EECE 576L

Advanced Topics in Deep Learning

Lecture 5: Flow Matching and Diffusion

Renjie Liao

University of British Columbia

2026/27 Winter Session, Term 1 (Fall 2026)

Outline

- **Flows and flow matching**

  Velocity fields, conditional paths, and regression

- Diffusion and score SDEs

  Denoising, scores, and reverse-time sampling

- Connections and practice

  Probability-flow ODEs, fast sampling, and conditioning

Normalizing Flows

A normalizing flow learns an invertible map from noise to data space.

$$Z \sim \mathcal{N}(0, I), \quad F_\theta = f_m \circ \cdots \circ f_1, \quad X_1 = F_\theta(Z)$$



Example layers in two dimensions: affine scaling and nonlinear shear

$$f_1(u, v) = (e^a u + b, v), \qquad f_2(u, v) = (u, v + c \sin u)$$

$$a, b, c \in \mathbb{R}$$

- Differentiable bijections with nonzero Jacobian determinants.
- Tractable inverses and determinants for efficient likelihood evaluation.

Many small invertible steps lead to a continuous flow.

[1] DSL-Lab, 2024. [2] Dinh, Sohl-Dickstein, and Bengio, ICLR 2017.

Flows as Velocity Fields

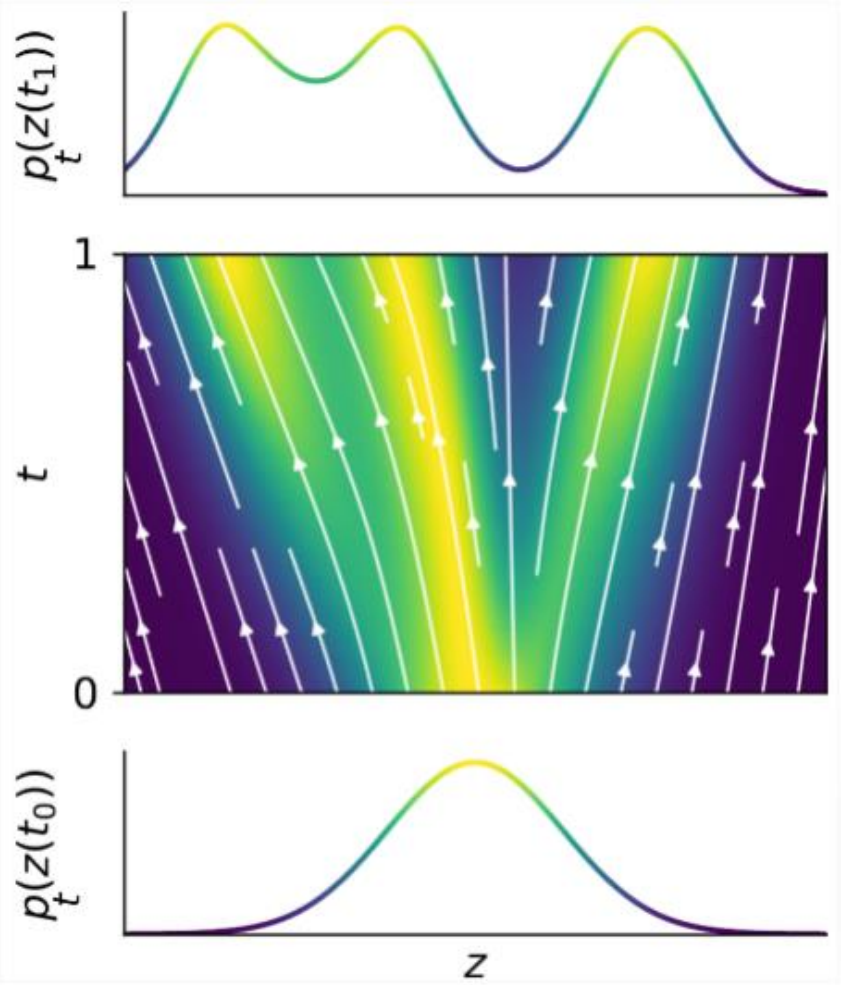
A neural network predicts how the current sample should move.

$$\frac{dX_t}{dt} = v_\theta(X_t, t)$$

$$X_0 = Z \sim \mathcal{N}(0, I)$$

Generation time

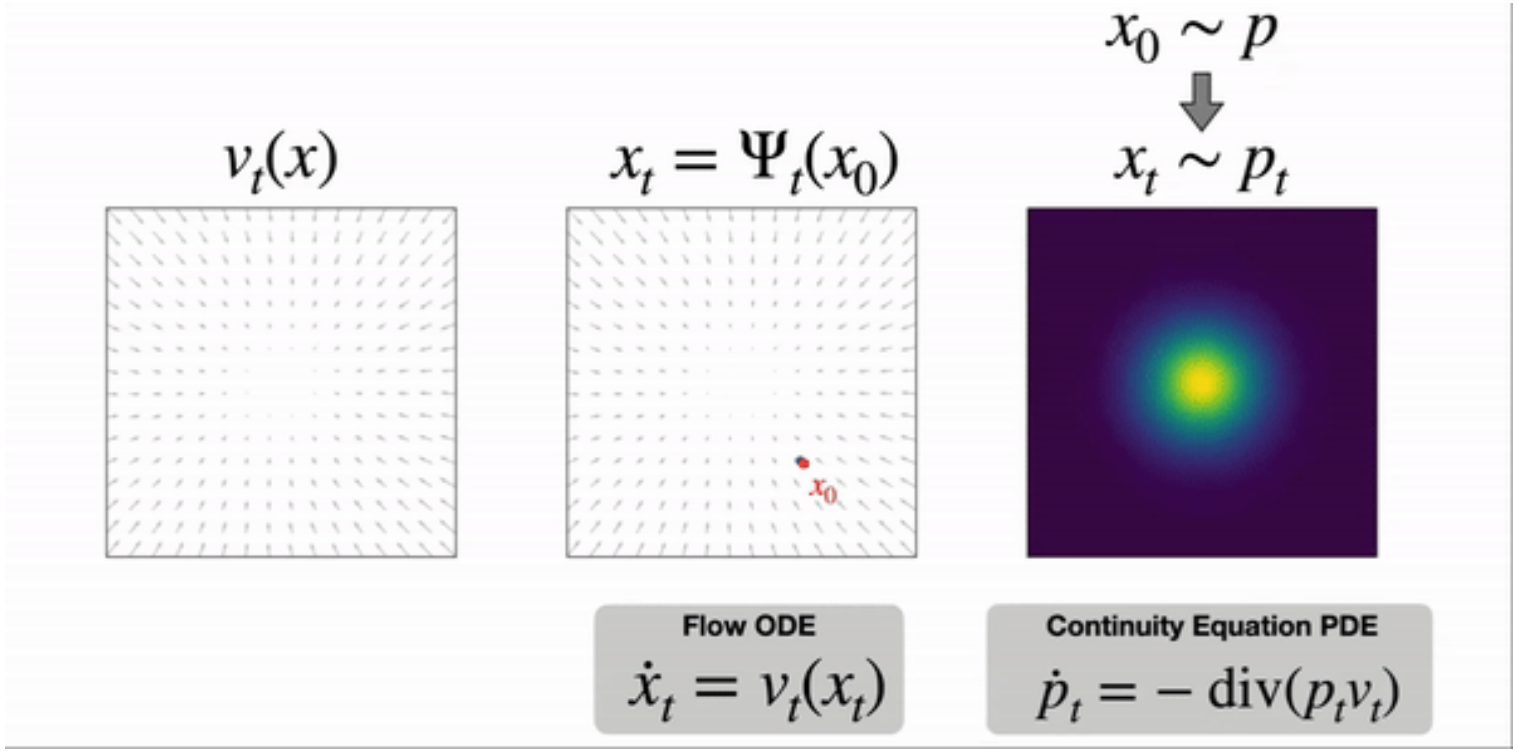
Generation starts at zero and ends at one.



