}(\mu_k \,\|\, \pi) + h \int \big(\log(\mu_k/\pi) + 1\big)\, \partial_t \mu_t^{(\mathrm{lin})}\big|_{t=0} \, dx,$$

using $\delta D_{\mathrm{KL}}(\mu \,\|\, \pi) = \int (\log(\mu/\pi) + 1)\, \delta\mu \, dx$. The perturbation is $\partial_t \mu_t^{(\mathrm{lin})}|_{t=0} = -\nabla\cdot(\mu_k v)$. Integration by parts (the constant $+1$ drops since $\int \nabla\cdot(\mu_k v)\, dx = 0$):

$$D_{\mathrm{KL}}(\mu_{k+1} \,\|\, \pi) \approx D_{\mathrm{KL}}(\mu_k \,\|\, \pi) + h \int \nabla \log(\mu_k/\pi) \cdot v \, d\mu_k. \tag{10.10}$$

This is linear in v and identifies $\nabla \log(\mu_k/\pi)$ as the "gradient" of $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$ at μ_k in the $L^2(\mu_k)$-pairing.

The quadratic problem. Substituting (10.9) and (10.10) into the JKO objective:

$$D_{\mathrm{KL}}(\mu_{k+1} \,\|\, \pi) + \tfrac{1}{2h} W_2^2(\mu_{k+1}, \mu_k) \approx D_{\mathrm{KL}}(\mu_k \,\|\, \pi) + h \int \nabla \log(\mu_k/\pi) \cdot v \, d\mu_k + \tfrac{h}{2} \int \|v\|^2 \, d\mu_k,$$

a strictly convex quadratic in v over $L^2(\mu_k)$. Pointwise stationarity in $v(x)$ gives

$$v(x) = -\nabla \log(\mu_k/\pi)(x). \tag{10.11}$$

From the proximal step to Fokker–Planck. Substituting (10.11) into the continuity equation and letting $h \to 0$ ($\mu_k \to \mu_t$, $\mu_{k+1} \to \mu_{t+dt}$):

$$\partial_t \mu_t + \nabla\cdot(\mu_t v_t) = 0, \qquad v_t = -\nabla \log(\mu_t/\pi).$$

Writing $\pi \propto e^{-V}$ so $\nabla \log \pi = -\nabla V$, expanding $\nabla \log \mu_t = (\nabla \mu_t)/\mu_t$, and simplifying:

$$\partial_t \mu_t = \nabla\cdot(\mu_t \nabla V) + \Delta \mu_t, \tag{10.12}$$

which is the Fokker–Planck equation for the *Langevin SDE*

$$dX_t = -\nabla V(X_t)\, dt + \sqrt{2}\, dW_t. \tag{10.13}$$

Proposition 10.1 (Langevin = Wasserstein gradient flow). *For $\pi \propto e^{-V}$, the law μ_t of the Langevin SDE (10.13) is the small-step limit of the JKO scheme (10.7): the velocity field driving the continuity equation is the $L^2(\mu_t)$-steepest-descent direction for $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$, namely $v_t = -\nabla \log(\mu_t/\pi)$.*

The construction parallels the Euclidean case term-by-term: proximal step (10.3) $\leftrightarrow$ JKO (10.7); quadratic penalty $\|y-x_k\|^2 \leftrightarrow$ infinitesimal transport cost $\int \|v\|^2 d\mu_k$ (10.9); linearization of V around $x_k \leftrightarrow$ linearization of $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$ around μ_k (10.10); minimizer $-\nabla V(x_k) \leftrightarrow$ minimizer $-\nabla \log(\mu_k/\pi)$ (10.11); flow $\dot{x} = -\nabla V \leftrightarrow$ Fokker–Planck.

Identifying the metric tensor. The parallel goes further: just as the Euclidean preconditioned flow (10.5) reads $\dot{x} = -M^{-1}\nabla V(x)$, the Wasserstein flow has the same structural form once we identify the metric tensor explicitly. Velocity fields and density perturbations are related by the continuity equation: a perturbation $\delta\mu$ generated by a potential-field velocity $v = \nabla\phi$ satisfies $\delta\mu = -\nabla\cdot(\mu\nabla\phi)$. Define the elliptic operator

$$\mathcal{L}_\mu \;:=\; -\nabla\cdot(\mu\,\nabla\,\cdot), \tag{10.14}$$

acting on smooth scalar potentials. The map $\phi \mapsto \delta\mu = \mathcal{L}_\mu\phi$ is the bijection between potentials and tangent vectors at μ, and the W_2 inner product on velocities can be written as

$$\int \nabla\phi_1 \cdot \nabla\phi_2 \, d\mu \;=\; \int \phi_1 \, \mathcal{L}_\mu\phi_2 \, dx \;=\; \langle \phi_1, \mathcal{L}_\mu\phi_2 \rangle_{L^2},$$

showing that $\mathcal{L}_\mu$ *is* the W_2 metric tensor in the potential representation. The Fokker–Planck equation can then be written as

$$\partial_t \mu_t \;=\; -\mathcal{L}_{\mu_t}\big[\log(\mu_t/\pi)\big], \tag{10.15}$$

where $\log(\mu_t/\pi)$ is the $L^2(dx)$ first variation of $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$ at μ_t. Compare with (10.5): $\dot{x} = -M^{-1}\nabla V$. The elliptic operator $\mathcal{L}_\mu$ plays the role of M^{-1} — the inverse metric — while $\log(\mu/\pi)$ plays the role of ∇V, the bare first variation. Both flows are "preconditioned" steepest descents on their objective, with the preconditioner determined by the metric: constant M on $\mathbb{R}^d$; the position-dependent $\mathcal{L}_\mu = -\nabla\cdot(\mu\,\nabla\,\cdot)$ on $\mathcal{P}(\mathbb{R}^d)$. The formal perspective is brought by Otto [54]. Discussions of other metric tensors one may refer to [19].

10.1.3 Convergence rate: log-Sobolev as Polyak–Łojasiewicz

In Euclidean optimization with α-strongly convex V, the Polyak–Łojasiewicz (PL) inequality

$$V(x) - V(x^*) \;\le\; \tfrac{1}{2\alpha}\,\|\nabla V(x)\|^2 \tag{10.16}$$

holds and, combined with the dissipation identity $\frac{d}{dt}V(x_t) = -\|\nabla V(x_t)\|^2$ along (10.2), yields by Grönwall

$$V(x_t) - V(x^*) \;\le\; e^{-2\alpha t}\big(V(x_0) - V(x^*)\big).$$

The sampling analogue is the *log-Sobolev inequality*: π satisfies LSI with constant $\alpha > 0$ if for all μ,

$$D_{\mathrm{KL}}(\mu \,\|\, \pi) \;\le\; \tfrac{1}{2\alpha}\, \mathbb{E}_\mu\big[\|\nabla \log(\mu/\pi)\|^2\big]\,. \tag{10.17}$$

The right-hand side is $\frac{1}{2\alpha}\int \|v\|^2 d\mu$ with $v = \nabla \log(\mu/\pi)$ the JKO-velocity field (10.11), so LSI is exactly the PL inequality "$\mathcal{F}(\mu) \le \frac{1}{2\alpha}\|v\|^2_{L^2(\mu)}$" in the W_2 geometry. Combined with the dissipation identity along the WGF, $\frac{d}{dt} D_{\mathrm{KL}}(\mu_t \,\|\, \pi) = -\mathbb{E}_{\mu_t}[\|\nabla \log(\mu_t/\pi)\|^2] = -\|v_t\|^2_{L^2(\mu_t)}$, LSI yields exponential decay:

$$D_{\mathrm{KL}}(\mu_t \,\|\, \pi) \;\le\; e^{-2\alpha t}\, D_{\mathrm{KL}}(\mu_0 \,\|\, \pi)\,. \tag{10.18}$$

When does π satisfy LSI? The Bakry–Émery criterion gives LSI with constant α whenever V is α-strongly convex — so e.g. Gaussians and strongly log-concave posteriors. Beyond strong convexity, LSI is preserved under bounded perturbations. Thus, mixtures and other mild non-log-concave perturbations still admit LSI, though with a constant that degrades with the perturbation size. We mention that for a different, Fisher-Rao metric, the PL inequality holds under much milder conditions for the distributions; see [13].

Worked example: Gaussian target. Take $\pi = \mathcal{N}(0, I)$ on $\mathbb{R}^d$, so $V(x) = \frac{1}{2}\|x\|^2$ is 1-strongly convex; this means $\alpha = 1$. Initialize at $\mu_0 = \mathcal{N}(m_0, \Sigma_0)$. The Langevin SDE preserves Gaussianity: $\mu_t = \mathcal{N}(m_t, \Sigma_t)$ with the moment ODEs

$$\dot{m}_t = -m_t, \qquad \dot{\Sigma}_t = -2\Sigma_t + 2I, \tag{10.19}$$

obtained directly from Itô's formula. Solving:

$$m_t = e^{-t} m_0, \qquad \Sigma_t = e^{-2t}\Sigma_0 + (1 - e^{-2t})I. \tag{10.20}$$

Mean decays at rate 1; covariance approaches I at rate 2. The KL between two Gaussians is $D_{\mathrm{KL}}(\mathcal{N}(m,\Sigma)\,\|\,\mathcal{N}(0,I)) = \frac{1}{2}\big(\mathrm{tr}\Sigma - d - \log\det\Sigma + \|m\|^2\big)$; substituting (10.20) and expanding verifies $D_{\mathrm{KL}}(\mu_t \,\|\, \pi) = O(e^{-2t})$, matching the LSI rate (10.18) with $\alpha = 1$.

For an anisotropic target $\pi = \mathcal{N}(0, \Lambda^{-1})$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, the dynamics decouple into d independent 1D systems. The i-th coordinate satisfies $\dot{m}_i = -\lambda_i m_i$, $\dot{\Sigma}_{ii} = -2\lambda_i \Sigma_{ii} + 2$, contracting at rate λ_i. The global mixing rate in W_2 is set by the slowest direction $\lambda_{\min}$: equilibration takes continuous time $\asymp 1/\lambda_{\min}$, and we will see in the next subsection that this becomes the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ in discrete iteration counts.

10.2 Acceleration in Optimization

Once Langevin is recognized as a gradient flow, it is natural to ask whether sampling admits an analogue of *accelerated* optimization. Momentum methods (Polyak heavy ball, Nesterov) improve the mixing time on κ-conditioned strongly convex problems from κ to $\sqrt{\kappa}$. The sampling analogue — underdamped Langevin — is expected to achieve the same square-root improvement, and on Gaussian targets the calculation is short, closed-form, and worth working out in full.

10.2.1 Momentum in continuous-time optimization

The Polyak heavy-ball iteration $x_{k+1} = x_k - h\nabla V(x_k) + \mu(x_k - x_{k-1})$ has continuous-time limit (under the scaling $\gamma = (1-\mu)/\sqrt{h}$)

$$\ddot{x}_t + \gamma\,\dot{x}_t + \nabla V(x_t) = 0, \tag{10.21}$$

a damped Newtonian equation with friction γ in potential V. We use heavy ball because its ODE has constant coefficients and admits a clean closed-form spectral analysis; the closely related Nesterov acceleration is discussed briefly later.

For the 1D quadratic $V(x) = \frac{\lambda}{2}x^2$, (10.21) is the linear oscillator

$$\ddot{x}_t + \gamma\,\dot{x}_t + \lambda x_t = 0,$$

with characteristic equation $\nu^2 + \gamma\nu + \lambda = 0$ and roots

$$\nu_\pm \;=\; \tfrac{1}{2}\big(-\gamma \pm \sqrt{\gamma^2 - 4\lambda}\big). \tag{10.22}$$

The asymptotic decay rate of $|x_t|$ is $|\Re\,\nu_+|$, optimized over $\gamma > 0$ at $\gamma_* = 2\sqrt{\lambda}$, where both roots coincide at $-\sqrt{\lambda}$. Compared to gradient flow $\dot{x} = -\lambda x$ with rate λ, heavy ball improves the rate to $\sqrt{\lambda}$.

Anisotropic quadratic: continuous-time rates. For $V(x) = \frac{1}{2}x^\top \Lambda x$ on $\mathbb{R}^d$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, the eigendirections decouple. The slowest mode controls equilibration, giving continuous-time convergent times:

$$T_{\text{gradient}} \;=\; 1/\lambda_{\min}, \qquad T_{\text{momentum}} \;=\; 1/\sqrt{\lambda_{\min}} \quad (\text{at } \gamma_* = 2\sqrt{\lambda_{\min}}). \tag{10.23}$$

Beyond quadratics: a Lyapunov argument. The spectral analysis is special to quadratics, but the $\sqrt{\alpha}$ rate persists for any α-strongly convex V (recovering $\alpha = \lambda$ in the 1D quadratic and $\alpha = \lambda_{\min}$ in the anisotropic case). Writing $\xi_t := x_t - x^*$ and choosing $\gamma = 2\sqrt{\alpha}$, define the augmented energy

$$\mathcal{E}(t) \;:=\; \big(V(x_t) - V(x^*)\big) \;+\; \tfrac{1}{2}\,\big|\dot{x}_t + \sqrt{\alpha}\,\xi_t\big|^2. \tag{10.24}$$

The cross-shift $\sqrt{\alpha}\,\xi$ inside the kinetic term is what closes the dissipation budget. From (10.21), $\ddot{x} + \sqrt{\alpha}\dot{x} = -\sqrt{\alpha}\dot{x} - \nabla V$, so

$$\dot{\mathcal{E}} \;=\; \langle \nabla V, \dot{x}\rangle + \big\langle \dot{x} + \sqrt{\alpha}\,\xi,\, -\sqrt{\alpha}\dot{x} - \nabla V\big\rangle \;=\; -\sqrt{\alpha}\,|\dot{x}|^2 \;-\; \alpha\,\langle \xi, \dot{x}\rangle \;-\; \sqrt{\alpha}\,\langle \xi, \nabla V\rangle.$$

Adding $\sqrt{\alpha}\,\mathcal{E}$, the $\alpha\langle\xi, \dot{x}\rangle$ cross terms cancel and

$$\dot{\mathcal{E}} + \sqrt{\alpha}\,\mathcal{E} \;=\; \sqrt{\alpha}\Big[\big(V - V^*\big) - \langle \nabla V, \xi\rangle + \tfrac{\alpha}{2}|\xi|^2\Big] \;-\; \tfrac{\sqrt{\alpha}}{2}\,|\dot{x}|^2 \;\le\; 0,$$

where the bracket is ≤ 0 by α-strong convexity at (x_t, x^*). Hence $\mathcal{E}(t) \le \mathcal{E}(0)\,e^{-\sqrt{\alpha}\,t}$. With initial momentum $\dot{x}_0 = 0$, strong convexity also gives $\mathcal{E}(0) \le 2\big(V(x_0) - V(x^*)\big)$, so

$$V(x_t) - V(x^*) \;\le\; 2\big(V(x_0) - V(x^*)\big)\,e^{-\sqrt{\alpha}\,t}. \tag{10.25}$$

From continuous time to iterations. The iteration count of each method, suitably discretized, is (continuous-time convergence time)/(step size). For gradient descent the stable forward-Euler step is $h \lesssim 1/\lambda_{\max}$. For heavy ball at the optimal damping $\gamma_* = 2\sqrt{\lambda_{\min}}$, every mode is under- or critically damped: the Jacobian eigenvalues at x^* are $\nu_\pm^{(i)} = -\sqrt{\lambda_{\min}} \pm i\sqrt{\lambda_i - \lambda_{\min}}$ of magnitude $\sqrt{\lambda_i}$, with stiffest oscillation frequency $\sqrt{\lambda_{\max}}$. A leapfrog-style symplectic-with-friction discretization is linearly stable for $h \lesssim 1/\sqrt{\lambda_{\max}}$ (the oscillator CFL of the leapfrog substep [37]; the friction half-step is the exact OU map and unconditionally stable), and preserves the continuous-time $\sqrt{\alpha}$ rate. The resulting iteration counts are

$$K_{\text{gradient}} \asymp \kappa, \qquad K_{\text{momentum}} \asymp \sqrt{\kappa}. \tag{10.26}$$

The condition number $\kappa = \lambda_{\max}/\lambda_{\min}$.

Beyond quadratics, the heavy-ball discretization is more delicate: forward Euler on (10.21) with constant momentum — the Polyak iteration — can fail to converge on certain strongly convex but non-quadratic problems [46]. Nesterov's extrapolated-gradient update $y_k = x_k + \beta(x_k - x_{k-1})$, $x_{k+1} = y_k - h\nabla V(y_k)$ repairs this by evaluating the gradient at the look-ahead point; from the symplectic perspective of [72, 64, 63], this update is a structure-preserving discretization of the *high-resolution* ODE

$$\ddot{x} + 2\sqrt{\alpha}\,\dot{x} + \sqrt{h}\,\nabla^2 V(x)\,\dot{x} + \nabla V(x) = 0,$$

which augments (10.21) by a Hessian-modulated friction of order $\sqrt{h}$. The extra friction term — invisible at the level of (10.21), is what restores the $\sqrt{\kappa}$ rate beyond quadratics.

10.3 Acceleration in Sampling: Underdamped Langevin

The sampling counterpart introduces a momentum variable $P_t \in \mathbb{R}^d$ and runs the joint (X, P) SDE

$$\begin{aligned} dX_t &= P_t\,dt, \\ dP_t &= -\nabla V(X_t)\,dt - \gamma P_t\,dt + \sqrt{2\gamma}\,dW_t. \end{aligned} \tag{10.27}$$

The invariant measure on (X, P)-space is $\pi(x) \otimes \mathcal{N}(0, I)(p)$; the marginal in X is the target π. The drift in X is the momentum (no friction); the drift in P has restoring force $-\nabla V$, friction $-\gamma P$, and noise $\sqrt{2\gamma}\,dW_t$ in the precise proportion that keeps the joint Gibbs distribution invariant; see exercise at the end of this section.

Explicit calculation: 1D Gaussian. Take $\pi = \mathcal{N}(0, 1/\lambda)$, $V(x) = \frac{\lambda}{2}x^2$ on $\mathbb{R}$. The system (10.27) is linear in (X, P), so all moments satisfy closed ODEs. Write

$$m_t = \mathbb{E}[X_t], \quad n_t = \mathbb{E}[P_t], \quad \Sigma_t = \mathrm{Var}(X_t), \quad \Pi_t = \mathrm{Var}(P_t), \quad C_t = \mathrm{Cov}(X_t, P_t).$$

Means. Taking expectations in (10.27):

$$\dot{m}_t = n_t, \qquad \dot{n}_t = -\lambda\, m_t - \gamma\, n_t.$$

Eliminating n_t via $n_t = \dot{m}_t$ gives

$$\ddot{m}_t + \gamma\,\dot{m}_t + \lambda\, m_t = 0, \tag{10.28}$$

which is the heavy-ball ODE (10.21). The mean decays at rate $|\Re\,\nu_+|$ from (10.22), optimized at $\sqrt{\lambda}$ when $\gamma_* = 2\sqrt{\lambda}$.

Second moments. Using Itô's formula for X_t^2, P_t^2, and X_tP_t in (10.27):

$$\dot{\Sigma}_t = 2C_t, \quad \dot{\Pi}_t = -2\lambda\, C_t - 2\gamma\,\Pi_t + 2\gamma, \quad \dot{C}_t = \Pi_t - \lambda\,\Sigma_t - \gamma\, C_t.$$

The stationary point is $(\Sigma_\infty, \Pi_\infty, C_\infty) = (1/\lambda, 1, 0)$, recovering $\pi \otimes \mathcal{N}(0,1)$. Writing $\Sigma_t' = \Sigma_t - 1/\lambda$, $\Pi_t' = \Pi_t - 1$ and C_t as deviations from stationarity, the perturbations satisfy

$$\frac{d}{dt}\begin{pmatrix}\Sigma_t' \\ \Pi_t' \\ C_t\end{pmatrix} = \underbrace{\begin{pmatrix}0 & 0 & 2 \\ 0 & -2\gamma & -2\lambda \\ -\lambda & 1 & -\gamma\end{pmatrix}}_{A}\begin{pmatrix}\Sigma_t' \\ \Pi_t' \\ C_t\end{pmatrix}.$$

The characteristic polynomial $\det(A - \nu I) = 0$ expands to

$$\nu^3 + 3\gamma\,\nu^2 + (2\gamma^2 + 4\lambda)\,\nu + 4\gamma\lambda = 0.$$

Substituting $\nu = -\gamma$ gives 0, so the polynomial factors as

$$(\nu + \gamma)\big(\nu^2 + 2\gamma\,\nu + 4\lambda\big) = 0, \tag{10.29}$$

with roots $\nu = -\gamma$ and $\nu = -\gamma \pm \sqrt{\gamma^2 - 4\lambda}$. Real parts of the three roots:

- Underdamped $\gamma < 2\sqrt{\lambda}$: $-\gamma$ and $-\gamma \pm i\sqrt{4\lambda - \gamma^2}$, all with $\Re\,\nu = -\gamma$.
- Overdamped $\gamma > 2\sqrt{\lambda}$: the slowest is $-\gamma + \sqrt{\gamma^2 - 4\lambda}$ with magnitude $\gamma - \sqrt{\gamma^2 - 4\lambda} \sim 2\lambda/\gamma$ for large γ.
- Critical $\gamma = 2\sqrt{\lambda}$: (10.29) reduces to $(\nu + 2\sqrt{\lambda})^3 = 0$, a triple root at $-2\sqrt{\lambda}$.

W_2 rate. For 1D Gaussians, $W_2^2(\mathcal{N}(m,\Sigma), \mathcal{N}(0, 1/\lambda)) = m^2 + (\sqrt{\Sigma} - \sqrt{1/\lambda})^2$. The mean piece decays at the mean rate $|\Re\,\nu_+|$; the variance piece decays at the covariance rate, since $\sqrt{\Sigma} - \sqrt{\Sigma_\infty} \sim (\Sigma - \Sigma_\infty)/(2\sqrt{\Sigma_\infty})$ near stationarity. A short check shows the mean piece dominates, giving

$$\text{rate}_{W_2} \;=\; \begin{cases}\gamma/2 & \gamma \le 2\sqrt{\lambda}, \\ \big(\gamma - \sqrt{\gamma^2 - 4\lambda}\big)/2 & \gamma > 2\sqrt{\lambda},\end{cases} \tag{10.30}$$

optimized at $\gamma_* = 2\sqrt{\lambda}$ with rate $\sqrt{\lambda}$. Overdamped Langevin $dX_t = -\lambda X_t\,dt + \sqrt{2}\,dW_t$ on the same target has W_2 rate λ, so the continuous-time slow-mode rates differ by a factor $\sqrt{\lambda}$.

Anisotropic Gaussian: continuous-time rates. For an anisotropic Gaussian $\pi = \mathcal{N}(0, \Lambda^{-1})$ on $\mathbb{R}^d$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, the dynamics decouple. The slowest mode controls equilibration:

$$T_{\mathrm{ULA}} \;=\; 1/\lambda_{\min}, \qquad T_{\mathrm{under}} \;=\; 1/\sqrt{\lambda_{\min}} \quad (\text{at } \gamma_* = 2\sqrt{\lambda_{\min}}). \tag{10.31}$$

The discretization story parallels the optimization side, and for suitable discretizations, the underdamped Langevin algorithm has an iteration complexity $O(\sqrt{\kappa})$, for Gaussian targets.

For general strongly log-concave targets, whether sampling admits a genuine $\sqrt{\kappa}$ acceleration — matching the textbook Nesterov rate for optimization — is a longstanding open question. The first continuous-time accelerated result came from a hypocoercivity / PDE analysis of underdamped Langevin dynamics by Cao, Lu and Wang [12], which was subsequently translated to an improved discrete-time guarantee by Zhang, Chewi, Li, Balasubramanian and Erdogdu [73]. Closing the remaining gap to the full $\sqrt{\kappa}$ rate, in discrete time and without restrictive smoothness assumptions, remains an active area; recent progress includes the shifted-composition framework of Altschuler, Chewi and Zhang [3].

Why momentum helps, intuitively. Overdamped Langevin is diffusive: traversing a length ℓ takes time $\sim \ell^2/(\text{diffusion})$. Underdamped Langevin with friction tuned to the slow direction is ballistic over short times: traversal time $\ell/\|P\|$, controlled by velocity. Replacing ℓ^2 by ℓ along the long axis is the $\sqrt{\kappa}$ gain.

HMC as a variant. Hamiltonian Monte Carlo (HMC) [53] alternates two moves: deterministic Hamiltonian flow on (X, P) with $H(x, p) = V(x) + \frac{1}{2}\|p\|^2$ for time $T > 0$ (typically integrated by leapfrog), followed by momentum refresh $P \leftarrow \xi \sim \mathcal{N}(0, I)$. There is no friction during the deterministic phase; dissipation comes entirely from the refresh. HMC achieves $\sqrt{\kappa}$-type rates on Gaussian targets via the same mechanism.

10.4 Preconditioning: Metric and Ensembles

Acceleration improves the dependence of condition numbers by changing the order of the dynamics. *Preconditioning* attacks the same problem differently: deform the geometry so the target looks isotropic. In Euclidean optimization this includes the Newton / natural-gradient family. Two popular routes carry over to sampling: *metric preconditioning* (specify a metric tensor) and *ensemble preconditioning* (build a sampler that automatically adapts to the geometry via interactions within a population).

Recap from Section 10.1: metric preconditioning in optimization. With a constant metric $M \succ 0$ the preconditioned gradient flow (10.5) becomes $\dot{x} = -M^{-1}\nabla V(x)$. When $V(x) = \frac{1}{2}x^\top \Lambda x$, choosing $M = \Lambda$ turns the flow into $\dot{x} = -x$, independent of Λ. The convergent time collapses from κ to $O(1)$ — preconditioning eliminates the condition number entirely.

Metric preconditioning in sampling: Riemannian Langevin. The analogous construction on the sampling side uses a position-dependent metric $M(x) \succ 0$:

$$dX_t = \big[-M(X_t)\nabla V(X_t) + \nabla\cdot M(X_t)\big]\,dt + \sqrt{2M(X_t)}\,dW_t, \tag{10.32}$$

where $(\nabla\cdot M)_i := \sum_j \partial_j M_{ij}$ is an Itô correction that keeps π invariant; see our exercise. For constant M the divergence term vanishes: $dX_t = -M\nabla V(X_t)\,dt + \sqrt{2M}\,dW_t$.

Worked example: anisotropic Gaussian. Let $\pi = \mathcal{N}(0,\Sigma)$ with $\Sigma = \mathrm{diag}(\lambda_1,\dots,\lambda_d)$, $\lambda_{\max} = 1$, $\kappa = 1/\lambda_{\min}$. Plain Langevin $dX = -\Sigma^{-1}X\,dt + \sqrt{2}\,dW$ has mixing time $\asymp \kappa$ (slowest mode $\lambda_{\min}$). Preconditioned Langevin with $M = \Sigma$ becomes $dX_t = -X_t\,dt + \sqrt{2\Sigma}\,dW_t$, exactly plain Langevin in the rescaled coordinates $Y = \Sigma^{-1/2}X$ where π becomes $\mathcal{N}(0,I)$. Mixing time is $O(1)$ — κ-independent, exactly the optimization analogue.

Riemannian Langevin often requires choosing M via inverse Fisher information, expected Hessian, or local empirical estimates; each comes with computational cost, positive-definiteness worries, and the Itô correction $\nabla\cdot M$.

10.4.1 Affine invariance: a precise definition

Before describing ensemble methods we make the notion of affine invariance, used loosely in the previous draft of this section, precise. Let $\mathbf{X} = (X^{(1)},\dots,X^{(L)}) \in (\mathbb{R}^d)^L$ denote an ensemble of L walkers, and consider a Markov process

$$d\mathbf{X}_t = b(\mathbf{X}_t)\,dt + \sigma(\mathbf{X}_t)\,d\mathbf{W}_t \tag{10.33}$$

on $(\mathbb{R}^d)^L$ targeting the product distribution $\pi^{\otimes L}(\mathbf{x}) = \prod_\ell \pi(x^{(\ell)})$. Any invertible affine map

$$\phi_{A,c}(x) := Ax + c, \quad A \in GL(d),\ c \in \mathbb{R}^d, \tag{10.34}$$

acts on the ensemble coordinate-wise: $\Phi_{A,c}(\mathbf{X}) := (\phi_{A,c}(X^{(1)}),\dots,\phi_{A,c}(X^{(L)}))$.

Definition 10.2 (Affine invariance [34])**.** *The ensemble process* (10.33) *is* affine invariant *if, for every invertible affine $\phi_{A,c}$, the law of $\Phi_{A,c}(\mathbf{X}_t)$ under the SDE targeting $\pi^{\otimes L}$ equals the law of the SDE* (10.33) *run* with the same drift and noise functions, but targeting the pushforward $(\phi_{A,c})_\#\pi^{\otimes L} = ((\phi_{A,c})_\#\pi)^{\otimes L}$*, with the corresponding affine-transformed initial condition.*

Concretely, this requires the drift to transform like a vector field and the diffusion like a $(1,1)$-tensor under $\Phi_{A,c}$:

$$b(\Phi_{A,c}(\mathbf{X})) = A\,b_\pi(\mathbf{X})\big|_{\pi\leftarrow(\phi_{A,c})_\#\pi}, \quad \sigma\sigma^\top(\Phi_{A,c}(\mathbf{X})) = A\,\sigma\sigma^\top_\pi(\mathbf{X})\,A^\top\big|_{\pi\leftarrow(\phi_{A,c})_\#\pi}. \tag{10.35}$$

The practical consequence is what makes the definition useful: every member of the affine orbit $\{(\phi_{A,c})_\#\pi : A \in GL(d), c \in \mathbb{R}^d\}$ is "the same problem" for the sampler. In particular, an affine-invariant sampler cannot distinguish $\mathcal{N}(0,\Sigma)$ from $\mathcal{N}(0,I)$ for any $\Sigma \succ 0$: its iteration count on any anisotropic Gaussian is identical to its iteration count on the isotropic one. This is the sampling analogue of Newton's method in optimization, whose iterates are affine-equivariant.

10.4.2 Ensemble preconditioning: affine invariant ensemble Langevin

How does one design an affine-invariant SDE? The cleanest answer is to build the metric M from objects that transform covariantly with the ensemble. The empirical mean and covariance,

$$\bar{X}(\mathbf{X}) := \frac{1}{L}\sum_{\ell=1}^{L} X^{(\ell)}, \qquad C(\mathbf{X}) := \frac{1}{L}\sum_{\ell=1}^{L}\big(X^{(\ell)}-\bar{X}\big)\big(X^{(\ell)}-\bar{X}\big)^{\top}, \tag{10.36}$$

are the simplest such objects: under $\mathbf{X} \mapsto \Phi_{A,c}(\mathbf{X})$,

$$\bar{X}(\Phi_{A,c}(\mathbf{X})) = A\bar{X}(\mathbf{X}) + c, \qquad C(\Phi_{A,c}(\mathbf{X})) = A\,C(\mathbf{X})\,A^{\top}. \tag{10.37}$$

That is, the empirical mean is affine-equivariant as a point and the empirical covariance is affine-equivariant as a quadratic form, exactly the transformation rules required by (10.35) when we use $M = C(\mathbf{X})$ as the preconditioner.

Proposition 10.3 (ALDI: affine-invariant Langevin [31])**.** *Let $\mathbf{X}_t = (X_t^{(1)},\dots,X_t^{(L)})$ evolve according to the coupled SDE*

$$dX_t^{(\ell)} \;=\; -\,C(\mathbf{X}_t)\,\nabla V(X_t^{(\ell)})\,dt \;+\; \frac{d+1}{L}\big(X_t^{(\ell)}-\bar{X}(\mathbf{X}_t)\big)\,dt \;+\; \sqrt{2\,C(\mathbf{X}_t)}\,dW_t^{(\ell)}, \tag{10.38}$$

for $\ell = 1,\dots,L$, with $\{W^{(\ell)}\}$ independent Brownian motions, $L > d$, and any matrix square root $\sqrt{2C(\mathbf{X}_t)}$. Then:

(i) $\pi^{\otimes L}$ is stationary;

(ii) the dynamics (10.38) is affine invariant in the sense of Definition 10.2.

Equation (10.38) is the ensemble *Riemannian Langevin* (10.32) applied to each walker with metric $M = C(\mathbf{X}_t)$ shared across the ensemble. The first term is the preconditioned gradient drift; the third is the corresponding multiplicative noise. The middle term — the "$(d+1)/L$" drift — is the Itô correction $\nabla\cdot M$ from (10.32), computed for the empirical-covariance metric on the joint state space $(\mathbb{R}^d)^L$: a direct calculation gives $[\nabla\cdot C(\mathbf{X})]_\ell = \frac{d+1}{L}(X^{(\ell)}-\bar{X})$, which is what guarantees $\pi^{\otimes L}$ is invariant.

Affine invariance from (10.37). Under $\mathbf{X} \to \Phi_{A,c}(\mathbf{X})$, the four terms of (10.38) transform consistently:

- Gradient drift: $-C(\mathbf{X})\nabla V(X^{(\ell)}) \to -A\,C(\mathbf{X})A^{\top}\nabla\tilde{V}(AX^{(\ell)}+c)$, where $\tilde{V}$ is the potential of $(\phi_{A,c})_{\#}\pi$. Since $\nabla\tilde{V}(\phi_{A,c}(x)) = A^{-\top}\nabla V(x)$, this equals $-A\,C(\mathbf{X})\nabla V(X^{(\ell)})$ — the $A^{\top}$ and $A^{-\top}$ cancel, and the new drift is exactly A times the old drift, as required by (10.35).
- Itô correction: $\frac{d+1}{L}(X^{(\ell)}-\bar{X}) \to \frac{d+1}{L}(AX^{(\ell)}-A\bar{X}) = A\cdot\frac{d+1}{L}(X^{(\ell)}-\bar{X})$.
- Diffusion: $\sqrt{2C(\mathbf{X})}\,dW^{(\ell)} \to \sqrt{2A\,C(\mathbf{X})A^{\top}}\,dW^{(\ell)}$, whose covariance is $A\,(2C(\mathbf{X}))\,A^{\top}$ — again the transformation rule (10.35).

Affine invariance is built in by construction; no metric to estimate externally, no inverse or square root of an explicit preconditioner.

Affine invariant ensemble samplers The affine-invariant Langevin diffusion of [31, 30] uses the empirical covariance of the ensemble as the preconditioner M; ensemble underdamped Langevin schemes [45] similarly precondition the kinetic dynamics. A recent line of affine-invariant ensemble HMC algorithms [17] combines Goodman–Weare's ensemble preconditioning with HMC which achieves $O(d^{1/4})$ autocorrelation time complexity for any d-dimensional Gaussian distributions.

The classical ancestor of all of these is the affine-invariant *stretch move* of Goodman–Weare [34], which is gradient-free and uses Metropolis acceptance to correct discretization bias. We present the stretch move alongside Metropolis in the Metropolis correction section.

Week 7 Exercises

1. **(Nesterov's vanishing-friction ODE.)** This exercise works out the continuous-time analysis of Nesterov's accelerated gradient method, which has a different ODE limit from Polyak's heavy ball.