[3] Chen et al., NeurIPS 2018. [4] CERN Indico lecture material, 2024.

Particles and Probability Density

Moving samples changes the density while preserving total probability.



$$\partial_t p_t(x) + \nabla_x \cdot \big(p_t(x) v_t(x)\big) = 0$$

Derivation: Appendix A2, probability conservation.

[5] Lipman et al., ICLR 2023. [6] Yaron Lipman, flow-matching lecture slides.

Endpoint Conditions and a Probability Path

Generation should move a Gaussian distribution into the data distribution.

$$p_0 = \mathcal{N}(0, I), \qquad p_t \Longrightarrow p_{\text{data}} \quad (t \uparrow 1)$$

The arrow denotes convergence in distribution.

The continuity equation relates a chosen density path to its velocity.

$$\partial_t p_t(x) + \nabla_x \cdot \big(p_t(x) u_t(x)\big) = 0$$

The endpoints leave many possible paths. We choose one that is easy to sample.

Next: construct a path around each training example, then combine those paths.

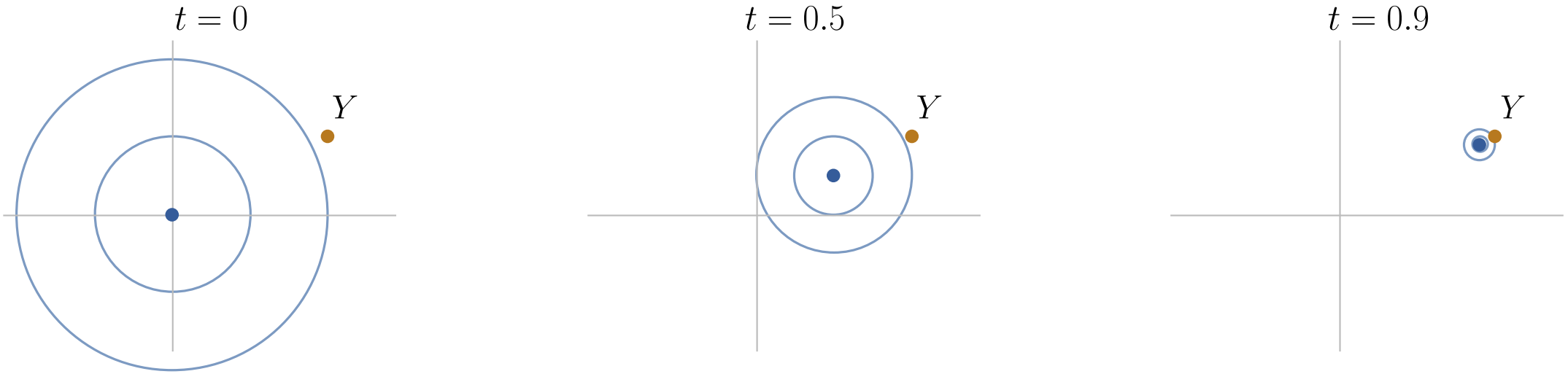
[5] Lipman et al., ICLR 2023.

A Conditional Gaussian Path

Fix one data example. Move a Gaussian's center toward it and shrink its spread.

$$p_t(x \mid Y) = \mathcal{N}\big(x; \mu_t(Y), \sigma_t^2 I\big)$$

$$\mu_t(Y) = tY, \qquad \sigma_t = 1 - t, \qquad 0 \le t < 1$$



Blue: Gaussian mean. Gold: fixed data example.

At the endpoint, the Gaussian concentrates at the fixed data example.

[5] Lipman et al., ICLR 2023.

An Affine Map Samples the Chosen Path

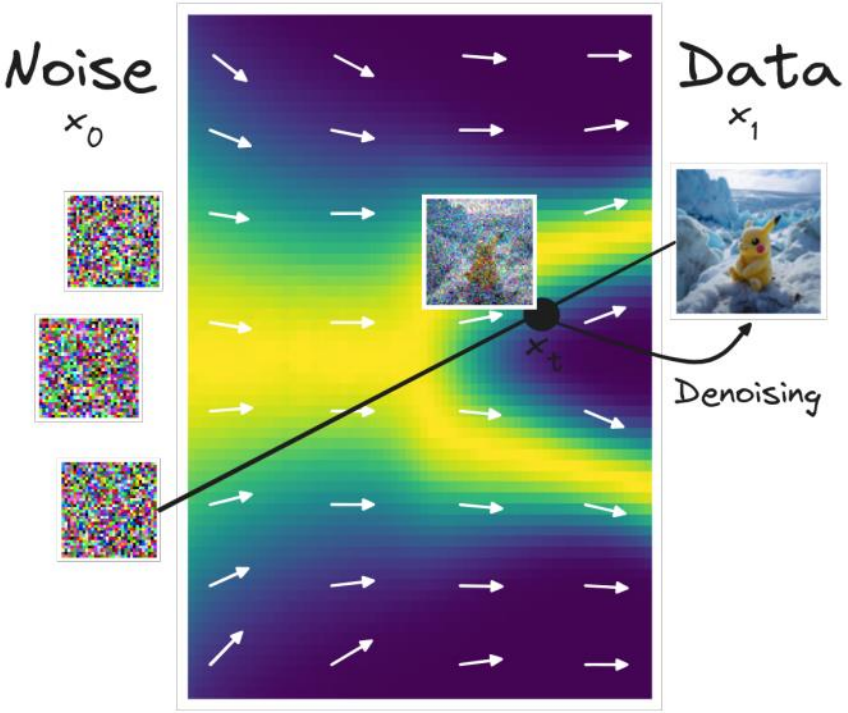
A shifted and scaled Gaussian realizes the chosen density.

$$\Psi_t(Z \mid Y) = \mu_t(Y) + \sigma_t Z$$

$$Z \sim \mathcal{N}(0, I)$$

$$X_t = (1 - t)Z + tY$$

$$X_t \mid Y \sim p_t(\cdot \mid Y)$$



Sample a time and compute the state directly. No ODE solve is needed in training.

[5] Lipman et al., ICLR 2023. [6] Yaron Lipman, flow-matching lecture slides.