   Nesterov's iteration $y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$, $x_{k+1} = y_k - h\,\nabla V(y_k)$ converges, under the scaling $t = k\sqrt{h}$, to the second-order ODE [68]

   $$\ddot{x}_t + \frac{3}{t}\,\dot{x}_t + \nabla V(x_t) = 0. \tag{10.39}$$

   Note the scaling $t = k\sqrt{h}$ (one continuous time unit per $1/\sqrt{h}$ iterations), as opposed to $t = kh$ for plain gradient descent and Polyak heavy ball. The friction $\gamma_t = 3/t$ is *time-varying*: large at early times (heavily damped while V is being explored) and decaying to zero (allowing late-time ballistic motion through flat regions).

   (a) *(Lyapunov function for the convex case.)* Assume V is convex (not necessarily strongly convex) with minimizer x^*. Define the energy

   $$\mathcal{E}(t) = t^2\big(V(x_t) - V(x^*)\big) + 2\big\|x_t + \tfrac{t}{2}\dot{x}_t - x^*\big\|^2.$$

   Differentiate $\mathcal{E}(t)$ to show

   $$\dot{\mathcal{E}}(t) = 2t\big[\big(V(x_t) - V(x^*)\big) - \langle x_t - x^*,\, \nabla V(x_t)\rangle\big].$$

   Then use convexity of V, equivalently $V(x^*) \geq V(x) + \langle x^* - x, \nabla V(x)\rangle$, to conclude $\dot{\mathcal{E}}(t) \leq 0$.

   (b) *($O(1/t^2)$ rate.)* From part (a), $\mathcal{E}(t) \leq \mathcal{E}(0)$. The ODE (10.39) is singular at $t = 0$ (the friction coefficient $3/t$ blows up), and the standard initial condition is $\dot{x}_0 = 0$; in any case, the term $\frac{t}{2}\dot{x}_t$ in $\mathcal{E}(t)$ vanishes at $t = 0$, so $\mathcal{E}(0) = 2\|x_0 - x^*\|^2$. From $\mathcal{E}(t) \geq t^2(V(x_t) - V(x^*))$, conclude

   $$V(x_t) - V(x^*) \;\leq\; \frac{2\|x_0 - x^*\|^2}{t^2}. \tag{10.40}$$

   This is the continuous-time analogue of Nesterov's $O(1/k^2)$ discrete rate. Compare with the $O(1/t)$ rate of plain gradient flow on convex problems (Week 2 Exercise 2(a)(iii)) and explain why the t^2 weight in $\mathcal{E}$ is matched specifically to the $3/t$ friction in (10.39): with constant friction γ, no such quadratic-in-t Lyapunov function works.

   (c) *(Strongly convex case and connection to heavy ball.)* For α-strongly convex V, Nesterov's accelerated method uses a different schedule with constant momentum coefficient $\beta = (1 - \sqrt{\alpha h})/(1 + \sqrt{\alpha h})$, $y_k = x_k + \beta(x_k - x_{k-1})$, $x_{k+1} = y_k - h\nabla V(y_k)$. Substitute the iteration and expand $\beta = 1 - 2\sqrt{\alpha h} + O(\alpha h)$

for small h. Under the time scaling $t = k\sqrt{h}$, match the $\sqrt{h}$ coefficients in the Taylor expansion to show that the leading-order ODE limit is

$$\ddot{x}_t + 2\sqrt{\alpha}\,\dot{x}_t + \nabla V(x_t) = 0,$$

the constant-friction heavy ball (10.21) at critical damping $\gamma = 2\sqrt{\alpha}$ analyzed in Section 10.2. Conclude that, in the strongly convex case, the continuous-time skeletons of Nesterov-SC and Polyak heavy ball coincide; the $\gamma_t = 3/t$ schedule of (10.39) is specifically tuned to the convex (non-strongly-convex) regime.

2. (Stationary distribution of underdamped Langevin.) This exercise verifies that the joint Gibbs measure $\pi(x) \otimes \mathcal{N}(0, I)(p)$ is invariant for the underdamped Langevin SDE (10.27), by deriving its Fokker–Planck equation and showing $\pi(x) \otimes \mathcal{N}(0, I)(p)$ solves the stationary equation. This justifies the precise balance between friction and noise that distinguishes underdamped Langevin from a generic damped Newtonian system.

Recall the SDE on $(X_t, P_t) \in \mathbb{R}^{2d}$:

$$dX_t = P_t\,dt, \qquad dP_t = -\nabla V(X_t)\,dt - \gamma P_t\,dt + \sqrt{2\gamma}\,dW_t,$$

where W_t is a d-dimensional Brownian motion and $\gamma > 0$. Denote the joint density of (X_t, P_t) by $\rho_t(x, p)$.

(a) *(Fokker–Planck equation.)* Apply Itô's formula to a test function $\phi(x, p) \in C_c^\infty(\mathbb{R}^{2d})$ along (X_t, P_t) to compute $\frac{d}{dt}\mathbb{E}[\phi(X_t, P_t)]$. Equating with $\int \phi\,\partial_t\rho_t\,dx\,dp$ and integrating by parts in x and in p separately, derive

$$\partial_t\rho_t = -p \cdot \nabla_x\rho_t + \nabla_p \cdot \big[(\nabla V(x) + \gamma p)\rho_t\big] + \gamma\Delta_p\rho_t. \tag{10.41}$$

Identify the three terms with the three components of the drift–diffusion: transport in x at velocity p, restoring force plus friction in p, and momentum noise.

(b) *(Stationarity of $\pi \otimes \mathcal{N}(0, I)$.)* Let

$$\rho_\infty(x, p) := \pi(x)\,\mathcal{N}(p; 0, I) = \pi(x) \cdot (2\pi)^{-d/2} e^{-\|p\|^2/2}.$$

Compute the four relevant gradients $\nabla_x\rho_\infty$, $\nabla_p\rho_\infty$, $\Delta_p\rho_\infty$, and $\nabla_p \cdot (p\,\rho_\infty)$. Substitute into the right-hand side of (10.41) and verify, term by term, that each piece cancels:

- The transport term $-p \cdot \nabla_x\rho_\infty = p \cdot \nabla V(x)\,\rho_\infty$ cancels against $\nabla_p(\nabla V(x)\,\rho_\infty) = -p \cdot \nabla V(x)\,\rho_\infty$ (since $\nabla V(x)$ is independent of p and $\nabla_p\rho_\infty = -p\,\rho_\infty$).
- The friction term $\gamma\nabla_p \cdot (p\,\rho_\infty) = \gamma(d - \|p\|^2)\rho_\infty$ cancels against the noise term $\gamma\Delta_p\rho_\infty = \gamma(\|p\|^2 - d)\rho_\infty$.

Conclude $\partial_t\rho_\infty = 0$: the joint Gibbs $\pi \otimes \mathcal{N}(0, I)$ is stationary for underdamped Langevin, for every $\gamma > 0$.

(c) *(Role of the noise coefficient $\sqrt{2\gamma}$.)* Repeat the calculation of part (b) with a general noise coefficient σ in place of $\sqrt{2\gamma}$ (so the diffusion term in (10.41) becomes $\frac{\sigma^2}{2}\Delta_p\rho_t$). Show that the cancellation in the $(p, \|p\|^2)$ terms requires precisely $\sigma^2 = 2\gamma$. This is the *fluctuation–dissipation relation*: the noise and friction must balance to keep $\pi \otimes \mathcal{N}(0, I)$ invariant; any other ratio gives a different stationary distribution (or none). The marginal in X is the target π precisely because of this balance.

3. (Stationarity of Riemannian Langevin.) This exercise verifies that the Riemannian Langevin SDE (10.32) preserves the target $\pi \propto e^{-V}$ for any smooth $x \mapsto M(x) \succ 0$, justifying the Itô correction $\nabla\cdot M$ in the drift. The Fokker–Planck equation for an SDE with state-dependent diffusion has an extra term that the correction is designed to cancel.

Recall the SDE

$$dX_t = \big[-M(X_t)\nabla V(X_t) + \nabla\cdot M(X_t)\big]\,dt + \sqrt{2M(X_t)}\,dW_t,$$

where $(\nabla\cdot M)_i := \sum_j \partial_j M_{ij}$ and $\sqrt{2M(x)}$ is any matrix square root of $2M(x)$.

(a) *(Fokker–Planck for general SDE with state-dependent diffusion.)* For a general SDE $dX_t = b(X_t)\,dt + \sigma(X_t)\,dW_t$ with $D(x) := \frac{1}{2}\sigma(x)\sigma(x)^\top$, the Fokker–Planck equation is

$$\partial_t\rho_t = -\nabla\cdot(b\,\rho_t) + \sum_{i,j}\partial_i\partial_j\big(D_{ij}\rho_t\big). \tag{10.42}$$

Derive (10.42) by applying Itô's formula to a test function $\phi(x)$ along X_t and integrating by parts twice. (Note: each Hessian term contributes via $\frac{1}{2}\sigma\sigma^\top : \nabla^2\phi$, and integration by parts moves both derivatives onto ρ_t.)

(b) *(Specialize to Riemannian Langevin.)* With drift $b(x) = -M(x)\nabla V(x) + \nabla\cdot M(x)$ and $\sigma(x)\sigma(x)^\top = 2M(x)$ (so $D = M$), substitute into (10.42) to obtain

$$\partial_t\rho_t = \nabla\cdot\big[(M\nabla V - \nabla\cdot M)\rho_t\big] + \sum_{i,j}\partial_i\partial_j(M_{ij}\,\rho_t). \tag{10.43}$$

(c) *(Verify π is stationary.)* We claim the second term in (10.43) expands to

$$\sum_{i,j}\partial_i\partial_j(M_{ij}\rho_t) = \nabla\cdot\big[(\nabla\cdot M)\,\rho_t\big] + \nabla\cdot\big[M\nabla\rho_t\big].$$

Verify this identity by computing $\partial_j(M_{ij}\rho_t) = (\partial_j M_{ij})\rho_t + M_{ij}\partial_j\rho_t$, then ∂_i of this sum, and grouping the terms by what they apply ∂_i to.

Substituting this identity back into (10.43), the $\nabla\cdot M$ terms in the drift and diffusion *cancel*, leaving

$$\partial_t\rho_t = \nabla\cdot\big[M\nabla V\,\rho_t + M\nabla\rho_t\big] = \nabla\cdot\big[\rho_t\,M\,\nabla\log(\rho_t/\pi)\big], \tag{10.44}$$

using $\pi \propto e^{-V}$ so $\nabla V = -\nabla \log \pi$ and $\nabla V + \nabla \log \rho_t = \nabla \log(\rho_t/\pi)$. Conclude that $\rho_t = \pi$ makes the right-hand side vanish, i.e. π is stationary for Riemannian Langevin.

(d) *(Why the Itô correction is needed.)* Drop the $\nabla \cdot M$ term from the drift, i.e. run the "naive" SDE $dX_t = -M(X_t)\nabla V(X_t)\,dt + \sqrt{2M(X_t)}\,dW_t$. Repeating the calculation in (10.43) without the $\nabla \cdot M$ term, show that the analogue of (10.44) picks up an extra $\nabla \cdot [(\nabla \cdot M)\rho_t]$ piece, which does not vanish at $\rho_t = \pi$ unless $\nabla \cdot M \equiv 0$. Conclude that the Itô correction is exactly what is needed to keep π invariant under a position-dependent metric.

(e) *(Sanity check.)* Specialize to the constant-metric case $M(x) \equiv M_0$, a constant SPD matrix. Show that the Itô correction vanishes, $\nabla \cdot M_0 = 0$, and the SDE reduces to $dX_t = -M_0\nabla V(X_t)\,dt + \sqrt{2M_0}\,dW_t$. Apply the change of variables $Y_t = M_0^{-1/2}X_t$ and compute that Y_t satisfies plain Langevin with potential $\tilde{V}(y) = V(M_0^{1/2}y)$. Conclude that, in this constant-metric case, Riemannian Langevin is plain Langevin in rescaled coordinates — justifying the "preconditioning" interpretation of Section 10.4 and recovering the Gaussian worked example where $M_0 = \Sigma$ kills the condition number entirely.

11 Variational Inference

In Section 10 we cast sampling as minimization of $D_{\mathrm{KL}}(\cdot \,\|\, \pi)$ over the full space $\mathcal{P}(\mathbb{R}^d)$, and realized the descent via dynamics like Langevin which is the Wasserstein gradient flow. *Variational inference* (VI) takes the step further: solve the same optimization problem but *constrained to a parametric family* $\Pi_\Theta = \{q_\theta : \theta \in \Theta\}$ of tractable distributions, obtaining

$$\theta^* \in \arg\min_{\theta \in \Theta} D_{\mathrm{KL}}(q_\theta \,\|\, \pi)\,. \tag{11.1}$$

The minimizer q_{θ^*} is a fitted approximation to π rather than a stream of samples. The cost of the restriction is bias: q_{θ^*} generally differs from π. The benefit is that q_{θ^*} is cheap to evaluate (a fixed parametric formula) and cheap to sample from, and that the optimization (11.1) can be performed by standard stochastic gradient methods on θ in $\mathbb{R}^p$.

Connection to SOC: VI in path space versus static space. The reader has already seen one form of VI: the SOC formulation of Section 9.7 cast posterior sampling in generative sampling as KL minimization *on path space*,

$$\min_u \; D_{\mathrm{KL}}\big(\mathbb{P}^u \,\|\, \mathbb{P}^R\big)$$

(9.13), where $\mathbb{P}^u$ is a controlled path measure indexed by a drift $u_t(\cdot, \theta)$ (the variational parameter) and $\mathbb{P}^R$ is the target tilted path measure with the desired terminal marginal. The present section is the *static-space* analogue: the variational object is a distribution $q_\theta \in \mathcal{P}(\mathbb{R}^d)$ rather than a path measure on $C([0,1]; \mathbb{R}^d)$, and the target is the static $\pi \in \mathcal{P}(\mathbb{R}^d)$ rather than $\mathbb{P}^R$. The two problems share the same parametric KL-minimization structure, and the gradient machinery developed below has direct counterparts in the SOC pathwise-gradient framework (9.4). We will draw the analogy explicitly in Section 11.2.

We first state the VI objective and derive the gradient (11.1), then review parametric families (Gaussian, Gaussian mixture, normalizing flows) and the black-box VI estimator. The remainder of the section develops the Gaussian case in depth, the equivalence between Gaussian VI and Gaussian moment-closure of probability-space gradient flows, and concludes with two preconditioned VI algorithms — natural gradient and Bures–Wasserstein gradient — which arise as natural moment closures of the Fisher–Rao and Wasserstein gradient flows respectively.

11.1 The VI objective and ELBO

Write the target as $\pi(x) = e^{-V(x)}/Z$ with $Z = \int e^{-V}$ an intractable normalization constant. The KL divergence to a variational density q_θ splits as

$$D_{\mathrm{KL}}(q_\theta \,\|\, \pi) = \mathbb{E}_{q_\theta}[\log q_\theta(X)] + \mathbb{E}_{q_\theta}[V(X)] + \log Z =: -\mathcal{L}(\theta) + \log Z, \tag{11.2}$$

where $\mathcal{L}(\theta) := -\mathbb{E}_{q_\theta}[\log q_\theta(X)] - \mathbb{E}_{q_\theta}[V(X)]$ can be referred to as *evidence lower bound* (ELBO). Since $\log Z$ is a constant in θ, minimizing $D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$ is equivalent to maximizing $\mathcal{L}(\theta)$, and the latter does not require knowledge of Z.

The ELBO decomposes as $\mathcal{L} = -\mathbb{E}_{q_\theta}[V] + \mathcal{H}(q_\theta)$, where $\mathcal{H}(q_\theta) = -\mathbb{E}_{q_\theta}[\log q_\theta]$ is the differential entropy of q_θ. The first term pulls q_θ towards low-potential regions of π; the second term, the entropy regularizer, prevents q_θ from collapsing to a delta. The balance is the same one that drives the Langevin SDE (10.13): drift $-\nabla V$ (low-potential preference) plus diffusion $\sqrt{2}\,dW$ (entropy-spreading). VI optimizes this balance over a parametric family rather than letting it equilibrate over $\mathcal{P}(\mathbb{R}^d)$.

Reverse KL is mode-seeking. The choice $D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$ in (11.1) (the *reverse KL*) is not the only option. The *forward KL* $D_{\mathrm{KL}}(\pi \,\|\, q_\theta)$ instead leads to maximum-likelihood-style inference and is mode-covering: it penalizes q_θ heavily wherever π has mass but q_θ does not. The reverse KL penalizes wherever q_θ has mass but π does not, making it mode-seeking: it tends to fit one mode of a multimodal π rather than all modes. The forward KL is the natural choice for generative modeling (when samples from π are available and the score $\log q_\theta$ is to be optimized); the reverse KL is the natural choice for Bayesian inference (when $\pi =$ posterior is queryable but not sampleable).

11.2 Gradient of the VI objective

Two standard estimators express $\nabla_\theta \mathcal{L}$ as an expectation over q_θ, making stochastic gradient descent applicable. Both have direct counterparts in the path-space SOC gradient framework of Section 9.7.

Score-function estimator. Differentiating $\mathbb{E}_{q_\theta}[V(X)]$ requires moving ∇_θ inside an expectation whose underlying measure also depends on θ. Starting from $D_{\mathrm{KL}}(q_\theta \,\|\, \pi) = \int q_\theta \log(q_\theta/\pi)\,dx$ and applying the product rule under the integral,

$$\nabla_\theta D_{\mathrm{KL}}(q_\theta \,\|\, \pi) = \int \nabla_\theta q_\theta \cdot \log \frac{q_\theta}{\pi}\,dx \;+\; \int q_\theta\, \nabla_\theta \log \frac{q_\theta}{\pi}\,dx. \tag{11.3}$$

Two simplifications make this estimable. First, the *log-derivative trick* $\nabla_\theta q_\theta = q_\theta\, \nabla_\theta \log q_\theta$ turns the first integral into an expectation under q_θ. Second, the second integral $\int q_\theta\, \nabla_\theta \log(q_\theta/\pi) = \int q_\theta\, \nabla_\theta \log q_\theta = \nabla_\theta \int q_\theta = \nabla_\theta(1) = 0$ *vanishes identically*, since π has no θ-dependence and the score has mean zero under its own measure. Combining,

$$\nabla_\theta D_{\mathrm{KL}}(q_\theta \,\|\, \pi) \;=\; \mathbb{E}_{q_\theta}\Big[\log \frac{q_\theta(X)}{\pi(X)}\, \nabla_\theta \log q_\theta(X)\Big], \tag{11.4}$$

the *score-function* or *REINFORCE* gradient. The expectation can be replaced by a Monte Carlo average over draws $X^{(1)}, \dots, X^{(K)} \sim q_\theta$, giving a fully *black-box* estimator that needs only (i) the ability to sample q_θ and (ii) evaluation of $\log q_\theta, \nabla_\theta \log q_\theta, \log \pi$.

Two practical features.

- *Normalizing constant of π drops out.* If we replace $\log \pi$ by the unnormalized $\log \tilde{\pi} = -V$ (so that $\log \pi = -V - \log Z$ for some unknown Z), the extra $-\log Z$ inside the bracket contributes $-\log Z \cdot \mathbb{E}_{q_\theta}[\nabla_\theta \log q_\theta] = 0$ by the same mean-zero score identity. So (11.4) is exactly $\mathbb{E}_{q_\theta}[(\log q_\theta + V)\nabla_\theta \log q_\theta]$ — the unknown normalization Z never appears.
- *Control variates are free.* For any constant b (or any $b(X)$ that depends only on X and not on θ), $\mathbb{E}_{q_\theta}[b\,\nabla_\theta \log q_\theta] = 0$ by the same identity. So one can subtract any *baseline* from $\log q_\theta - \log \pi$ inside the expectation without changing the gradient. A common choice is the running average $b_k \approx \mathbb{E}_{q_\theta}[\log q_\theta - \log \pi] = -\mathcal{L}(\theta_k)$, which can substantially reduce the variance of Monte Carlo estimates of (11.4). This is the basis for practical REINFORCE-style optimization.

Reparametrization estimator. When q_θ admits a sampler of the form $X = T_\theta(\xi)$ with $\xi \sim \nu$ a fixed base distribution (e.g. $\xi \sim \mathcal{N}(0, I)$ and $T_\theta(\xi) = m + L\xi$ for $C = LL^\top$), the expectation can be rewritten with the randomness moved outside the θ-dependence: $\mathbb{E}_{q_\theta}[h(X)] = \mathbb{E}_\nu[h(T_\theta(\xi))]$. For $\mathcal{F}(\theta) := D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$ this gives

$$\nabla_\theta \mathcal{F}(\theta) = \mathbb{E}_\nu\big[\nabla V(T_\theta(\xi)) \cdot \nabla_\theta T_\theta(\xi)\big] + \nabla_\theta \mathbb{E}_\nu\big[\log q_\theta(T_\theta(\xi))\big], \tag{11.5}$$

the first term coming from $\mathbb{E}_{q_\theta}[V(X)] = \mathbb{E}_\nu[V(T_\theta(\xi))]$ and the second from the entropy $-\mathcal{H}(q_\theta) = \mathbb{E}_{q_\theta}[\log q_\theta(X)]$. The first term uses only the gradient of V, not V itself, so the normalizing constant $\log Z$ of $\pi = e^{-V}/Z$ automatically does not enter. The entropy term admits a closed form for Gaussians, mixtures, and normalizing flows with tractable Jacobians, which is why these are the canonical reparametrization-friendly families. This is the *reparametrization trick* [43, 57], the backbone of modern VI in deep models. It requires T_θ to be differentiable in θ for almost every ξ, which excludes certain discrete or hard-thresholded families but accommodates Gaussians, mixture parametrizations, and normalizing flows. In practice the reparametrization estimator yields substantially lower variance than the score-function estimator (11.4) when both apply.

Analogy with path-space gradients in SOC. Both (11.4) and (11.5) have exact counterparts in the SOC pathwise-gradient framework (Section 9.7), and the analogy is worth making explicit because the two settings illuminate each other.

In SOC, the variational object is the controlled path measure $\mathbb{P}^u$ on $C([0,1];\mathbb{R}^d)$, parametrized by a drift $u_t(\cdot, \theta)$. A trajectory is generated by

$$X^u_t \;=\; T_\theta(W_{[0,t]}), \qquad dX^u_t = (b_t(X^u_t) + \sigma_t u_t(X^u_t, \theta))\,dt + \sigma_t\,dW_t, \tag{11.6}$$

i.e. the trajectory is a deterministic functional T_θ of the Brownian noise W. This is the *path-space reparametrization*: W plays the role of the base random variable ξ in (11.5); the SDE solution map T_θ plays the role of the static reparametrization map; and the objective $J(\theta) = \mathbb{E}^{\mathbb{P}^u}[\text{cost}]$ plays the role of $\mathbb{E}_{q_\theta}[V(X)]$. Differentiating through T_θ with

W held fixed gives the *pathwise gradient* (also called the *adjoint gradient*, since the chain rule through the SDE trajectory is most efficiently computed via a backward ODE — the adjoint equation of Section 9.7).

The score-function analogue uses *Girsanov's theorem*: the Radon–Nikodym derivative of $\mathbb{P}^u$ with respect to a reference $\mathbb{P}^0$ is

$$\frac{d\mathbb{P}^u}{d\mathbb{P}^0}(W) = \exp\!\left(\int_0^1 u_t \cdot dW_t - \tfrac{1}{2}\int_0^1 \|u_t\|^2\, dt\right),$$

the path-space analogue of $q_\theta(x)/\nu(x)$. Differentiating this log-density in θ gives a REINFORCE-style estimator on path space, used when the cost is not differentiable in trajectories or when discontinuous controls (jumps, discrete-time policies) are involved.

	Static VI (this section)	**Path-space VI** (SOC, Section 9.7)
Variational object	$q_\theta \in \mathcal{P}(\mathbb{R}^d)$	$\mathbb{P}^u \in \mathcal{P}(C([0,1];\mathbb{R}^d))$
Target	π	$\mathbb{P}^R$
Base randomness	$\xi \sim \nu$	Brownian W
Reparam. map	$X = T_\theta(\xi)$	$X^u_{[0,1]} = T_\theta(W)$ via SDE
Reparam. gradient	(11.5)	adjoint equation
Score-function gradient	log-derivative trick	Girsanov derivative
Objective	ELBO / KL	SOC cost (9.13)

In both settings, reparametrization gradients have substantially lower variance than score-function gradients when applicable (the trade-off being differentiability of the cost / map), and are the default in modern implementations. The conceptual unification is that "static VI" and "path-space SOC" are the same problem distinguished only by the dimension of the variational object; the same gradient estimators apply, with the static T_θ replaced by an SDE flow map.

11.3 Examples of variational families

The art of VI lies in choosing Π_Θ rich enough to approximate π well, but tractable enough that (11.1) can be solved. Four canonical families:

Mean field. $q_\theta(x) = \prod_{i=1}^d q_{\theta_i}(x_i)$, a product of one-dimensional marginals. The simplest VI family, dating to the 1990s [40]. Mean field is exact only on factorized π; its bias on correlated π can be severe (it systematically underestimates posterior variance). Useful as a baseline; routinely outperformed by structured families and by rotated mean field VI [20].

Gaussian. $q_{m,C}(x) = \mathcal{N}(x; m, C)$, parametrized by mean and covariance. The dimension of Θ is $d + d(d+1)/2$ (for full covariance), $2d$ (for diagonal), or $d + dr$ (for rank-r

plus diagonal). The Gaussian family is exact on Gaussian targets, and *Laplace's approximation* (a Gaussian centered at the mode with covariance $\nabla^2 V(x^*)^{-1}$) is a one-step special case. For the general non-Gaussian π it captures the location and global scale of mass but misses non-Gaussian features (skewness, heavy tails, multimodality). It is the most-studied VI family.

Gaussian mixtures. $q_\theta(x) = \sum_{k=1}^K w_k \mathcal{N}(x; m_k, C_k)$ with $\sum_k w_k = 1$, $w_k \geq 0$. Strictly more expressive than a single Gaussian; can capture multimodality and asymmetric tails. The parameter count grows linearly in K. Optimization is harder because of the discrete mixture-weight structure (one fix is to use the softmax parametrization $w_k = e^{\eta_k} / \sum_j e^{\eta_j}$); also, mixture VI is prone to "mode collapse" where several components overlap.

Normalizing flows. $q_\theta(x) = (\Phi_\theta)_\# \nu$, where $\Phi_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is a smooth invertible map parametrized by a neural network and ν is a tractable base distribution (e.g. $\mathcal{N}(0, I)$). The density is given by the change-of-variables formula $q_\theta(x) = \nu(\Phi_\theta^{-1}(x)) \, |\det \nabla \Phi_\theta^{-1}(x)|$, which makes both sampling and density evaluation tractable provided Φ_θ is designed so $\det \nabla \Phi_\theta$ is cheap to compute (triangular Jacobians via coupling layers [27]; autoregressive layers [55]). Normalizing flows can in principle approximate any continuous target arbitrarily well as the network width and depth grow, but require careful architecture design and substantial compute. They have dominated VI in modern Bayesian deep learning and density estimation.

11.4 Black-box variational inference

Black-box VI (BBVI) [56] is the generic recipe: replace the gradient (11.4) or (11.5) by a Monte Carlo estimate and run SGD on θ. The pseudocode for one iteration:

(1) Draw $X^{(1)}, \dots, X^{(K)} \sim q_{\theta_k}$.

(2) Compute the gradient estimator

$$\hat{g}_k = \frac{1}{K} \sum_{i=1}^K \big(\log q_{\theta_k}(X^{(i)}) - \log \pi(X^{(i)})\big) \, \nabla_\theta \log q_{\theta_k}(X^{(i)})$$

(or the reparametrization version when applicable).

(3) Update $\theta_{k+1} = \theta_k - h_k \, \hat{g}_k$.

The key practical knobs are (i) the family Π_Θ (Gaussian vs. mixture vs. flow), (ii) the gradient estimator (score-function vs. reparametrization, with the latter much preferred when available), (iii) the batch size K, and (iv) the step schedule h_k (Adam-like schedulers [42] are standard). Variance reduction for the score-function estimator (control variates, Rao-Blackwellization) is essential in high dimensions.

BBVI is widely deployed in probabilistic programming (Stan, Pyro, PyMC), where the user specifies V and an automatic-differentiation framework computes gradients on

the fly. The framework is generic enough to be applied to any model with a differentiable V, at the price of relatively crude convergence guarantees and a strong dependence on the choice of Π_Θ.

11.5 Gaussian VI and mixture VI

The Gaussian family $\Pi_{\mathrm{G}} = \{\mathcal{N}(m, C) : m \in \mathbb{R}^d, C \in \mathcal{S}^d_{++}\}$ deserves a closer look because for this family every quantity in the VI objective is explicit and the gradient takes a clean closed form. We work it out, then discuss which *flow on parameters* one obtains by descending in different geometries on $\Theta = \mathbb{R}^d \times \mathcal{S}^d_{++}$.

The gradient in closed form. For $q = \mathcal{N}(m, C)$ and $\pi \propto e^{-V}$, the reverse-KL objective is

$$\mathcal{F}(m, C) := D_{\mathrm{KL}}(\mathcal{N}(m, C) \,\|\, \pi) = \mathbb{E}_{\mathcal{N}(m,C)}[V(X)] - \tfrac{1}{2} \log\det C + \text{const.} \tag{11.7}$$

The Gaussian entropy contributes the $\log\det C$ term in closed form; the only remaining expectation is $\mathbb{E}[V(X)]$, which is generally non-explicit but easy to estimate by Monte Carlo or quadrature. Differentiating (11.7) via Stein's lemma (for any h, $\mathbb{E}_{\mathcal{N}(m,C)}[(X - m)h(X)^\top] = C\,\mathbb{E}_{\mathcal{N}(m,C)}[\nabla h(X)^\top]$, applied with h respectively the constant 1 and ∇V), one obtains the explicit Euclidean gradients

$$\nabla_m \mathcal{F} = \mathbb{E}_{\mathcal{N}(m,C)}[\nabla V(X)], \qquad \nabla_C \mathcal{F} = \tfrac{1}{2}\,\mathbb{E}_{\mathcal{N}(m,C)}[\nabla^2 V(X)] - \tfrac{1}{2} C^{-1}. \tag{11.8}$$

Vanilla SGD on these gradients is the simplest Gaussian VI algorithm. But the metric on Θ matters: different choices of metric give different parameter flows, with different convergence rates and different stability properties.

Parameter-space gradient flow. The most direct route: identify $\Theta = \mathbb{R}^d \times \mathcal{S}^d_{++}$ with a subset of $\mathbb{R}^{d+d^2}$ and run gradient descent in the ambient Euclidean metric. Substituting (11.8) gives the ODE

$$\dot m = -\mathbb{E}[\nabla V(X)], \qquad \dot C = -\tfrac{1}{2}\,\mathbb{E}[\nabla^2 V(X)] + \tfrac{1}{2}\, C^{-1}. \tag{11.9}$$

Bures–Wasserstein gradient flow. The Wasserstein-2 distance between two Gaussians has the closed form

$$W_2^2\big(\mathcal{N}(m_1, C_1), \mathcal{N}(m_2, C_2)\big) = \|m_1 - m_2\|^2 + \operatorname{tr}\big(C_1 + C_2 - 2(C_1^{1/2} C_2 C_1^{1/2})^{1/2}\big), \tag{11.10}$$

called the *Bures–Wasserstein* (BW) distance on $\mathbb{R}^d \times \mathcal{S}^d_{++}$. It is the restriction of the Wasserstein-2 metric on $\mathcal{P}(\mathbb{R}^d)$ to the Gaussian submanifold. The BW gradient of $\mathcal{F}$ is

$$\operatorname{grad}_{\mathrm{BW}} \mathcal{F} = \big(\nabla_m \mathcal{F},\ 4\,\operatorname{sym}(C\,\nabla_C \mathcal{F})\big), \tag{11.11}$$

and the BW gradient-descent flow on (11.7) becomes [44]

$$\dot m = -\mathbb{E}_{\mathcal{N}(m,C)}[\nabla V(X)], \qquad \dot C = 2I - C\,\mathbb{E}_{\mathcal{N}(m,C)}[\nabla^2 V(X)] - \mathbb{E}_{\mathcal{N}(m,C)}[\nabla^2 V(X)]\, C. \tag{11.12}$$

On a Gaussian target $\pi = \mathcal{N}(m^*, C^*)$, $C_t = C^*$ is a fixed point of (11.12) (plug in $\mathbb{E}[\nabla^2 V] = (C^*)^{-1}$). Two structural features: the parametrization is intrinsic — the same trajectory results whether one stores C, the Cholesky factor L, or $\log C$.

Fisher–Rao gradient flow / natural gradient. A second canonical metric on Π_{G} is the *Fisher–Rao* metric, induced by the Fisher information of the parametric family. For Gaussians, computing $\mathcal{I}(\theta)$ for $\theta = (m, C)$ and inverting gives the natural-gradient flow [19]

$$\dot{m} = -C\,\mathbb{E}_{\mathcal{N}(m,C)}[\nabla V(X)], \qquad \dot{C} = C - C\,\mathbb{E}_{\mathcal{N}(m,C)}[\nabla^2 V(X)]\,C. \tag{11.13}$$

Equivalently, $\frac{d(m,C)}{dt} = -\mathcal{I}(m,C)^{-1}\nabla\mathcal{F}$. The Fisher–Rao gradient is invariant under reparametrization of θ (this is exactly Amari's natural gradient [4]), and on a Gaussian target the flow (11.13) converges to π at a rate independent of the condition number of C^*. It also preserves $C \succ 0$ exactly under the continuous-time flow. The Fisher–Rao flow is also *affine invariant*.

Remark 11.1 (Connection to Gaussian moment closure). The flows (11.12) and (11.13) are not just gradient descents on a parametric objective; they also arise as *Gaussian moment closures* of probability-space gradient flows [44, 19]. Take the Wasserstein gradient flow $\partial_t \rho_t = \nabla\cdot(\rho_t \nabla V) + \Delta\rho_t$ of $D_{\mathrm{KL}}(\cdot\,\|\,\pi)$ from Section 10.1; assume $\rho_t = \mathcal{N}(m_t, C_t)$ remains Gaussian and project the PDE onto its first two moments. The resulting ODEs for (m_t, C_t) are *exactly* (11.12). Repeating the projection for the Fisher–Rao gradient flow of $D_{\mathrm{KL}}(\cdot\,\|\,\pi)$ gives *exactly* (11.13). The reason: the Bures–Wasserstein metric is the restriction of W_2 to the Gaussian submanifold of $\mathcal{P}(\mathbb{R}^d)$, and the Fisher–Rao metric on Gaussians is the restriction of the Fisher–Rao metric on $\mathcal{P}(\mathbb{R}^d)$. Restricting the gradient flow on $\mathcal{P}(\mathbb{R}^d)$ to a parametric family and gradient-descending in the restricted metric give the same parameter ODE.

The moment-closure perspective is often the easiest way to derive these flows in practice, since it avoids working out the Fisher information matrix explicitly; the same recipe applies to other metrics. The Fisher–Rao gradient flow at the PDE level has uniform exponential convergence rates and rich geometric convexity properties, see [13].

Mixture extension. For Gaussian mixtures $q_\theta = \sum_k w_k\,\mathcal{N}(m_k, C_k)$ the same machinery generalizes: each component has BW or Fisher–Rao gradients with respect to (m_k, C_k), and the weights w_k get their own natural-gradient updates from the categorical Fisher information; see [48, 24, 21, 14, 15].