Deriving the Conditional Velocity

Differentiate the trajectory while holding its noise and data endpoints fixed.

$$\frac{d}{dt}\Psi_t(Z \mid Y) = Y - Z$$

Recover the initial noise from the current position.

$$x = (1-t)Z + tY \quad \Longrightarrow \quad Z = \frac{x - tY}{1-t}$$

Substitute to obtain a field defined at any current location.

$$u_t(x \mid Y) = \frac{Y - x}{1-t}, \qquad u_t(X_t \mid Y) = Y - Z, \qquad t < 1$$

The affine map makes this density and velocity satisfy continuity. Appendix A6.

[5] Lipman et al., ICLR 2023.

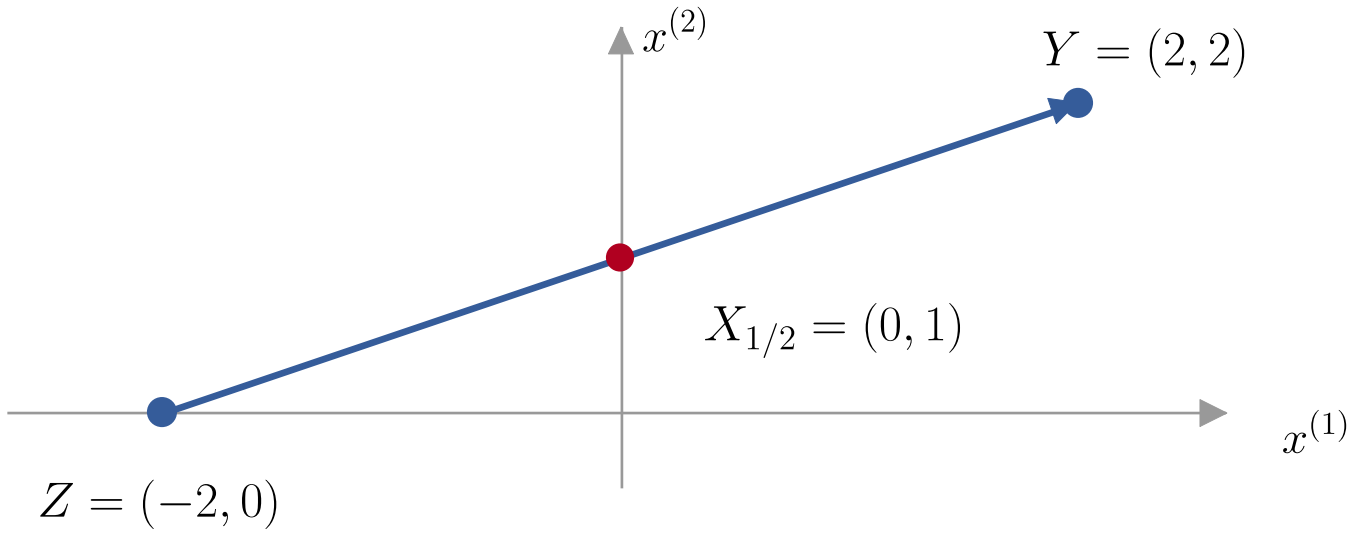
A Known Velocity Target

A line between two samples gives an exact velocity target.

$$\dot{X}_t = Y - Z$$

$$Y - Z = (4, 2)$$

Example: a two-dimensional training pair



[5] Lipman et al., ICLR 2023.

Conditional Flow Matching: Training

A minibatch update is supervised regression.

1. Draw data, noise, and a random time.

$$Y \sim p_{\text{data}}, \quad Z \sim \mathcal{N}(0, I), \quad t \sim \mathcal{U}(0, 1)$$

2. Form the input and target.

$$X_t = (1 - t)Z + tY, \qquad U = Y - Z$$

3. Predict velocity and update the parameters.

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}\Big[\|v_\theta(X_t, t) - U\|^2\Big]$$

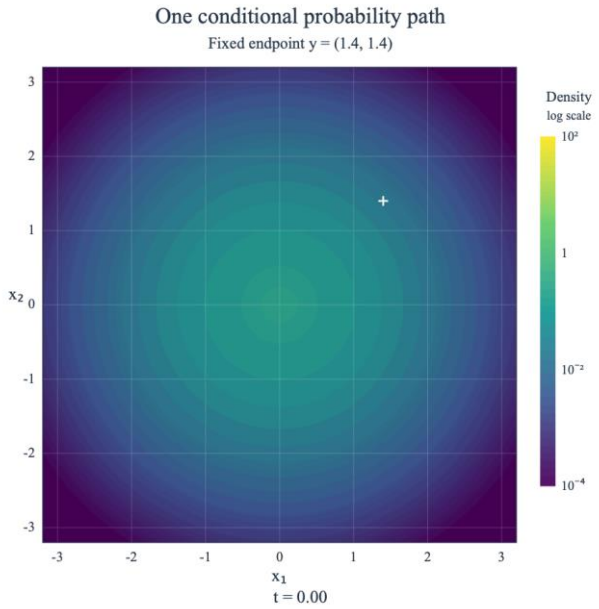
Training samples the path directly. It does not integrate the learned ODE.

[5] Lipman et al., ICLR 2023.

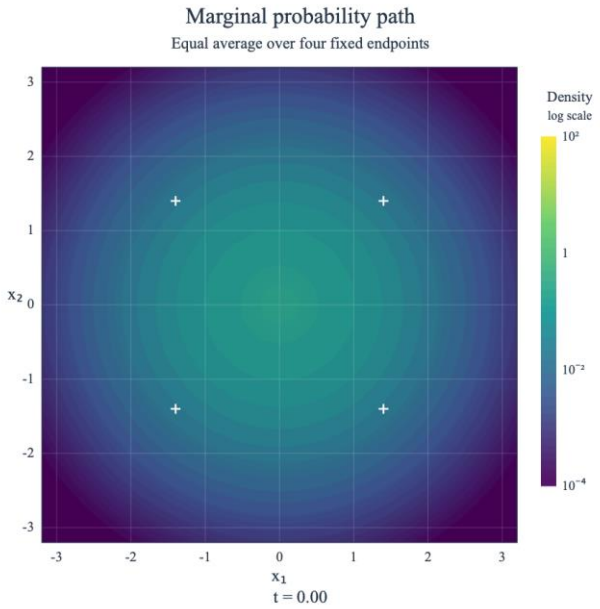
Many Conditional Paths, One Marginal Field

Conditional probability paths combine into one marginal density.

Conditional path: fix one data example



Marginal path: average over data



$$u_t(x) = \mathbb{E}[Y - Z \mid X_t = x]$$

White crosses mark the data endpoints. Pairwise targets train the marginal field.

[5] Lipman et al., ICLR 2023.

Result 1: Conditional Paths Define a Marginal Flow

Each derived velocity transports its conditional Gaussian.