Numerical stability. Gaussian and Gaussian-mixture VI are notoriously fragile in practice. The dominant failure modes are (i) loss of positive-definiteness in C during finite-step gradient updates; (ii) blow-up or vanishing of $\det C$ when V has strongly anisotropic curvature; and (iii) mode collapse for mixtures, when components drift together. Studying stabilization of them is an active research field.

11.6 A Unifying Formal Perspective via VI

We close by stepping back and viewing the methods of this course through the lens of KL minimization. The principle is simple — *optimize a KL divergence between two probability measures, one of which is parametrized* — and the variations along two axes recover essentially all algorithmic frameworks we have studied.

Axis 1: which KL? $D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$ (reverse, q-mass penalized where π has no mass) vs. $D_{\mathrm{KL}}(\pi \,\|\, q_\theta)$ (forward, π-mass penalized where q has no mass). Reverse KL never requires samples from π; the unknown normalization Z cancels. Forward KL expands as

$$D_{\mathrm{KL}}(\pi \,\|\, q_\theta) = \int \pi \log \pi \, dx - \int \pi \log q_\theta \, dx,$$

the first term constant in θ and the second a log-likelihood-style expectation under π. So $\min_\theta D_{\mathrm{KL}}(\pi \,\|\, q_\theta)$ requires samples from π — exactly the data setting of maximum likelihood estimation (MLE).

Axis 2: static or path-space? The variational object lives either on $\mathcal{P}(\mathbb{R}^d)$ (static) or on $\mathcal{P}(C([0,1];\mathbb{R}^d))$ (path measures). The static case has $q_\theta = q_\theta(x)$ a distribution on $\mathbb{R}^d$; the path-space case has $q_\theta = \mathbb{P}^u$ a path measure indexed by a control u.

The 2×2 grid:

	Static $\mathcal{P}(\mathbb{R}^d)$	**Path space** $\mathcal{P}(C([0,1];\mathbb{R}^d))$
Reverse KL $\min D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$	VI (Section 11)	SOC / RL (Section 9.7)
Forward KL $\min D_{\mathrm{KL}}(\pi \,\|\, q_\theta)$	MLE	Diffusion, flow matching, VAE, SFT

(In the forward-KL cases, samples from the target π are required; in the reverse-KL cases, only query access to π via $V = -\log \tilde{\pi}$ is needed.) Let us walk through each cell.

Top-left: Reverse-KL static = VI. This is the present section. π is queryable through $V = -\log \tilde{\pi}$ (typically a Bayesian posterior), q_θ is a parametric family one can sample and evaluate, and one minimizes $D_{\mathrm{KL}}(q_\theta \,\|\, \pi)$ to fit q_θ as an approximation. Sample access to π is not required.

Bottom-left: Forward-KL static = MLE. π is accessible only through data $X_1, \ldots, X_N \sim \pi$, and q_θ is a parametric model. Then

$$\min_\theta D_{\mathrm{KL}}(\pi \,\|\, q_\theta) = \min_\theta \big(\text{const} - \mathbb{E}_\pi[\log q_\theta]\big) \approx \min_\theta \Big(-\tfrac{1}{N} \textstyle\sum_{i=1}^N \log q_\theta(X_i)\Big),$$

the empirical risk for maximum-likelihood estimation. Logistic regression, exponential-family models, and any classical statistical inference with a parametric likelihood live in this cell.

Top-right: Reverse-KL path = SOC / RL. The SOC formulation of Section 9.7 minimizes $D_{\mathrm{KL}}\big(\mathbb{P}^u \,\|\, \mathbb{P}^R\big)$ between a controlled path measure $\mathbb{P}^u$ and a tilted target path measure $\mathbb{P}^R$. The same structure underlies reinforcement learning: a policy π_θ induces a path measure over trajectories, and the entropic RL objective is a KL between this path measure and a tilted target. More generally, expected reward maximization with a KL regularizer reduces to a path-space KL via KL-control duality (9.13). Reinforcement learning from human feedback (RLHF) and other LLM post-training methods are special cases: the canonical objective $\mathbb{E}_{\pi_\theta}[r] - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})$ is exactly reverse-KL path-space VI with π_{ref} playing the role of $\mathbb{P}$ in (9.13).

Bottom-right: Forward-KL path = diffusion, flow matching, VAE, SFT. Score-based diffusion models and flow-matching train by minimizing $D_{\mathrm{KL}}\big(\mathbb{P}^{\mathrm{data}} \,\|\, \mathbb{P}^\theta\big)$ between the true forward noising process and the parametric reverse process. Samples from $\mathbb{P}^{\mathrm{data}}$ are available (the training set is i.i.d. from π^{data}), and the loss reduces to a score-matching or velocity-matching expectation. Variational autoencoders (VAEs) minimize a forward-KL path-space ELBO with an encoder–decoder structure. Supervised fine-tuning (SFT) of large language models is the discrete analogue: data consists of trajectories of tokens i.i.d. from a target distribution (e.g. human-preferred outputs), and the loss is the log-likelihood of these trajectories under the model.

The unifying message. Sampling, inference, generative modeling, control, reinforcement learning, language-model fine-tuning — all instances of *minimize a KL divergence between two probability measures, one of which is parametrized*. The variations are which measure is parametrized (forward vs. reverse KL), which space the measures live on (static $\mathbb{R}^d$ vs. path space), and what is the parametric family (Gaussian, mixture, normalizing flow, drift, neural-network policy, language-model logits). Once one sees this, algorithm design reduces to choosing the four pieces (target measure, variational family, KL direction, gradient estimator), and the machinery developed in this section —score-function / REINFORCE, reparametrization / pathwise, natural / Bures–Wasserstein preconditioning — carries across the entire landscape. Diffusion models, RLHF, BBVI, SOC, and Bayesian inversion are not separate fields; they can be understood as different parametrizations of the same underlying optimization problem.

Week 8 Exercises

Exercise 1 computes explicit convergence rates of the three parameter flows of Section 11.5 on a Gaussian target. Exercise 2 shows that the right coordinate in which the VI objective is convex (for log-concave targets) is the square-root coordinate $A = C^{1/2}$, and identifies this as the underpinning of the Bures–Wasserstein metric. Exercise 3 explains why explicit Euler on the Euclidean parameter flow is unstable and how the Bures–Wasserstein proximal-gradient method fixes this.

1. (Gaussian VI on a Gaussian target: explicit rates.) Let $\pi = \mathcal{N}(m^*, C^*)$, so $V(x) = \frac{1}{2}(x - m^*)^\top (C^*)^{-1}(x - m^*) + \text{const}$ and $\nabla V(x) = (C^*)^{-1}(x - m^*)$, $\nabla^2 V = (C^*)^{-1}$. Then for $q = \mathcal{N}(m, C)$,

$$\mathbb{E}_q[\nabla V(X)] = (C^*)^{-1}(m - m^*), \qquad \mathbb{E}_q[\nabla^2 V(X)] = (C^*)^{-1}.$$

Substitute into the three flows and compute the convergence rate.

(a) *(Euclidean.)* Flow (11.9) becomes

$$\dot{m} = -(C^*)^{-1}(m - m^*), \qquad \dot{C} = \tfrac{1}{2}\big(C^{-1} - (C^*)^{-1}\big).$$

The m-equation is autonomous linear, with explicit solution $m_t - m^* = e^{-t(C^*)^{-1}}(m_0 - m^*)$ and slowest rate $1/\lambda_{\max}(C^*)$.
For C, take $C_0 = \lambda_0 I$ so C_t stays diagonal in the eigenbasis of C^*. The eigenvalues λ_t of C_t (with corresponding $\lambda_\star$ of C^*) satisfy $\dot{\lambda}_t = (1/\lambda_t - 1/\lambda_\star)/2$, which integrates to the implicit closed form [19, (E.21)]

$$\lambda_t - \lambda_\star \;=\; (\lambda_0 - \lambda_\star)\,\exp\!\Big(-\tfrac{t}{2\lambda_\star^2} - \tfrac{\lambda_t - \lambda_0}{\lambda_\star}\Big).$$

The leading exponential gives rate $|\lambda_t - \lambda_\star| = O(e^{-t/(2\lambda_\star^2)})$; for C^* the slowest mode gives $\|C_t - C^*\|_2 = O(e^{-t/(2\lambda_{\max}(C^*)^2)})$. *Both rates κ-dependent*; the C-rate is even worse, scaling as $\lambda_{\max}^{-2}$.

(b) *(Bures–Wasserstein.)* Flow (11.12) becomes

$$\dot{m} = -(C^*)^{-1}(m - m^*), \qquad \dot{C} = 2I - C\,(C^*)^{-1} - (C^*)^{-1}\,C.$$

Mean equation is the same as Euclidean: rate $1/\lambda_{\max}(C^*)$.
Closed form for C when C_0, C^* are simultaneously diagonalizable (e.g. $C_0 = \lambda_0 I$). Then C_t and C^* commute throughout, and in the common eigenbasis with C^* having eigenvalues λ_i, the eigenvalues $c_i(t)$ of C_t satisfy $\dot{c}_i = 2 - 2c_i/\lambda_i$, giving

$$c_i(t) \;=\; \lambda_i + (\lambda_0 - \lambda_i)\,e^{-2t/\lambda_i}. \tag{11.14}$$

Slowest mode: rate $2/\lambda_{\max}(C^*)$. In the general (non-commuting) case the same rate follows from linearization around C^*. BW improves the C-rate over Euclidean by a factor of $\lambda_{\max}(C^*)$ but both flows remain κ-dependent.

(c) *(Fisher–Rao: fully explicit.)* Flow (11.13) becomes

$$\dot{m} = -C\,(C^*)^{-1}(m - m^*), \qquad \dot{C} = C - C\,(C^*)^{-1}\,C.$$

This flow admits a complete closed-form solution. Let $D := C^{-1}$; differentiating $DC = I$ gives $\dot{D} = -C^{-1}\dot{C}\,C^{-1}$. Substitute the C-equation:

$$\dot{D} = -C^{-1}\big(C - C(C^*)^{-1}C\big)C^{-1} = -D + (C^*)^{-1}.$$

A *linear* ODE in D. Solving:

$$C_t^{-1} \;=\; e^{-t}\,C_0^{-1} \;+\; (1 - e^{-t})\,(C^*)^{-1}, \tag{11.15}$$

i.e. the precision interpolates linearly in e^{-t} between C_0^{-1} and $(C^*)^{-1}$. In the eigenbasis ($C_0 = \lambda_0 I$) this is the rational function [19, (E.22)]

$$\lambda_t \;=\; \frac{\lambda_\star}{1 + (\lambda_\star/\lambda_0 - 1)\,e^{-t}}, \qquad |\lambda_t - \lambda_\star| = O(e^{-t}).$$

Rate e^{-t}, *independent of* C^*. For the mean, $C_t \to C^*$ gives $\dot{m} \to -(m - m^*)$, hence $\|m_t - m^*\| \sim e^{-t}$ as well. Both quantities decay at the same κ-independent rate.

(d) *(Summary.)*

Flow	rate for $\|m - m^*\|$	rate for $\|C - C^*\|$
Euclidean	$1/\lambda_{\max}(C^*)$	$1/(2\lambda_{\max}(C^*)^2)$
Bures–Wasserstein	$1/\lambda_{\max}(C^*)$	$2/\lambda_{\max}(C^*)$
Fisher–Rao	1	1

The Fisher–Rao rate is κ-independent because the Fisher–Rao metric on $\mathcal{P}(\mathbb{R}^d)$ is diffeomorphism invariant: the affine whitening $x \mapsto (C^*)^{-1/2}(x - m^*)$ reduces the target to $\mathcal{N}(0, I)$, where the rate is trivially 1, and invariance carries this back to the original coordinates.

2. (Convexity: $A = C^{1/2}$ vs C coordinates.) For log-concave targets $\pi \propto e^{-V}$ with V convex, one might hope the Gaussian-VI objective

$$\mathcal{F}(m, C) = \mathbb{E}_{\mathcal{N}(m,C)}[V] - \tfrac{1}{2}\log\det C + \text{const}$$

is convex in (m, C). It is not. The right convex coordinate is the square root $A = C^{1/2}$.

(a) *(1D: convexity in (m, σ).)* Let $d = 1$ and V be convex. Use the reparametrization $X = m + \sigma Z$ with $Z \sim \mathcal{N}(0, 1)$: $\mathbb{E}[V(X)] = \mathbb{E}_Z[V(m + \sigma Z)]$. For each fixed Z, $(m, \sigma) \mapsto m + \sigma Z$ is affine, so $(m, \sigma) \mapsto V(m + \sigma Z)$ is convex. Expectation of convex is convex. The entropy $-\log\sigma$ is convex in $\sigma > 0$. So $\mathcal{F}$ is jointly convex on $\mathbb{R} \times \mathbb{R}_{>0}$.

(b) *(1D: the C coordinate fails.)* Let $u(C) := \mathbb{E}_{\mathcal{N}(0,C)}[V(X)]$. By the Gaussian heat equation, $\partial_C u = \frac{1}{2}\mathbb{E}[V''(X)]$. Applying this identity twice gives

$$\partial_C^2 u = \tfrac{1}{4}\,\mathbb{E}_{\mathcal{N}(0,C)}[V''''(X)].$$

So u is convex in C iff $\mathbb{E}[V''''] \geq 0$ — a condition on the *fourth* derivative. *Counterexample.* $V(x) = \frac{1}{2}x^2 + \epsilon\cos x$ is strongly convex for $\epsilon < 1$, but $V''''(x) = \epsilon\cos x$ takes negative values, so $u(C)$ is not convex.

(c) *(Multi-d.)* Let V be convex on $\mathbb{R}^d$ and $A = \Sigma^{1/2} \in \mathcal{S}^d_{++}$, so $X = m + AZ$ with $Z \sim \mathcal{N}(0, I_d)$ has law $\mathcal{N}(m, \Sigma)$. Repeating (a): $(m, A) \mapsto V(m + AZ)$ is convex for each Z (convex composed with affine), so $\mathbb{E}_Z[V(m + AZ)]$ is jointly convex in (m, A). The entropy $-\frac{1}{2}\log\det\Sigma = -\log\det A$ is convex on the PSD cone. Hence *for log-concave π, $\mathcal{F}$ is jointly convex in (m, A) where $A = \Sigma^{1/2}$.*
By contrast, $\Sigma \mapsto \Sigma^{1/2}$ is concave, so convexity in (m, A) does *not* transfer to (m, Σ).

(d) *(BW interpretation.)* The Bures–Wasserstein distance on Gaussians is

$$W_2^2\big(\mathcal{N}(m_0, \Sigma_0), \mathcal{N}(m_1, \Sigma_1)\big) = \|m_0 - m_1\|^2 + \inf_{O \in O(d)} \|\Sigma_0^{1/2} - \Sigma_1^{1/2} O\|_F^2.$$

Up to orthogonal ambiguity, this is the Euclidean metric on (m, A) with $A = \Sigma^{1/2}$. Combined with (c): *for log-concave π, $\mathcal{F}$ is geodesically convex on the Bures–Wasserstein manifold of Gaussians* [44]. This is the geometric statement behind convergence of the BW gradient flow on log-concave targets.

3. (Discretization on the Euclidean manifold fails; proximal gradient fixes it.) The parameter flows of Section 11.5 are well-behaved in continuous time but their discretization requires care. This exercise shows that explicit Euler on the Euclidean parameter flow (11.9) is unstable, and that the natural fix is to treat the non-smooth entropy via a *proximal* step. The Bures–Wasserstein geometry is what then makes that prox step closed-form.

(a) *(Euclidean entropy Hessian blows up.)* Consider the entropy $E(C) := -\frac{1}{2}\log\det C$ on $\mathcal{S}^d_{++}$, viewed as a function on the *flat* parameter manifold (Euclidean metric on C). Show that the Euclidean gradient and Hessian are

$$\nabla_C E(C) = -\tfrac{1}{2}C^{-1}, \qquad \nabla_C^2 E(C) : H \mapsto \tfrac{1}{2}\,C^{-1} H\, C^{-1}.$$

Conclude $\|\nabla^2 E(C)\|_{\mathrm{op}} = \frac{1}{2}/\lambda_{\min}(C)^2$, which blows up as $C \to \partial\mathcal{S}^d_{++}$.

(b) *(Explicit Euler on (11.9) fails.)* The unbounded Hessian means no fixed step size $h > 0$ satisfies $h \cdot \mathrm{Lip}(\nabla\mathcal{F}) < 1$ uniformly on the trajectory. An overshoot in the update $C_{k+1} = C_k - h\nabla_C\mathcal{F}(C_k)$ can produce a non-positive-definite C_{k+1} in a single step. *This is the standard failure mode of forward-Euler on objectives with a non-smooth (here: barrier-like) term.*

(c) *(Proximal gradient: split smooth + non-smooth.)* The standard remedy in non-smooth optimization is the *proximal gradient* (forward-backward) method: split $\mathcal{F} = \mathbb{E}[V] + E$ into a Lipschitz-smooth part $\mathbb{E}[V]$ (when V has bounded Hessian) and a non-smooth part E, then take a forward step on the smooth part and a *proximal* (implicit) step on the non-smooth part. The prox step

$$q_{k+1} \;=\; \operatorname*{arg\,min}_{q \in \mathcal{P}_{\mathrm{G}}} \Big(E(q) + \tfrac{1}{2h}\, d(q, q_{k+1/2})^2 \Big)$$

treats the entropy implicitly, which is what restores stability: the implicit step "knows about" the barrier at $\partial \mathcal{S}^d_{++}$ that forward-Euler ignored.

Choosing the distance d is a separate geometric choice. The natural choice on the Gaussian family is the Bures–Wasserstein distance $d = W_2$ (Exercise 2(d) showed the BW geometry is the right convex setting for $\mathcal{F}$); with this choice the prox of E is closed-form, as we now derive. The resulting algorithm is the *Forward-Backward Gaussian VI* (FB-GVI) algorithm of [26]. The same proximal principle at the density level on $\mathcal{P}(\mathbb{R}^d)$ is the *JKO scheme* for the Wasserstein gradient flow of KL.

(d) *(Derive the BW prox closed form.)* We derive the explicit closed-form expression for the BW prox step, in the simultaneously-diagonalizable setting where $C_{k+1/2}$ and C_{k+1} commute. Write $\tilde{q} := q_{k+1/2} = \mathcal{N}(\tilde{m}, \tilde{C})$ and let $q := q_{k+1} = \mathcal{N}(m, C)$ be the variable.

(i) Use the Bures–Wasserstein distance between commuting Gaussians, $W_2^2(q, \tilde{q}) = \|m - \tilde{m}\|^2 + \|C^{1/2} - \tilde{C}^{1/2}\|_F^2$, to write the prox objective

$$J(m, C) = -\tfrac{1}{2} \log\det C + \tfrac{1}{2h}\|m - \tilde{m}\|^2 + \tfrac{1}{2h}\|C^{1/2} - \tilde{C}^{1/2}\|_F^2.$$

(ii) *Mean.* Show $\nabla_m J = 0$ gives $m_{k+1} = \tilde{m}$. The mean is unchanged by the prox step — intuitive, since the entropy depends only on C.

(iii) *Covariance.* Set $A := C^{1/2}$, $\tilde{A} := \tilde{C}^{1/2}$, both symmetric PSD and commuting. Rewrite $-\frac{1}{2}\log\det C = -\log\det A$ and show that $\nabla_A J = 0$ gives the *matrix quadratic*

$$A^2 - \tilde{A}\, A - hI = 0.$$

Solve for the PSD root (using the commuting eigenbasis and the scalar quadratic $a^2 - \tilde{a}\, a - h = 0$):

$$C_{k+1}^{1/2} \;=\; \tfrac{1}{2}\tilde{C}^{1/2} \;+\; \tfrac{1}{2}\big(\tilde{C} + 4hI\big)^{1/2}. \tag{11.16}$$

(iv) *Positive-definiteness is preserved.* The discriminant $\tilde{C} + 4hI$ is strictly positive-definite for any $h > 0$, hence $(\tilde{C} + 4hI)^{1/2}$ is well-defined and PSD, and $C_{k+1}^{1/2} \succ 0$ regardless of h. Squaring gives $C_{k+1} \succ 0$. *This is the stability gain over (b): the proximal step automatically respects the PSD cone, regardless of step size.*

12 Metropolis Correction

So far every algorithm in this note has been a discretization of a continuous-time flow, with a residual bias controlled by the step size. Sampling has a tool with no analogue in optimization that removes this bias *exactly*, for any step size: the Metropolis–Hastings accept/reject step. We develop it in three stages: (i) reversibility and stationarity as measure-theoretic notions; (ii) the classical Metropolis–Hastings ratio for proposals that admit densities, covering random-walk Metropolis and MALA; (iii) an extension to proposals built from *deterministic involutions on an extended state space*, which is the right framework for proposals like Goodman–Weare's stretch move and HMC, where the proposal is not a density on $\mathbb{R}^d$.

12.1 Reversibility and Stationarity

Let $P(x, dy)$ be a Markov transition kernel on a measurable space $(\mathbb{R}^d, \mathcal{B})$: $P(x, \cdot)$ is a probability measure for each x, and $x \mapsto P(x, A)$ is measurable for each $A \in \mathcal{B}$. Recall a probability measure π is *stationary* (or invariant) for P if

$$\int \pi(dx)\, P(x, A) = \pi(A) \quad \text{for every } A \in \mathcal{B}. \tag{12.1}$$

A sufficient (but not necessary) condition for stationarity is *reversibility*: π is reversible for P if the joint law of (X, Y) with $X \sim \pi$ and $Y \mid X \sim P(X, \cdot)$ is symmetric in (X, Y). As a measure identity on $\mathbb{R}^d \times \mathbb{R}^d$,

$$\pi(dx)\, P(x, dy) \;=\; \pi(dy)\, P(y, dx). \tag{12.2}$$

We refer to (12.2) as the *detailed balance* condition. Note carefully: the identity is between two measures on $\mathbb{R}^d \times \mathbb{R}^d$, not between two densities. This distinction will matter shortly, when we work with proposals that have Dirac-mass components and no joint density.

Proposition 12.1 (Detailed balance implies stationarity)**.** *If* (12.2) *holds, then* π *is stationary for* P*.*

Proof. Integrate (12.2) over x:

$$\int \pi(dx)\, P(x, A) \overset{(12.2)}{=} \int \pi(dy)\, \mathbf{1}_A(y) \int P(y, dx) \;=\; \int_A \pi(dy) \;=\; \pi(A).$$

□

Detailed balance is strictly stronger than stationarity (one can construct non-reversible chains with π stationary, e.g. underdamped Langevin) but it is the most convenient sufficient condition because it can be checked locally, kernel by kernel, without computing the integral in (12.1). The Metropolis–Hastings algorithm is precisely the recipe for *constructing* a kernel P from an arbitrary proposal so that (12.2) holds.

12.2 Metropolis–Hastings for Proposals with Densities

The setup. Let $Q(x, dy)$ be a *proposal kernel*: an arbitrary Markov kernel on $\mathbb{R}^d$ that we know how to simulate from. We want to modify Q so that the resulting chain is reversible with respect to a target $\pi(dx) = \pi(x)\,dx$ known up to a normalizing constant.

The Metropolis–Hastings algorithm at state X_k is:

(1) Propose $Y \sim Q(X_k, \cdot)$.

(2) Compute an acceptance probability $\alpha(X_k, Y) \in [0, 1]$ (to be specified).

(3) With probability $\alpha(X_k, Y)$, set $X_{k+1} = Y$; otherwise set $X_{k+1} = X_k$.

The resulting transition kernel is

$$P(x, dy) = \alpha(x, y)\, Q(x, dy) + \delta_x(dy)\,\big(1 - \textstyle\int \alpha(x, y')\, Q(x, dy')\big). \tag{12.3}$$

The $\delta_x(dy)$ component represents the rejection event: with positive probability the chain stays at x. Note that $P(x, dy)$ is generally *not* absolutely continuous in dy even when $Q(x, dy)$ is, precisely because of this Dirac mass. This is the reason (12.2) must be interpreted as a measure identity.

When the proposal has a density. Assume now that $Q(x, dy) = q(y \mid x)\,dy$ for some conditional density $q(\cdot \mid x)$. Write out (12.2) as a relation between joint densities (where they exist) plus diagonal mass:

$$\pi(dx)\,P(x, dy)\big|_{\text{off-diagonal}} = \alpha(x, y)\,\pi(x)\,q(y \mid x)\,dx\,dy,$$

and similarly with $x \leftrightarrow y$. The diagonal contributions $\pi(dx)\,\delta_x(dy)\cdot(\ldots)$ are symmetric under swapping $x \leftrightarrow y$ automatically. So detailed balance reduces to

$$\alpha(x, y)\,\pi(x)\,q(y \mid x) \;=\; \alpha(y, x)\,\pi(y)\,q(x \mid y) \qquad \text{for almost every } (x, y). \tag{12.4}$$

The largest acceptance probability $\alpha \in [0, 1]$ satisfying (12.4) is the classical *Metropolis–Hastings ratio*:

$$\alpha(x, y) = \min\left(1,\ \frac{\pi(y)\,q(x \mid y)}{\pi(x)\,q(y \mid x)}\right). \tag{12.5}$$

To verify: when the ratio is ≤ 1, $\alpha(x, y)$ equals the ratio and $\alpha(y, x) = 1$, making (12.4) read $\pi(y)q(x \mid y) \cdot 1 = 1 \cdot \pi(y)q(x \mid y)$. The other case is symmetric. So P defined by (12.3) with α from (12.5) is reversible for π, hence π-stationary.

Two practical features make (12.5) usable: α depends on π only through the ratio $\pi(y)/\pi(x)$, which cancels the normalizing constant $Z = \int e^{-V}$; and $\pi(y)/\pi(x) = e^{V(x)-V(y)}$ needs only function evaluations of V.

Why $\min(1, r)$ is the optimal choice. The detailed-balance condition (12.4) has many solutions. For any function $s : \mathbb{R}^d \times \mathbb{R}^d \to (0, \infty)$ satisfying $s(x, y) = s(y, x)$ (symmetry), the acceptance

$$\tilde{\alpha}(x, y) = \frac{s(x, y)}{\pi(x)\, q(y \mid x)}$$

satisfies (12.4), provided we clip to $[0, 1]$, i.e. require $s(x, y) \le \min(\pi(x)q(y \mid x),\ \pi(y)q(x \mid y))$. The largest admissible s is the pointwise minimum $s^*(x, y) = \min(\pi(x)q(y \mid x),\ \pi(y)q(x \mid y))$, which recovers exactly $\tilde{\alpha} = \min(1,\ \pi(y)q(x \mid y)/[\pi(x)q(y \mid x)])$ — the Metropolis–Hastings rate (12.5). Any other valid choice $\tilde{\alpha} \le \alpha_{\mathrm{MH}}$ pointwise.

Thus among all reversible kernels of the form (12.3) built from a given proposal q, the Metropolis–Hastings choice (12.5) dominates all others. The same statement holds in the involution framework below, with $\min(1, r)$ replaced by the Jacobian-corrected version (12.8); the argument is identical.

Random-walk Metropolis. The simplest choice is a symmetric Gaussian proposal $q(\cdot \mid x) = \mathcal{N}(x, h^2 I_d)$. Symmetry $q(y \mid x) = q(x \mid y)$ collapses (12.5) to

$$\alpha(x, y) = \min\bigl(1,\ e^{V(x) - V(y)}\bigr). \tag{12.6}$$

This is *random-walk Metropolis* (RWM): gradient-free and applicable to any density known up to a constant. The step h is a trade-off knob: larger h gives bigger moves but more rejections, smaller h gives high acceptance but small moves. We work out the optimal trade-off in Section 12.4 below; the headline result is that the optimal step on a d-dimensional product target scales as $h \asymp d^{-1/2}$ with acceptance rate ≈ 0.234 [32].

Composition with Langevin: MALA. Replacing the symmetric proposal by the ULA-step proposal $q(\cdot \mid x) = \mathcal{N}(x - h\nabla V(x),\ 2hI_d)$ gives the *Metropolis-adjusted Langevin algorithm* (MALA) [60, 59]. The proposal density is no longer symmetric — the gradient drift in $q(y \mid x)$ and $q(x \mid y)$ differs — so the full ratio (12.5) is needed. MALA inherits ULA's gradient information while removing its bias: π is targeted exactly. The gradient information also gives a better dimensional scaling: $h \asymp d^{-1/3}$ with optimal acceptance ≈ 0.574 [58] (Section 12.4).

12.3 Beyond Density Proposals: Deterministic Involutions

The density-form derivation above assumed $q(y \mid x)$ exists. For some of the most useful proposals — including the Goodman–Weare stretch move and HMC — the proposed Y is a *deterministic function* of the current state and an auxiliary random variable, so $Q(x, \cdot)$ concentrates on a lower-dimensional set and admits no density on $\mathbb{R}^d$. The Metropolis–Hastings ratio (12.5) applied naively to such a proposal yields a ratio of singular measures, typically zero or infinity, and is wrong.

The fix has been understood since the foundational work of Green on reversible-jump MCMC [35]: introduce the auxiliary randomness explicitly, lift the proposal to a *deterministic involution* on an extended state space, and recover detailed balance there. The

acceptance ratio acquires a Jacobian factor accounting for the change of variables. [70] gave the general-state-space measure-theoretic treatment of detailed balance for possibly singular MH kernels, but did not phrase the result in terms of involutions. For the involution structure narrative, we follow the presentation of [6, 33].

Extended state space

Augment the state $X \in \mathbb{R}^d$ with an auxiliary variable $Z \in \mathcal{Z}$ on a measurable space carrying a reference measure $\nu(dz)$. The extended state space is $\mathbb{R}^d \times \mathcal{Z}$ and the extended target is the product measure

$$\tilde{\pi}(dx, dz) := \pi(dx)\, \nu(dz). \tag{12.7}$$

Sampling from $\tilde{\pi}$ and projecting onto the first coordinate samples π, since the X-marginal of $\tilde{\pi}$ is exactly π. The auxiliary Z encodes the randomness of the proposal — e.g. the choice of partner walker and stretch factor in Goodman–Weare, or the initial momentum in HMC.

Involutions

A *proposal* is now a measurable map $\phi : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}^d \times \mathcal{Z}$ that is an *involution*: $\phi \circ \phi = \mathrm{id}$. Concretely, the proposal sends (X_k, Z_k) to $\phi(X_k, Z_k)$, and applying ϕ a second time returns to the starting state.

Remark 12.2 (Why involutions). The involution requirement $\phi^2 = \mathrm{id}$ replaces the forward/reverse density structure of the previous subsection. Density-based MH involves *both* $q(y \mid x)$ and $q(x \mid y)$ separately. For deterministic proposals there is no $q(x \mid y)$ to write down — but if ϕ is an involution, the reverse move from the proposed state is just ϕ again, automatically. Algebraically, $\phi^2 = \mathrm{id}$ is what makes the swap $x \leftrightarrow y$ in (12.2) a well-defined symmetry operation on the extended space.

Acceptance ratio for involutions

Assume ϕ is a C^1 diffeomorphism on $\mathbb{R}^d \times \mathcal{Z}$ satisfying $\phi^2 = \mathrm{id}$, and that the extended target has a density against $dx\, \nu(dz)$: $\tilde{\pi}(dx, dz) = \tilde{\pi}(x, z)\, dx\, \nu(dz)$ where, by (12.7), the density factorizes as $\tilde{\pi}(x, z) = \pi(x)\, n(z)$ in the case $\nu(dz) = n(z)\, dz$. The argument below extends with no change to general ν via Radon–Nikodym, with $\tilde{\pi}(x, z)$ interpreted as the density of $\tilde{\pi}$ against $dx \otimes \nu$.

The Metropolis–Hastings step on the extended space is:

(1) Draw $Z_k \sim \nu$ (independent of X_k).

(2) Propose $(X', Z') := \phi(X_k, Z_k)$.

(3) Accept with probability

$$\alpha(X_k, Z_k) = \min\left(1,\ \frac{\tilde{\pi}(\phi(X_k, Z_k))}{\tilde{\pi}(X_k, Z_k)}\, \left|\det D\phi(X_k, Z_k)\right|\right); \tag{12.8}$$

if accepted, set $X_{k+1} = X'$; otherwise $X_{k+1} = X_k$.

The form (12.8) has two new ingredients compared with the classical formula (12.5): the auxiliary variable Z has been promoted into the state, and a *Jacobian factor* $|\det D\phi|$ has appeared. The Jacobian accounts for the change-of-variables in ϕ: when ϕ stretches a small volume in (x, z)-space, the proposed configuration covers proportionally more or less measure, and the acceptance probability compensates. In the special case $|\det D\phi| = 1$ (volume-preserving ϕ) the Jacobian drops out, and the ratio reduces to a pure density ratio in the extended target.

Proposition 12.3 (Detailed balance for involution proposals)**.** *Let ϕ be a C^1 involution on $\mathbb{R}^d \times \mathcal{Z}$ and let α be defined by (12.8). The Markov kernel (12.3) built from ϕ on the extended state is reversible for $\tilde{\pi}$: for any measurable $E \subset (\mathbb{R}^d \times \mathcal{Z})^2$,*

$$\int_E \tilde{\pi}(d\xi)\, \tilde{P}(\xi, d\xi') \;=\; \int_{E^*} \tilde{\pi}(d\xi)\, \tilde{P}(\xi, d\xi'), \tag{12.9}$$

where $E^* = \{(\xi', \xi) : (\xi, \xi') \in E\}$ is the swap of E. Projecting onto the x-coordinate yields a chain with π as stationary distribution.

Sketch. Write the extended kernel as in (12.3):

$$\tilde{P}(\xi, d\xi') = \alpha(\xi)\, \delta_{\phi(\xi)}(d\xi') \;+\; \big(1 - \alpha(\xi)\big)\, \delta_{\xi}(d\xi'),$$

i.e. with probability $\alpha(\xi)$ the chain moves to $\phi(\xi)$, otherwise it stays at ξ. The diagonal (rejection) part $(1 - \alpha(\xi))\, \delta_\xi(d\xi')$ contributes to (12.9) symmetrically since $\xi = \xi'$ on its support.