Averaging over clean endpoints gives a marginal (mixture) density:

$$p_t(x) = \mathbb{E}_Y[p_t(x \mid Y)]$$

Four equally weighted endpoints give an average of four Gaussian densities.

$$u_t(x) = \mathbb{E}[u_t(x \mid Y) \mid X_t = x] = \mathbb{E}[Y - Z \mid X_t = x]$$

$$\partial_t p_t(x) + \nabla_x \cdot (p_t(x) u_t(x)) = 0$$

Start with Gaussian noise and follow this field. The sample distribution follows the prescribed path and converges to the data distribution.

Exact field, matching initialization and a unique ODE flow. Proof: Appendix A4.

[5] Lipman et al., ICLR 2023.

Result 2: Conditional Regression Has the Same Gradient

The pairwise target is available; the desired marginal velocity is unknown.

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}\|v_\theta(X_t, t) - U\|^2, \qquad U = Y - Z$$

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\|v_\theta(X_t, t) - u_t(X_t)\|^2$$

$$\mathcal{L}_{\mathrm{CFM}} = \mathcal{L}_{\mathrm{FM}} + C, \qquad \nabla_\theta \mathcal{L}_{\mathrm{CFM}} = \nabla_\theta \mathcal{L}_{\mathrm{FM}}$$

The constant is residual target variance and is independent of the network.

Training needs no density, posterior-weight calculation, or marginal-field labels.

The conditional loss can stay positive at the correct field. Proof: Appendix A5.

[5] Lipman et al., ICLR 2023.

Flow Matching: Sampling

A new noise sample follows the learned velocity field.

$$X_0 \sim \mathcal{N}(0, I), \qquad \frac{dX_t}{dt} = v_\theta(X_t, t)$$

Euler with N steps, indexed from zero to N − 1:

$$h = \frac{1}{N}, \quad t_k = kh, \quad x_{k+1} = x_k + h\, v_\theta(x_k, t_k), \quad x_0 \sim \mathcal{N}(0, I)$$

Return the final state as a generated sample.

The field is reevaluated after every update.

Step size and solver order control numerical error and computation.

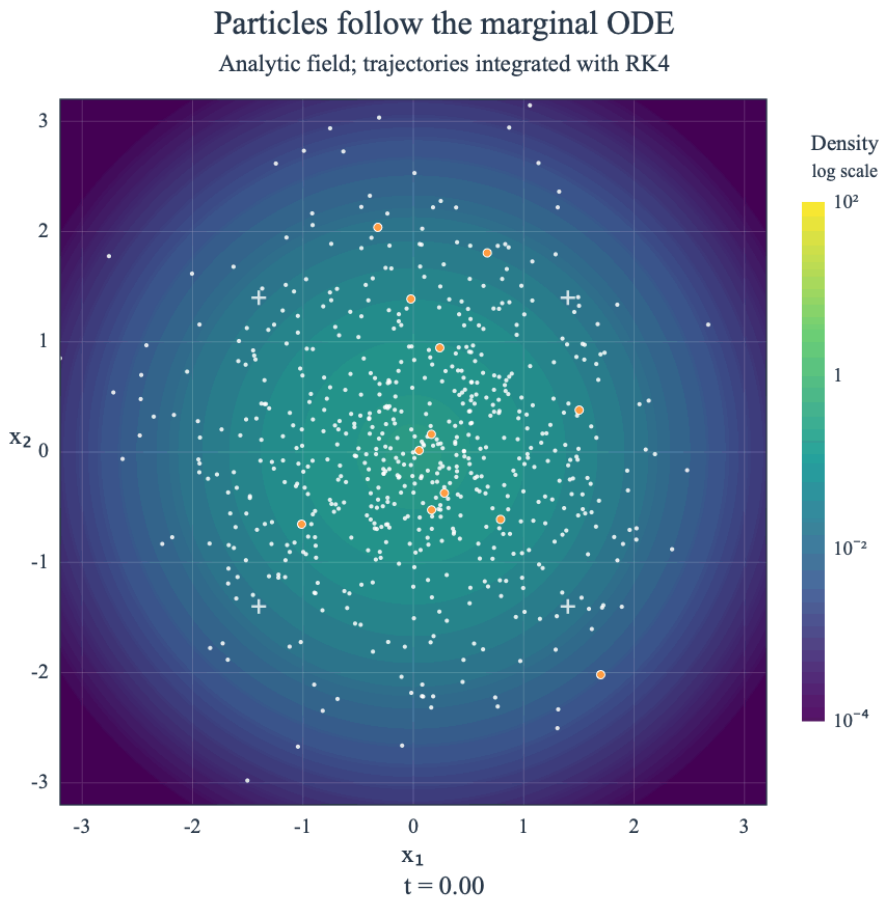
[5] Lipman et al., ICLR 2023.

The Marginal Flow in Action

Particles follow the exact marginal velocity field.

$$\frac{dX_t}{dt} = u_t(X_t)$$

Straight conditional training paths can produce curved marginal trajectories.

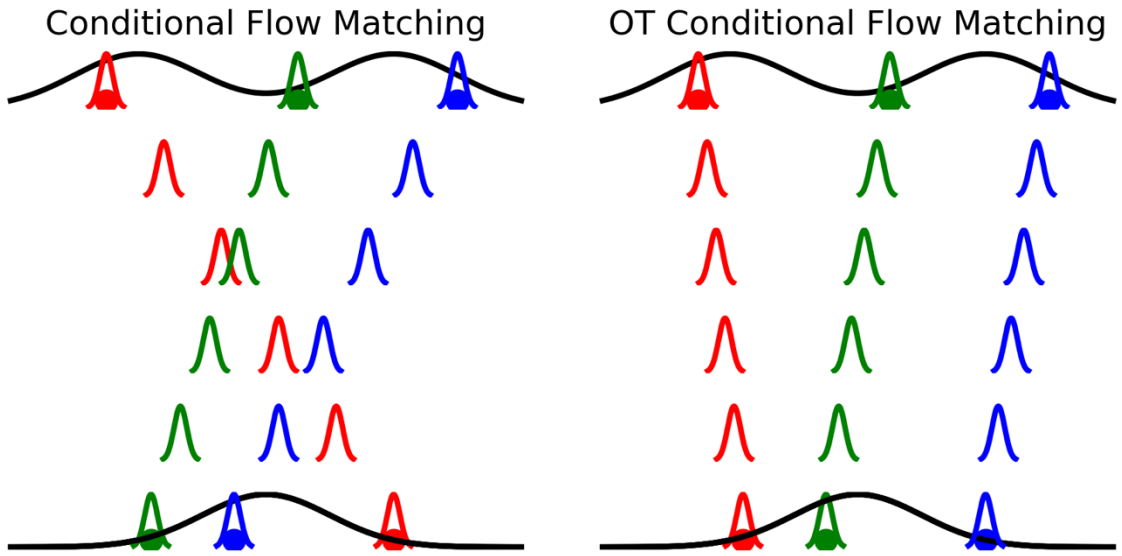


Analytic example; the animation stops before the singular endpoint.

[5] Lipman et al., ICLR 2023.

Pairing Noise and Data

Vanilla matching pairs noise and data independently.



Minibatch optimal transport minimizes total squared displacement within a batch.

Changing the pairing changes the regression problem.

[7] Tong et al., TMLR 2024.

Outline

- Flows and flow matching

  Velocity fields, conditional paths, and regression

- **Diffusion and score SDEs**

  Denoising, scores, and reverse-time sampling

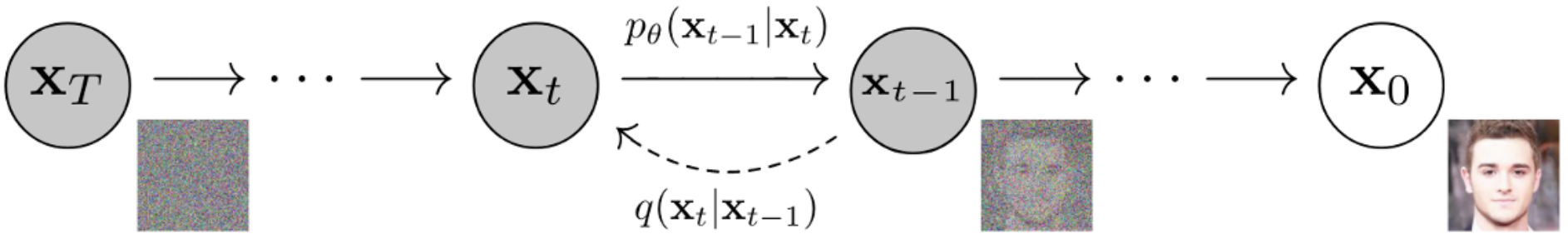
- Connections and practice

  Probability-flow ODEs, fast sampling, and conditioning

Diffusion: Learning to Reverse Noising

Choose an easy corruption process. Learn how to generate in reverse.

$$X_0 := Y \sim p_{\text{data}}$$



Known forward process: data becomes approximately Gaussian noise.

Learned reverse process: noisy states become samples resembling data.

Forward diffusion starts with clean data at step zero. Larger indices mean more noise.

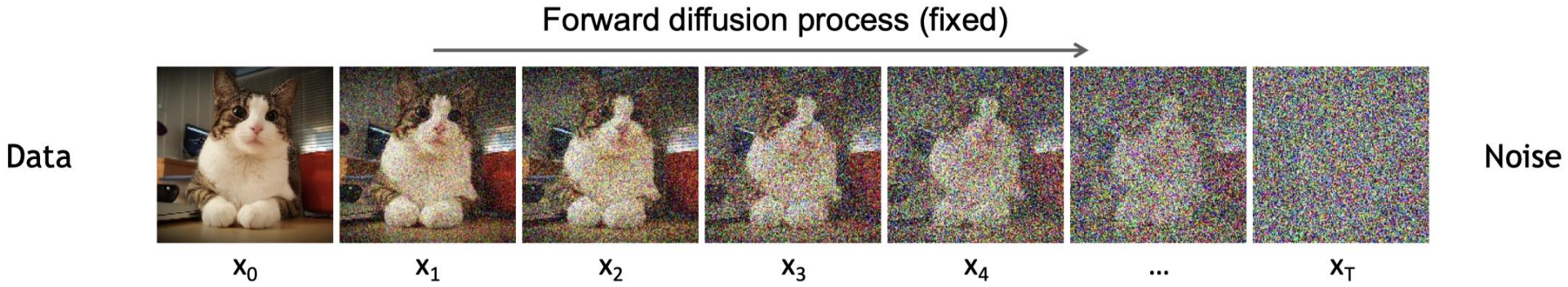
[8] Ho et al., NeurIPS 2020. [9] Song, Meng, and Vahdat, CVPR 2023 tutorial.

DDPM: A Fixed Noising Chain

A small Gaussian perturbation at every discrete step.

$$X_k = \sqrt{\alpha_k}\, X_{k-1} + \sqrt{\beta_k}\, \epsilon_k, \qquad \alpha_k = 1 - \beta_k$$

$$\epsilon_k \sim \mathcal{N}(0, I), \qquad X_0 \sim p_{\text{data}}, \qquad k = 1, \ldots, K$$



Each adjacent state adds a small, independent Gaussian perturbation.

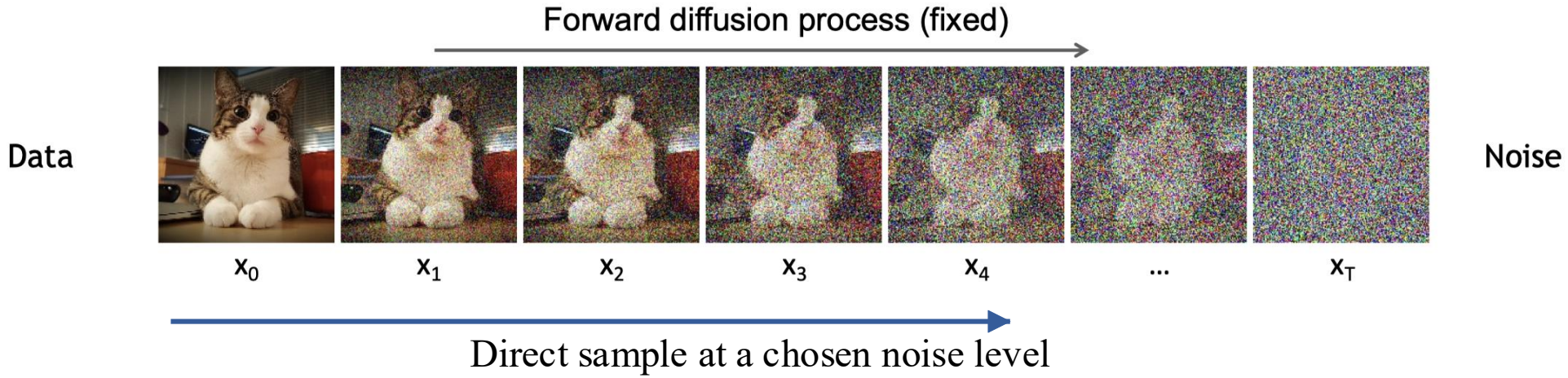
[8] Ho et al., NeurIPS 2020. [9] Song, Meng, and Vahdat, CVPR 2023 tutorial.