The off-diagonal (acceptance) part requires the change of variables. Writing $\xi = (x, z)$ and using the density form $\tilde{\pi}(d\xi) = \tilde{\pi}(\xi)\, d\xi$, the left-hand side of (12.9) on the acceptance event is

$$\int \mathbf{1}_E(\xi, \phi(\xi))\, \tilde{\pi}(\xi)\, \alpha(\xi)\, d\xi.$$

Change variables $\eta := \phi(\xi)$ so $\xi = \phi(\eta)$ (using $\phi^2 = \mathrm{id}$) and $d\xi = |\det D\phi(\eta)|\, d\eta$:

$$= \int \mathbf{1}_E(\phi(\eta), \eta)\, \tilde{\pi}(\phi(\eta))\, \alpha(\phi(\eta))\, |\det D\phi(\eta)|\, d\eta.$$

The indicator $\mathbf{1}_E(\phi(\eta), \eta)$ equals $\mathbf{1}_{E^*}(\eta, \phi(\eta))$ by the definition of E^*. The integrand on the right-hand side of (12.9) (over E^*) is $\tilde{\pi}(\eta)\, \alpha(\eta)$. So (12.9) reduces to the pointwise identity

$$\tilde{\pi}(\phi(\eta))\, \alpha(\phi(\eta))\, |\det D\phi(\eta)| \;=\; \tilde{\pi}(\eta)\, \alpha(\eta) \qquad \text{for a.e. } \eta.$$

Plug in (12.8): both sides equal $\min\big(\tilde{\pi}(\eta),\ \tilde{\pi}(\phi(\eta))\, |\det D\phi(\eta)|\big)$, which is symmetric in η and $\phi(\eta)$ (using $\phi^2 = \mathrm{id}$ and $|\det D\phi(\eta)| \cdot |\det D\phi(\phi(\eta))| = 1$). Reversibility (12.9) follows. The marginal in x is π because $\tilde{\pi} = \pi \otimes \nu$ is a product measure, so π-stationarity of the x-chain follows from $\tilde{\pi}$-stationarity on the extended space. □

Remark 12.4 (Classical MH as a special case). The classical density-form algorithm of Equation (12.5) is itself an instance of (12.8) with extended state $(x, y) \in \mathbb{R}^{2d}$, reference $\nu(dy \mid x) = q(y \mid x)\, dy$, involution $\phi(x, y) = (y, x)$ (the swap), and Jacobian $|\det D\phi| = 1$. Plugging in, $\tilde{\pi}(y, x)/\tilde{\pi}(x, y) = \pi(y)q(x \mid y)/[\pi(x)q(y \mid x)]$, which is exactly the classical ratio. This is worked out in detail in Week 8 Exercise 1.

Worked example: Goodman–Weare stretch move

The *stretch move* of Goodman–Weare [34] is an example where the involution framework can be applied. Evolve L walkers $\mathbf{X} = (X^{(1)}, \dots, X^{(L)}) \in (\mathbb{R}^d)^L$ in parallel, targeting $\pi^{\otimes L}$ on the joint space. To update walker ℓ:

(1) Pick a partner $j \neq \ell$ uniformly at random.

(2) Draw a stretch factor $Z \sim p(z)$ with density $p(z) \propto 1/\sqrt{z}$ on $[1/a, a]$ for some $a > 1$ (typically $a = 2$).

(3) Propose the new walker $Y = X^{(j)} + Z\big(X^{(\ell)} - X^{(j)}\big)$.

(4) Accept with a probability to be determined.

The proposed Y lies on the line from $X^{(j)}$ through $X^{(\ell)}$, scaled by Z. There is no density for Y on $\mathbb{R}^d$: given $\mathbf{X}$, the proposal is supported on a 1D ray.

Casting as an involution. Treat $(\mathbf{X}, Z)$ as the extended state, with reference measure $\nu(dz) = p(z)\,dz$ on $[1/a, a]$. Define the proposal map

$$\phi(\mathbf{X}, Z) \;=\; (\mathbf{X}',\, 1/Z), \qquad X^{(k)\prime} = \begin{cases} X^{(j)} + Z\big(X^{(\ell)} - X^{(j)}\big) & k = \ell, \\ X^{(k)} & k \neq \ell. \end{cases} \tag{12.10}$$

We check that ϕ is an involution. The walker update inverts under $Z \to 1/Z$ because $X^{(j)} + Z\big(X^{(\ell)} - X^{(j)}\big) = X^{(\ell)\prime}$ implies $X^{(\ell)} = X^{(j)} + (1/Z)\big(X^{(\ell)\prime} - X^{(j)}\big)$, exactly the update produced by ϕ acting on $(\mathbf{X}', 1/Z)$. The partner $X^{(j)}$ and the other walkers are fixed throughout, so applying ϕ a second time recovers $(\mathbf{X}, Z)$.

The Jacobian. Compute $D\phi$ as a block matrix on coordinates $(X^{(\ell)}, Z)$, holding all other walkers fixed. From (12.10): $\partial X^{(\ell)\prime}/\partial X^{(\ell)} = Z\, I_d$, $\partial X^{(\ell)\prime}/\partial Z = X^{(\ell)} - X^{(j)}$, $\partial(1/Z)/\partial X^{(\ell)} = 0$, and $\partial(1/Z)/\partial Z = -1/Z^2$. The block structure is upper triangular in $(X^{(\ell)}, Z)$, so

$$\big|\det D\phi(\mathbf{X}, Z)\big| = \big|\det(Z\, I_d)\big| \cdot |-1/Z^2| = Z^d \cdot Z^{-2} = Z^{d-2}. \tag{12.11}$$

The acceptance ratio. The target on the extended space is $\tilde\pi(\mathbf{X}, Z) = \pi^{\otimes L}(\mathbf{X})\, p(Z)$. The non-updated walkers $X^{(k)}$ for $k \neq \ell$ cancel in the density ratio, leaving

$$\frac{\tilde\pi(\phi(\mathbf{X}, Z))}{\tilde\pi(\mathbf{X}, Z)} = \frac{\pi(X^{(\ell)\prime})}{\pi(X^{(\ell)})}\,\frac{p(1/Z)}{p(Z)}.$$

With $p(z) \propto z^{-1/2}$, $p(1/Z)/p(Z) = (1/Z)^{-1/2}/Z^{-1/2} = Z^{1/2}/Z^{-1/2} = Z$. Combining with the Jacobian (12.11) and (12.8):

$$\alpha(\mathbf{X}, Z) = \min\left(1,\; \frac{\pi(X^{(\ell)\prime})}{\pi(X^{(\ell)})} \;\cdot\; Z \;\cdot\; Z^{d-2}\right) = \min\left(1,\; Z^{d-1}\,\frac{\pi(X^{(\ell)\prime})}{\pi(X^{(\ell)})}\right). \tag{12.12}$$

This is the stretch-move acceptance ratio [34], now *derived* as a consequence of the involution framework.

Remark 12.5 (The role of the $1/\sqrt{z}$ density). The involution framework actually works for *any* choice of density $p(z)$ on $[1/a, a]$: a different choice just produces a different value of $p(1/Z)/p(Z)$ in the acceptance ratio. The specific choice $p(z) \propto z^{-1/2}$ has the property $p(1/Z)/p(Z) = Z$, which combines with the Jacobian Z^{d-2} to give the clean Z^{d-1} form in (12.12). The original [34] chose $p \propto z^{-1/2}$.

Affine invariance. The stretch move is affine invariant in the sense of Definition 10.2. Applying $A \in GL(d)$ to all walkers sends $X^{(j)} + Z(X^{(\ell)} - X^{(j)})$ to $AX^{(j)} + Z(AX^{(\ell)} - AX^{(j)}) = A[X^{(j)} + Z(X^{(\ell)} - X^{(j)})]$, so the proposal commutes with A. The acceptance ratio $\pi(X^{(\ell)\prime})/\pi(X^{(\ell)})$ pulls back through A to the analogous ratio for the pushforward target $A_{\#}\pi$, and the Jacobian Z^{d-1} is unchanged. So the stretch move on π in coordinates $\mathbf{X}$ is identical, step for step, to the stretch move on $A_{\#}\pi$ in coordinates $A\mathbf{X}$. Affine invariance is built in by the geometry of the proposal — there is no metric to estimate. The `emcee` package [29] is the standard implementation and dominates Bayesian inference in computational astrophysics.

HMC as another involution

Hamiltonian Monte Carlo [53] fits the same framework and we sketch it briefly. The auxiliary variable is a momentum $P \in \mathbb{R}^d$ with reference measure $\nu(dp) = \mathcal{N}(0, I_d)$, so the extended target is the Boltzmann distribution on phase space $\tilde{\pi}(x, p) \propto e^{-V(x) - \|p\|^2/2}$. The proposal is $\phi = F \circ \Phi_T$, where Φ_T is the (approximate) Hamiltonian flow on (x, p) for time T under $H(x, p) = V(x) + \frac{1}{2}\|p\|^2$ and $F(x, p) = (x, -p)$ is momentum negation. One checks $\phi^2 = \mathrm{id}$: $\phi^2 = F\Phi_T F\Phi_T = FF\Phi_{-T}\Phi_T = \mathrm{id}$, using time reversibility $F\Phi_T F = \Phi_{-T}$ of the Hamiltonian flow. When Φ_T is integrated exactly, H is conserved so $\tilde{\pi}$ is preserved exactly and the acceptance ratio (12.8) becomes 1; with leapfrog as the discrete approximation, H is conserved up to $O(h^2)$ per step and the ratio is the corresponding energy error $\alpha = \min(1, e^{-\Delta H})$ where ΔH is the leapfrog energy drift. The leapfrog Jacobian is exactly 1 because the scheme is symplectic, which is why $|\det D\phi| = 1$ does not appear in the standard HMC acceptance ratio.

12.4 Dimensional Scaling of Step Size

The Metropolis correction removes bias for any step size, but step size still controls efficiency through the trade-off between move length and acceptance rate. For the algorithms above, this trade-off admits a clean asymptotic answer in high dimensions: the optimal step size h shrinks as a negative power of the dimension d, with the exponent determined by how well the underlying continuous-time process approximates the truth. We work out the canonical isotropic Gaussian case $\pi = \mathcal{N}(0, I_d)$ explicitly — both because the calculation is short and because the scaling exponents are *universal*: they extend to broad classes of d-fold product targets under mild regularity [32, 58, 7].

Throughout, write the log-acceptance ratio as $\log r$ and let $X \sim \pi$. The chain has non-vanishing acceptance as $d \to \infty$ iff $\mathbb{E}_{X\sim\pi}[\log r]$ stays bounded.

Random-walk Metropolis: $h_\star \asymp d^{-1/2}$. On $\pi = \mathcal{N}(0, I_d)$, $V(x) = \frac{1}{2}\|x\|^2$ and the RWM proposal is $Y = X + hZ$ with $Z \sim \mathcal{N}(0, I_d)$ independent. The log-acceptance ratio is

$$\log r \;=\; V(X) - V(Y) \;=\; \tfrac{1}{2}\|X\|^2 - \tfrac{1}{2}\|X + hZ\|^2 \;=\; -h\, X \cdot Z \;-\; \tfrac{h^2}{2}\|Z\|^2.$$

At stationarity $X \sim \mathcal{N}(0, I_d)$ is independent of Z, so $X \cdot Z \sim \mathcal{N}(0, d)$ and $\mathbb{E}\|Z\|^2 = d$ with $\mathrm{Var}\|Z\|^2 = 2d$. To leading order in d,

$$\log r \;\overset{d}{\approx}\; \mathcal{N}\Big(-\tfrac{h^2 d}{2},\; h^2 d\Big).$$

(The mean is $-h^2 d/2$; the $-hX \cdot Z$ piece contributes the $h^2 d$ variance and the $\|Z\|^2$ fluctuations contribute only $O(h^4 d)$.) For $\log r$ to remain $O(1)$ as $d \to \infty$,

$$h^2\, d = O(1) \quad \Longleftrightarrow \quad h_\star \asymp d^{-1/2}. \tag{12.13}$$

Setting $h = \ell\, d^{-1/2}$ for a constant ℓ gives $\log r \to \mathcal{N}(-\ell^2/2,\, \ell^2)$, and the limiting expected acceptance $\mathbb{E}[\min(1, e^{\log r})] = 2\Phi(-\ell/2)$ is maximized at $\ell^* \approx 2.38$, giving the famous *optimal RWM acceptance rate* $\alpha^* \approx 0.234$ [32]. Traversing an $O(1)$ distance requires $O(d)$ iterations.

MALA: $h_\star \asymp d^{-1/3}$. The MALA proposal $Y = X - h\nabla V(X) + \sqrt{2h}\, Z = (1-h)X + \sqrt{2h}\, Z$ incorporates a leading-order "correction" from the gradient drift that cancels exactly the $O(h^2 d)$ bias of RWM. Direct computation of the log-acceptance ratio

$$\log r \;=\; V(X) - V(Y) - \tfrac{1}{4h}\big(\|X - Y + h\nabla V(Y)\|^2 - \|Y - X + h\nabla V(X)\|^2\big)$$

(the standard MH ratio for an asymmetric proposal with $\nabla V(x) = x$ on the isotropic Gaussian) reduces, after expanding $\|Y - X + h\nabla V(X)\|^2 = \|hX - hX + \sqrt{2h}Z\|^2$· careful algebra, to a quantity whose mean at stationarity is $O(h^3 d)$ rather than $O(h^2 d)$ [58]. Asking $h^3 d = O(1)$ gives

$$h_\star \asymp d^{-1/3}, \tag{12.14}$$

with optimal acceptance rate $\alpha^* \approx 0.574$ [58]. Traversing an $O(1)$ distance requires $O(d^{1/3})$ iterations. MALA improves on RWM by a factor of $d^{2/3}$ in iteration count — the reward of using gradient information.

HMC: $h_\star \asymp d^{-1/4}$. For HMC with leapfrog of $L = T/h$ steps and total integration time $T = O(1)$, the energy drift along a trajectory satisfies $\Delta H = O(h^2)$ per coordinate (leapfrog is a second-order symplectic integrator). On $\pi = \mathcal{N}(0, I_d)$ the per-step energy errors of independent coordinates accumulate independently; across all d coordinates, $\Delta H \sim \mathcal{N}(ch^4 d, \sigma^2 h^4 d)$ for some constants c, σ [7]. The acceptance ratio $\min(1, e^{-\Delta H})$ stays bounded away from 0 iff

$$h^4\, d = O(1) \quad \Longleftrightarrow \quad h_\star \asymp d^{-1/4}, \tag{12.15}$$

with optimal acceptance rate $\alpha^* \approx 0.651$ [7]. Each iteration costs $T/h = O(d^{1/4})$ gradient evaluations and moves a physical time $T \asymp 1$, so HMC needs $O(d^{1/4})$ gradient evaluations per effectively independent sample.

Summary.

	Optimal step $h_\star$	Optimal acceptance	Gradient evals per sample
RWM	$d^{-1/2}$	≈ 0.234	d
MALA	$d^{-1/3}$	≈ 0.574	$d^{1/3}$
HMC	$d^{-1/4}$	≈ 0.651	$d^{1/4}$

The exponents track the discretization order: RWM uses no information beyond function values (order 1/2); MALA uses the gradient (a 1st-order discretization of Langevin, contributing one extra half-power); HMC uses the symplectic leapfrog (a 2nd-order discretization of Hamiltonian flow, contributing yet another half-power). Higher-order integrators applied to HMC can push the exponent further, with $h_\star \asymp d^{-1/(2k)}$ for an order-k symplectic integrator [7].

What the scaling does not capture. The bounds above measure the cost *per effectively independent sample* on a product-form Gaussian target, where the dimensional challenge is uniform across coordinates. Two situations break the analysis. (i) Ill-conditioned targets: condition number κ enters multiplicatively, as in Section 10.2; the $d^{-1/4}$ HMC step is on the slowest direction, so the slow direction's step shrinks to $\sqrt{\lambda_{\min}}/d^{1/4}$ — preconditioning can recover this loss. (ii) the per-coordinate independence assumption fails, and the scaling can degrade.

Extensions: multiple-try, delayed rejection, NUTS. The involution framework can be used to calculate the acceptance rates for several refinements of Metropolis–Hastings widely used in practice. *Multiple-try Metropolis* [50] generates K candidate proposals at once and accepts one with weights involving all candidates, with the joint $(x, y_1, \ldots, y_K, k, x_1, \ldots, x_{K-1})$ state made into an involution on an extended product space. *Delayed-rejection* [71, 36] reuses the rejected proposal information by composing several MH attempts conditionally, satisfying a *conditional* detailed balance. *Biased progressive sampling* [8], the engine of the No-U-Turn Sampler (NUTS) [39], combines multinomial selection along a trajectory with a sequence of conditional accept/reject decisions. In all three cases the right framework is deterministic involutions on an appropriately extended space, often with the detailed-balance condition imposed conditionally on a selected event. We treat these as exercises (Week 8 Exercise 3).

For adaptive HMC specifically, the *Gibbs self-tuning* (GIST) framework of [11] unifies randomized HMC, multinomial HMC, and NUTS as a single class: treat each algorithm's tuning parameters (path length, step size, mass matrix) as auxiliary variables, augment the state to (x, p, α), and use a measure-preserving involution on the enlarged space. The framework also enables local step-size adaptation in NUTS while preserving reversibility [10, 9].

Week 8 Exercises

Exercise 1 shows that classical Metropolis–Hastings is itself an instance of the involution framework of Section 12.3; the remaining (optional) exercises extend the framework to multiple-try, delayed rejection, and NUTS, each requiring a different extended state and—in some cases—only a *conditional* form of detailed balance.

1. (Standard Metropolis–Hastings as an involution.) We show the density-based algorithm of (12.5) is the same as the involution formalism applied to a particular extended state and reference.

Take $\xi = (x, y) \in \mathbb{R}^{2d}$ with reference kernel $\nu(dy \mid x) = q(y \mid x)\, dy$ and extended target

$$\tilde{\pi}(dx, dy) := \pi(dx)\, q(y \mid x)\, dy. \tag{12.16}$$

The only minor extension from Section 12.3 is that the reference depends on x (a kernel rather than a fixed measure).

(a) *(The swap is an involution.)* Verify that

$$\phi(x, y) := (y, x) \tag{12.17}$$

satisfies $\phi^2 = \mathrm{id}$ and $|\det D\phi| = 1$.

(b) *(Marginal recovers π.)* Show $\int \tilde{\pi}(dx, dy) = \pi(dx)$. After the swap, the old y-coordinate becomes the new x-coordinate.

(c) *(Recover the MH acceptance.)* Plug (12.17) into (12.8) and use (12.16) to obtain

$$\alpha(x, y) = \min\left(1,\, \frac{\pi(y)\, q(x \mid y)}{\pi(x)\, q(y \mid x)}\right), \tag{12.18}$$

which is exactly (12.5). Classical MH is the simplest non-trivial involution: "propose, then swap".