Training Can Sample Any Noise Level Directly

One clean example and one Gaussian draw produce a chosen noisy state.

$$\bar{\alpha}_k = \prod_{j=1}^{k} \alpha_j, \qquad X_k = \sqrt{\bar{\alpha}_k}\, X_0 + \sqrt{1 - \bar{\alpha}_k}\, \epsilon$$

$$X_0 \sim p_{\text{data}}, \qquad \epsilon \sim \mathcal{N}(0, I)$$

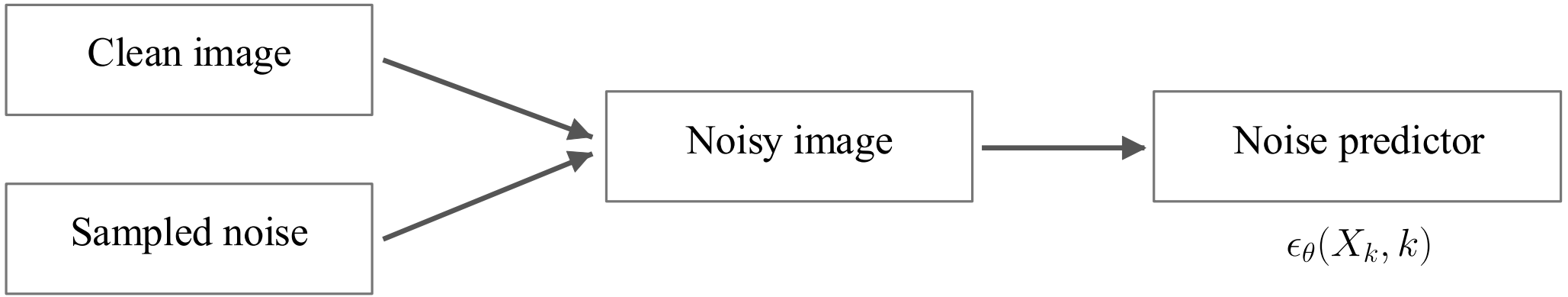


A closed-form marginal creates any chosen noisy state in one step.

[8] Ho et al., NeurIPS 2020. [9] Song, Meng, and Vahdat, CVPR 2023 tutorial.

DDPM: Why Random Noise Can Be Predicted

The noise is random, but it is present in the observed noisy image.



$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{k,X_0,\epsilon}\|\epsilon_\theta(X_k, k) - \epsilon\|_2^2$$

$$\mathbb{E}[\epsilon] = 0, \qquad \epsilon^*(x, k) = \mathbb{E}[\epsilon \mid X_k = x]$$

The conditional mean need not be zero. Exact recovery of each draw is not required.

[8] Ho et al., NeurIPS 2020. [10] Song et al., ICLR 2021.

Likelihood, ELBO, and Noise Regression

Likelihood is evaluated at a fixed clean observation. The noisy states are latent.

Evidence lower bound (ELBO)

$$\log p_\theta(x_0) \geq \mathcal{E}_\theta(x_0),$$
$$\mathcal{L}_{\mathrm{VLB}}(\theta) = -\mathbb{E}_{X_0 \sim p_{\mathrm{data}}} \mathcal{E}_\theta(X_0).$$

Lowercase: one clean observation. Uppercase: a random training draw.

A known noising chain makes this variational objective tractable.

For fixed reverse variance, each interior Gaussian KL term becomes:

$$\mathcal{D}_k(\theta) = w_k \, \mathbb{E}\|\epsilon - \epsilon_\theta(X_k, k)\|^2 + C_k, \qquad k \geq 2$$

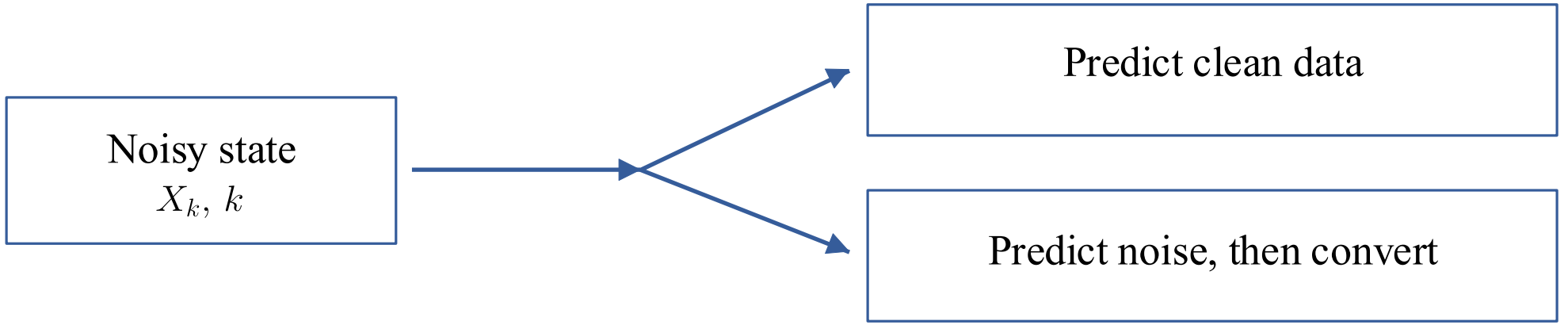
The weight is schedule-defined. The constant is independent of the network.

Unweighted noise MSE reweights the ELBO. Full bound: Appendix A16.

[8] Ho et al., NeurIPS 2020.

Predicting Clean Data

A network may directly estimate the clean training sample.



$$\mathcal{L}_{\text{data}} = \mathbb{E}\|\widehat{X}_{0,\theta}(X_k, k) - X_0\|^2$$

A noise prediction also yields a clean estimate:

$$\widehat{X}_{0,\theta}(x, k) = \frac{x - \sqrt{1-\bar{\alpha}_k}\,\epsilon_\theta(x, k)}{\sqrt{\bar{\alpha}_k}}$$

Loss weighting depends on the parameterization. See Appendix A17.

[8] Ho et al., NeurIPS 2020. [10] Song et al., ICLR 2021.

The Reverse Step as a Score Update

For Gaussian corruption, the conditional score is known from the sampled noise.

$$\sigma_k = \sqrt{1-\bar{\alpha}_k}, \qquad \nabla_x \log q_k(x \mid X_0) = -\frac{x - \sqrt{\bar{\alpha}_k} X_0}{\sigma_k^2} = -\frac{\epsilon}{\sigma_k}$$

$$s_\theta(x,k) := -\frac{\epsilon_\theta(x,k)}{\sigma_k} \approx \nabla_x \log q_k(x)$$

The trained noise predictor estimates the marginal score after this rescaling.

$$\mu_\theta(x,k) = \frac{1}{\sqrt{\alpha_k}}\left(x - \frac{\beta_k}{\sigma_k}\epsilon_\theta(x,k)\right) = \frac{x + \beta_k s_\theta(x,k)}{\sqrt{\alpha_k}}$$

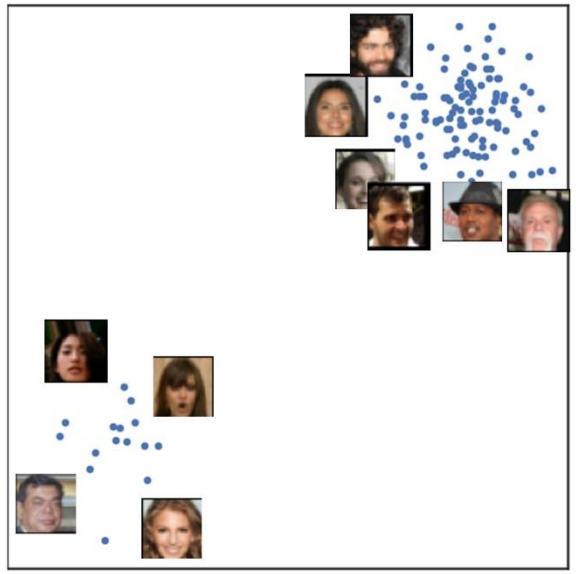
$$X_{k-1} = \mu_\theta(X_k,k) + \nu_k z, \quad z \sim \mathcal{N}(0,I), \quad \nu_1 = 0$$

Next: the geometry and denoising interpretation of this score.

[8] Ho et al., NeurIPS 2020. [10] Song et al., ICLR 2021.

The Score: A Local Direction of Increasing Density

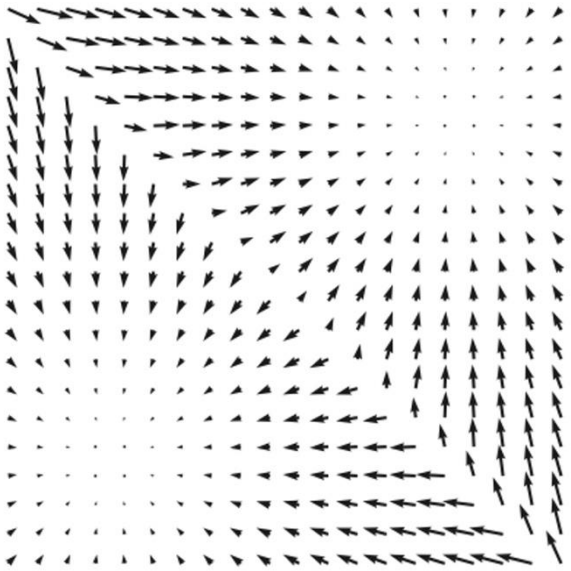
$$s_\tau(x) = \nabla_x \log q_\tau(x), \qquad q_0 = p_{\text{data}}$$



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

score matching



Scores

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Langevin dynamics



New samples

Arrows describe a local direction and magnitude, not a probability value.

[10] Song et al., ICLR 2021. [11] Yang Song, score-based modeling tutorial, 2021.

Known Corruption Gives a Known Conditional Score

Forward clock: clean data at zero, noise at one.

$$\tau \in [0,1], \qquad X_\tau = a_\tau X_0 + \sigma_\tau \epsilon, \qquad X_0 \sim p_{\text{data}}$$

$$q_\tau(x \mid x_0) = \mathcal{N}(x; a_\tau x_0, \sigma_\tau^2 I)$$

$$\nabla_x \log q_\tau(x \mid x_0) = -\frac{x - a_\tau x_0}{\sigma_\tau^2} = -\frac{\epsilon}{\sigma_\tau}$$

The clean example and sampled noise make this target available during training.

The marginal score remains unknown because it averages over possible clean sources.

Use positive noise levels for this formula; the zero-noise endpoint is singular.

[10] Song et al., ICLR 2021. [8] Ho et al., NeurIPS 2020.

Denoising Score Matching

Regress the score network onto the computable conditional target.

$$\mathcal{L}_{\mathrm{DSM}} = \mathbb{E}_{\tau, X_0, \epsilon}\left[\lambda(\tau)\left\| s_\theta(X_\tau, \tau) + \frac{\epsilon}{\sigma_\tau}\right\|_2^2\right]$$

A noisy image may have many plausible clean sources.

Squared-error regression averages their targets conditional on the noisy input.

$$s_\tau(x) = \mathbb{E}[\nabla_x \log q_\tau(x \mid X_0) \mid X_\tau = x]$$

At the ideal optimum, the learned function equals the marginal score.

[10] Song et al., ICLR 2021.

Noise and Score Prediction Are Linked

The same denoiser can provide a score estimate.

$$s_\theta(x,\tau) := -\frac{\epsilon_\theta(x,\tau)}{\sigma_\tau} \approx s_\tau(x)$$

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}\left[\frac{\lambda(\tau)}{\sigma_\tau^2}\,\|\epsilon_\theta(X_\tau,\tau)-\epsilon\|_2^2\right]$$

Choosing the score-loss weight proportional to noise variance recovers noise MSE.

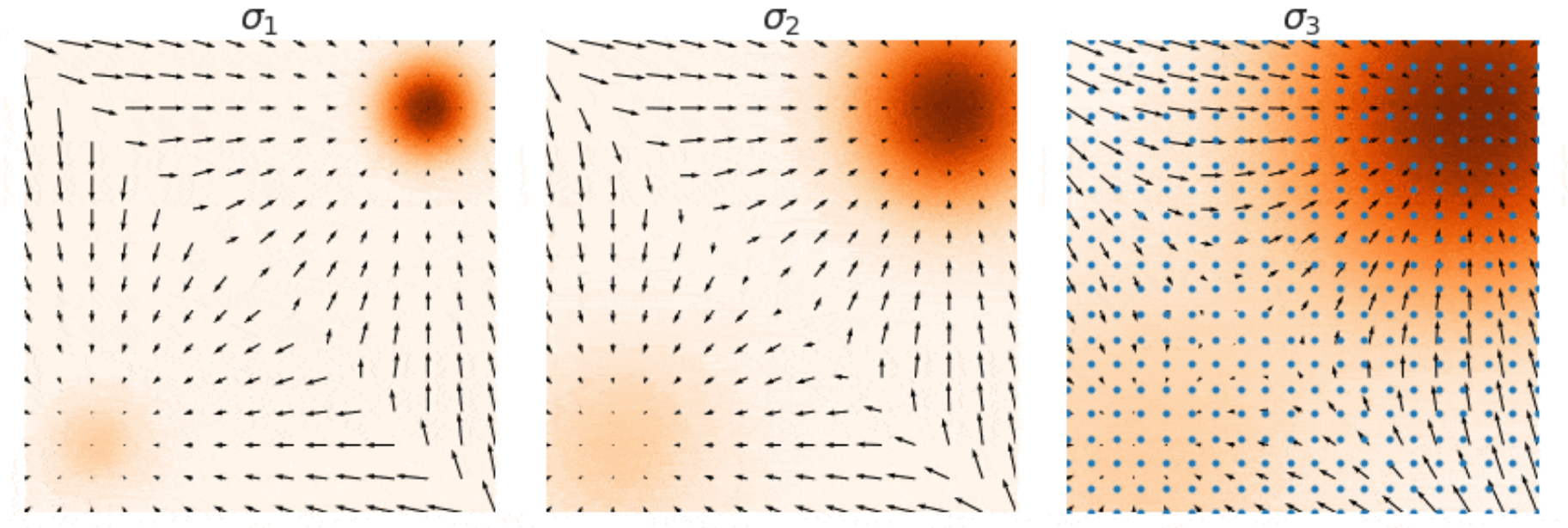
Output scaling and time weighting matter when comparing training objectives.

[10] Song et al., ICLR 2021. [8] Ho et al., NeurIPS 2020.