(d) *(Interpretation.)* The structure of standard MH is encoded in three choices: the extended state $(x, y) \in \mathbb{R}^{2d}$, the reference $\nu(dy \mid x) = q(y \mid x)\, dy$, and the swap $\phi(x, y) = (y, x)$. Random-walk Metropolis, MALA, and pCN are all this same involution; only q differs.

Takeaway. To design a new MCMC method, pick an extended state, reference measure, and involution; the acceptance ratio adapts automatically via (12.8).

2. *(Optional)* **(Multiple-try Metropolis.)** The MTM algorithm of [50] generates K candidates per step, selects one by a weight scheme, and accepts via a Metropolis correction. We derive it as an involution on a further-extended space.

Setup. Fix a proposal $q(\cdot \mid x)$ and a positive symmetric weight $w(x, y) = w(y, x)$ (in practice $w(x, y) = \pi(x)\, q(y \mid x)$). At $X_k = x$:

(1) Sample $y_1, \dots, y_K \overset{\text{iid}}{\sim} q(\cdot \mid x)$.

(2) Pick J with $P(J = j) \propto w(x, y_j)$.

(3) Sample $x_1, \dots, x_{K-1} \overset{\text{iid}}{\sim} q(\cdot \mid y_J)$; set $x_K := x$.

(4) Accept $X_{k+1} = y_J$ with probability below.

The extended state is $\xi = (x, y_1, \dots, y_K, J, x_1, \dots, x_{K-1})$ with target

$$\tilde{\pi}(\xi) = \pi(x) \prod_{k=1}^{K} q(y_k \mid x) \, \frac{w(x, y_J)}{\sum_k w(x, y_k)} \prod_{k=1}^{K-1} q(x_k \mid y_J). \tag{12.19}$$

(a) *(Marginal is π.)* Verify that the x-marginal of (12.19) is $\pi(x)$. The selection weights are what make the J-sum collapse to one.

(b) *(The MTM involution.)* The involution promotes y_J to the new x and inserts the old x at position J of the new y-block:

$$\begin{aligned} \phi : (x,\ y_1, \dots, y_K,\ J,\ x_1, \dots, x_{K-1}) \\ \mapsto \big(y_J,\ (x_1, \dots, x_{J-1},\ x,\ x_{J+1}, \dots, x_{K-1},\ y_J), \\ J,\ (y_1, \dots, y_{J-1},\ y_{J+1}, \dots, y_K)\big). \end{aligned} \tag{12.20}$$

Verify $\phi^2 = \mathrm{id}$. The Jacobian is $|\det D\phi| = 1$ (a coordinate permutation).

(c) *(Acceptance ratio.)* Compute $\tilde{\pi}(\phi(\xi))/\tilde{\pi}(\xi)$ using (12.19). With $w(x, y) = \pi(x)q(y \mid x)$ the products of q cancel; the surviving terms give

$$\alpha(\xi) = \min\left(1,\ \frac{\sum_{k=1}^{K} w(y_J, x_k)}{\sum_{k=1}^{K} w(x, y_k)}\right). \tag{12.21}$$

Hint. Track which factors contain x vs. y_J and use symmetry of w.

(d) *(Why MTM helps.)* Compare with (12.5): (12.21) is a K-sample average. If at least one y_k lands in a high-density region, the denominator is large and the ratio is bounded below regardless of how poor the other y_k's are. Vanilla MH commits to a single proposal. The cost is K density evaluations per step — worthwhile when these are cheap compared with mixing.

3. *(Optional)* **(Conditional detailed balance and delayed rejection.)** Standard detailed balance (12.2) demands $\pi(dx)P(x, dy)$ be symmetric under $x \leftrightarrow y$. Several modern methods relax this to symmetry on a specific event. We develop the simplest such case, *delayed rejection* (DR) [71, 36].

Conditional detailed balance. For a measurable event $E \subset \mathbb{R}^d \times \mathbb{R}^d$, the kernel P satisfies *conditional DB on E* if

$$\pi(dx)\, P(x, dy)\, \mathbf{1}_E(x, y) \;=\; \pi(dy)\, P(y, dx)\, \mathbf{1}_E(y, x). \tag{12.22}$$

If events $\{E_i\}$ partition the joint space and each sub-kernel respects conditional DB on its event, the total kernel is π-stationary by integration.

DR setup. At x, propose $y_1 \sim q_1(\cdot \mid x)$ with standard MH rate $\alpha_1(x, y_1)$ from (12.5). On acceptance, $X_{k+1} = y_1$. On rejection, propose $y_2 \sim q_2(\cdot \mid x, y_1)$ from a possibly different kernel (allowed to depend on y_1) and accept with probability $\alpha_2(x, y_1, y_2)$ to be determined.

(a) *(Why standard DB fails.)* Decompose the move $X_k \to X_{k+1}$ into stage-1-accept, stage-2-accept, and reject-both. Stage-1-accept is vanilla MH and satisfies (12.2) on its own; the stage-2-accept transition $x \to y_2$ is coupled through y_1, with no matching standard-DB reverse. We design α_2 to make stage 2 satisfy conditional DB on the event "stage 1 rejected".

(b) *(Conditional DB equation.)* The stage-2-accept contribution to the joint $\pi(dx)P(x, dy_2)$ is

$$\pi(x)\, q_1(y_1 \mid x)\big(1 - \alpha_1(x, y_1)\big)\, q_2(y_2 \mid x, y_1)\, \alpha_2(x, y_1, y_2)\, dx\, dy_1\, dy_2.$$

Conditional DB (swap $x \leftrightarrow y_2$, keep y_1 fixed) reads

$$\begin{aligned} &\pi(x)\, q_1(y_1 \mid x)\big(1 - \alpha_1(x, y_1)\big)\, q_2(y_2 \mid x, y_1)\, \alpha_2(x, y_1, y_2) \\ &\quad = \pi(y_2)\, q_1(y_1 \mid y_2)\big(1 - \alpha_1(y_2, y_1)\big)\, q_2(x \mid y_2, y_1)\, \alpha_2(y_2, y_1, x). \end{aligned} \tag{12.23}$$

(c) *(Solve for α_2.)* The largest $\alpha_2 \in [0, 1]$ satisfying (12.23) is

$$\alpha_2(x, y_1, y_2) = \min\left(1,\ \frac{\pi(y_2)\, q_1(y_1 \mid y_2)\, q_2(x \mid y_2, y_1)}{\pi(x)\, q_1(y_1 \mid x)\, q_2(y_2 \mid x, y_1)}\, \frac{1 - \alpha_1(y_2, y_1)}{1 - \alpha_1(x, y_1)}\right). \tag{12.24}$$

Compared with vanilla MH, the extra ratio of $(1 - \alpha_1)$ factors accounts for the conditioning on first-stage rejection.

(d) *(Why DR helps.)* Useful when π has features at multiple scales. Tune q_1 for high acceptance with small moves. A rejection signals y_1 in a low-density region, so q_2 can take a larger or differently-oriented step. Vanilla MH cannot use rejection information; DR pays the cost of a second proposal only when the first fails.

(e) *(More than two stages.)* Generalize (12.24) to a K-stage scheme: if stages $1, \ldots, J-1$ have all rejected, propose $y_J \sim q_J(\cdot \mid x, y_1, \ldots, y_{J-1})$ and accept with α_J satisfying conditional DB on the event "stages $1, \ldots, J-1$ rejected".

4. *(Optional)* **(Biased progressive sampling and NUTS: a unification.)** HMC with fixed integration time T requires manual tuning. The No-U-Turn Sampler (NUTS) [39] adapts T automatically. The original presentation seemed to break HMC's involution structure; Betancourt [8] reformulated NUTS as *biased progressive sampling*, combining (i) multinomial selection on an orbit (Exercise 2 analogue) and (ii) a sequence of accept/reject decisions each satisfying conditional DB (Exercise 3 analogue).

Setup. HMC's leapfrog map Φ generates an orbit $\mathcal{O} = \{\Phi^j(x_0, p_0) : -L/2 \le j \le L/2\}$ in phase space, balanced around (x_0, p_0). The target on orbits is $\tilde{\pi}(\mathcal{O}) \propto e^{-H(x,p)}$ for any representative point (leapfrog is volume-preserving). One step:

(1) Sample $p_0 \sim \mathcal{N}(0, I)$, fixing the orbit.

(2) Select $(x^*, p^*) \in \mathcal{O}$ as the new X_{k+1}.

Two selection mechanisms:

Multinomial. Sample one state from the entire orbit with weights

$$P\big(\text{select } (x_i, p_i)\big) = \frac{e^{-H(x_i,p_i)}}{\sum_j e^{-H(x_j,p_j)}}.$$

Combined with momentum-flip-and-reverse (an involution, as in Section 12.3), this is a multiple-try-style step: one accept/reject over the whole orbit, full (unconditional) DB.

Progressive. Walk along the orbit one leapfrog step at a time, deciding whether to "jump" the proposal pointer to the new endpoint at each extension. Each jump satisfies conditional DB; the proposal can travel arbitrarily far, controlled by an adaptive stopping criterion (the No U-Turn condition).

(a) *(Conditional DB for one progressive step.)* Current proposal at $(x^{(j)}, p^{(j)})$; extend the trajectory to $(x^{(j+1)}, p^{(j+1)})$. Let α_j be the probability of advancing the pointer, conditional on the extension. Show that the biased choice

$$\alpha_j = \min\left(1,\ \frac{e^{-H(x^{(j+1)},p^{(j+1)})}}{e^{-H(x^{(j)},p^{(j)})}}\right) \tag{12.25}$$

satisfies conditional DB (analogue of (12.23)) while being strictly more progressive than a uniform pick.

(b) *(Multinomial vs. progressive.)* Compare:

- Multinomial: one decision per orbit; stores all states; computes $\sum e^{-H}$; rejection only if the whole trajectory is bad.
- Progressive: one decision per leapfrog step; online (no storage); biased toward farther moves (each successful jump pushes the pointer forward).

"Biased toward long moves" is exactly what HMC wants: long trajectories explore π rapidly through ballistic dynamics. NUTS combines both [8]: build the orbit by progressively doubling its length, then select via multinomial draw over the constructed orbit. Conditional DB on the orbit; length adapts without user tuning.

(c) *(Unifying picture.)* Every accept-reject MCMC method in Section 12 is a target-preserving deterministic operation on an extended state, varying in:

- extended state and reference measure (the auxiliary randomness);
- involution ϕ;
- DB imposed unconditionally vs. conditionally on a partition.

The GIST framework [11] unifies adaptive HMC variants (randomized HMC, multinomial HMC, NUTS) further by treating tuning parameters as auxiliary variables of one involution.

References

[1] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[2] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Jason M Altschuler, Sinho Chewi, and Matthew S Zhang. Shifted composition IV: Toward ballistic acceleration for log-concave sampling. *arXiv:2506.23062*, 2025.

[4] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[5] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[6] Christophe Andrieu, Anthony Lee, and Sam Livingstone. A general perspective on the metropolis-hastings kernel. *arXiv preprint arXiv:2012.14881*, 2020.

[7] A Beskos, N Pillai, G Roberts, JM Sanz-Serna, and A Stuart. Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 2013.

[8] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[9] Nawaf Bou-Rabee, Bob Carpenter, Tore Selland Kleppe, and Sifan Liu. The within-orbit adaptive leapfrog no-u-turn sampler. *arXiv preprint arXiv:2506.18746*, 2025.

[10] Nawaf Bou-Rabee, Bob Carpenter, Tore Selland Kleppe, and Milo Marsden. Incorporating local step-size adaptivity into the No-U-Turn sampler using Gibbs self tuning. *arXiv preprint arXiv:2408.08259*, 2024.

[11] Nawaf Bou-Rabee, Bob Carpenter, and Milo Marsden. GIST: Gibbs self-tuning for locally adaptive Hamiltonian Monte Carlo. *arXiv preprint arXiv:2404.15253*, 2024.

[12] Yu Cao, Jianfeng Lu, and Lihan Wang. Complexity of randomized algorithms for underdamped Langevin dynamics. *Communications in Mathematical Sciences*, 19(7):1827–1853, 2021.

[13] José A Carrillo, Yifan Chen, Daniel Zhengyu Huang, Jiaoyang Huang, and Dongyi Wei. Fisher-rao gradient flow: Geodesic convexity and functional inequalities. *SIAM Journal on Mathematical Analysis*, 58(2):1062–1099, 2026.

[14] Baojun Che, Yifan Chen, Zhenghao Huan, Daniel Zhengyu Huang, and Weijie Wang. Stable derivative free gaussian mixture variational inference for bayesian inverse problems. *SIAM Journal on Scientific Computing*, 47(5):A2583–A2608, 2025.

[15] Baojun Che, Yifan Chen, Daniel Zhengyu Huang, Xinying Mao, and Weijie Wang. Adaptive exponential integration for stable gaussian mixture black-box variational inference. *arXiv preprint arXiv:2601.14855*, 2026.

[16] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023.

[17] Yifan Chen. New affine invariant ensemble samplers and their dimensional scaling. *arXiv preprint arXiv:2505.02987*, 2025.

[18] Yifan Chen, Xiaoou Cheng, Jonathan Niles-Weed, and Jonathan Weare. Convergence of unadjusted langevin in high dimensions: Delocalization of bias. *Communications on Pure and Applied Mathematics*, page e70032, 2024.

[19] Yifan Chen, Daniel Zhengyu Huang, Jiaoyang Huang, Sebastian Reich, and Andrew M Stuart. Sampling via gradient flows in the space of probability measures. *arXiv preprint arXiv:2310.03597*, 2023.

[20] Yifan Chen and Sifan Liu. Rotated mean-field variational inference and iterative gaussianization. *arXiv preprint arXiv:2510.07732*, 2025.

[21] Yifan Chen, Daniel Zhengyu Huang, Jiaoyang Huang, Sebastian Reich, and Andrew M Stuart. Efficient, multimodal, and derivative-free bayesian inference with fisher–rao gradient flows. *Inverse Problems*, 40(12):125001, 2024.

[22] Sinho Chewi. Log-concave sampling. *Book draft available at https://chewisinho.github. io*, 9:17–18, 2023.

[23] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Kamélia Daudel et al. Mixture weights optimisation for alpha-divergence variational inference. *Advances in Neural Information Processing Systems*, 34:4397–4408, 2021.

[25] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[26] Michael Ziyang Diao, Krishna Balasubramanian, Sinho Chewi, and Adil Salim. Forward-backward gaussian variational inference via jko in the bures-wasserstein space. In *International Conference on Machine Learning*, pages 7960–7991. PMLR, 2023.

[27] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[28] Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*.

[29] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306–312, 2013.

[30] Alfredo Garbuno-Inigo, Franca Hoffmann, Wuchen Li, and Andrew M Stuart. Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1):412–441, 2020.

[31] Alfredo Garbuno-Inigo, Nikolas Nüsken, and Sebastian Reich. Affine invariant interacting Langevin dynamics for Bayesian inference. *SIAM Journal on Applied Dynamical Systems*, 19(3):1633–1658, 2020.

[32] Andrew Gelman, Walter R Gilks, and Gareth O Roberts. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

[33] Nathan Glatt-Holtz, Justin Krometis, and Cecilia Mondaini. On the accept–reject mechanism for metropolis–hastings algorithms. *The Annals of Applied Probability*, 33(6B):5279–5333, 2023.

[34] Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science*, 5(1):65–80, 2010.

[35] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[36] Peter J Green and Antonietta Mira. Delayed rejection in reversible jump metropolis–hastings. *Biometrika*, 88(4):1035–1053, 2001.

[37] Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

[38] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[39] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[40] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[41] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[44] Marc Lambert, Sinho Chewi, Francis Bach, Silvère Bonnabel, and Philippe Rigollet. Variational inference via wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 35:14434–14447, 2022.

[45] Benedict Leimkuhler, Charles Matthews, and Jonathan Weare. Ensemble preconditioning for Markov chain Monte Carlo simulation. *Statistics and Computing*, 28:277–290, 2018.

[46] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[47] Ruilin Li, Hongyuan Zha, and Molei Tao. Sqrt (d) dimension dependence of langevin monte carlo. In *The International Conference on Learning Representations*, 2022.

[48] Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*, pages 3992–4002. PMLR, 2019.

[49] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[50] Jun S Liu, Faming Liang, and Wing Hung Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[51] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

[52] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[53] Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, pages 47–95, 2011.

[54] F Otto. The geometry of dissipative evolution equations: The porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174, 2001.

[55] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[56] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[57] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[58] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

[59] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341, 1996.

[60] Peter J Rossky, Jimmie D Doll, and Harold L Friedman. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

[61] Daniel Sanz-Alonso and Omar Al-Ghattas. A first course in monte carlo methods. *arXiv preprint arXiv:2405.16359*, 2024.

[62] Daniel Sanz-Alonso, Andrew Stuart, and Armeen Taeb. *Inverse problems and data assimilation*, volume 107. Cambridge University Press, 2023.

[63] Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, 195(1):79–148, 2022.

[64] Bin Shi, Simon S Du, Weijie Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

[65] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in neural information processing systems*, 36:62183–62223, 2023.

[66] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[67] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[68] Weijie Su, Stephen Boyd, and Emmanuel J Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

[69] Wenpin Tang and Hanyang Zhao. Score-based diffusion models via stochastic differential equations. *Statistic Surveys*, 19:28–64, 2025.

[70] Luke Tierney. A note on metropolis-hastings kernels for general state spaces. *Annals of applied probability*, pages 1–9, 1998.

[71] Luke Tierney and Antonietta Mira. Some adaptive monte carlo methods for bayesian inference. *Statistics in medicine*, 18(17-18):2507–2515, 1999.

[72] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[73] Matthew S Zhang, Sinho Chewi, Mufan Li, Krishnakumar Balasubramanian, and Murat A Erdogdu. Improved discretization analysis for underdamped Langevin Monte Carlo. *COLT*, 2023.