Sampling Along a Sequence of Score Fields

At one fixed noise level, score drift and random noise work together.

$$x \leftarrow x + \eta\, s_\theta(x, \tau) + \sqrt{2\eta}\, z, \qquad z \sim \mathcal{N}(0, I)$$



In this animation: high noise at right; refinement proceeds toward the left.

[10] Song et al., ICLR 2021. [11] Yang Song, score-based modeling tutorial, 2021.

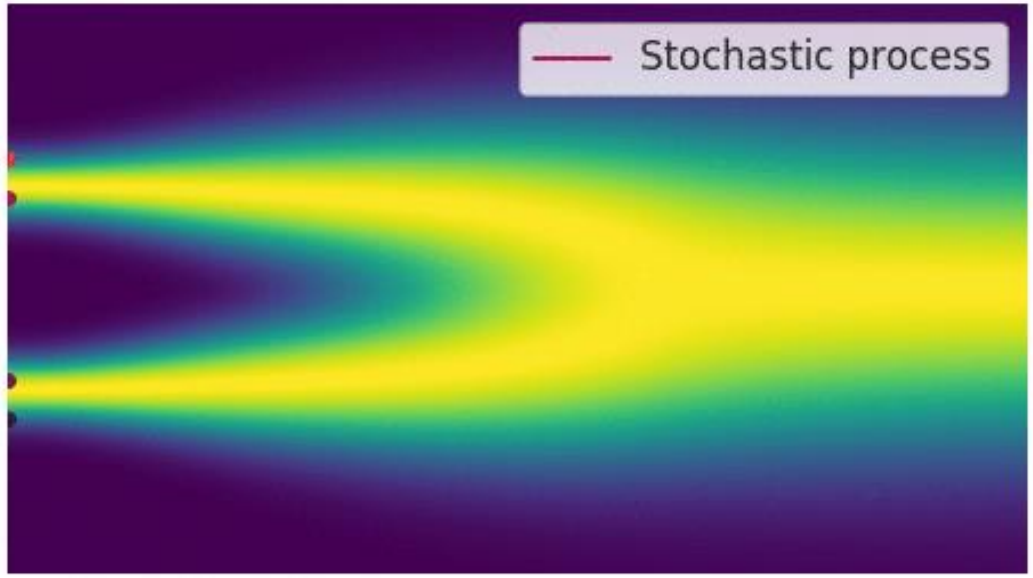
Continuous Noising: The Forward SDE

Linear drift gives Gaussian perturbation kernels with tractable training targets.

$$dX_\tau = f_\tau(X_\tau)\, d\tau + g_\tau\, dW_\tau, \qquad X_0 \sim p_{\text{data}}$$

Drift sets the mean motion. Brownian noise adds a random perturbation.



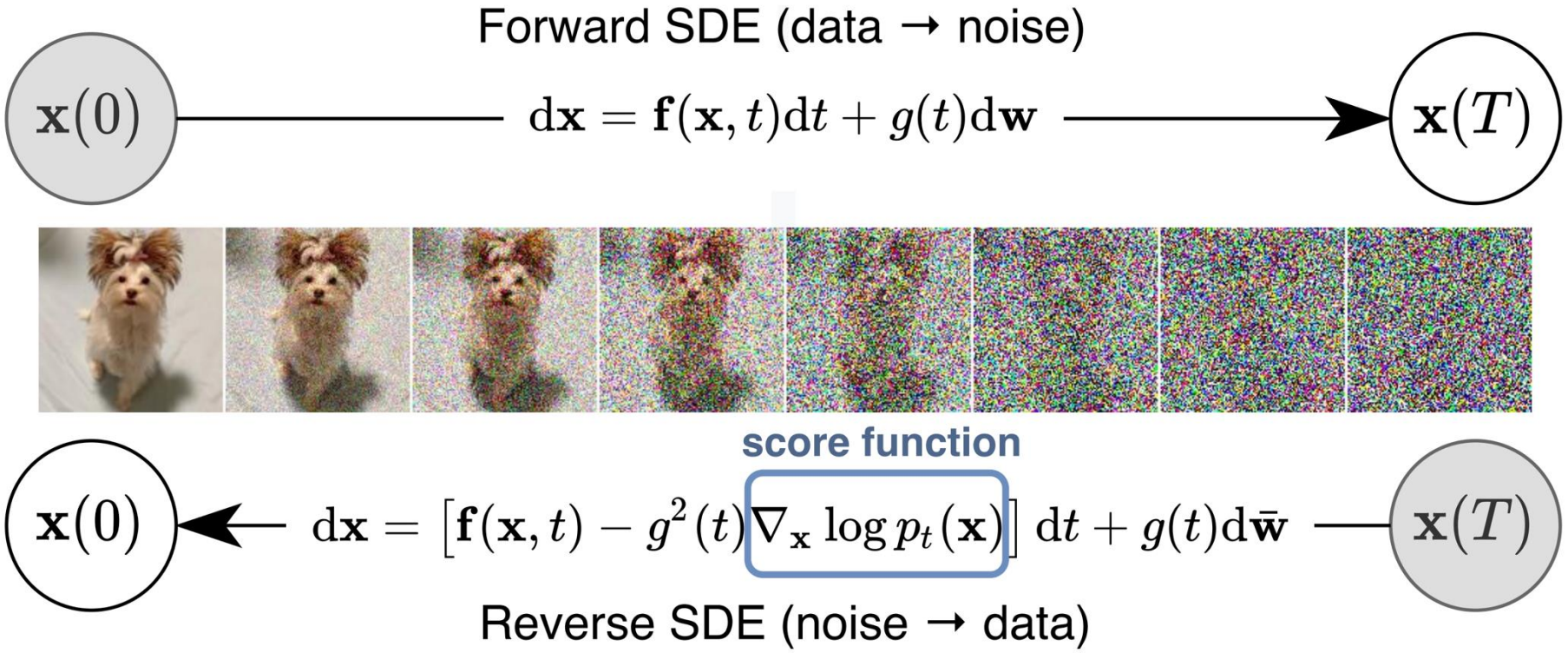


Forward clock: data to noise. The VP example is derived in Appendix A10.

[10] Song et al., ICLR 2021. [11] Yang Song, score-based modeling tutorial, 2021.

Forward Corruption and Reverse Generation

The score supplies the missing information for time reversal.



The original diagram uses forward time. Reversal takes negative time increments.

[10] Song et al., ICLR 2021. [11] Yang Song, score-based modeling tutorial, 2021.

The Reverse SDE in Generation Time

Generation time increases from noise to data.

$$\widetilde{X}_t = X_{1-t}, \qquad t \in [0,1], \qquad \widetilde{X}_0 \sim q_1 \approx \mathcal{N}(0, I)$$

$$d\widetilde{X}_t = \left[-f_{1-t}(\widetilde{X}_t) + g_{1-t}^2 \, s_{1-t}(\widetilde{X}_t) \right] dt + g_{1-t} \, dB_t$$

Reverse the drift and add a score-dependent correction.

The exact score gives the time reversal of the chosen forward diffusion.

In practice, insert the learned score estimate at the corresponding forward time.

$$s_{1-t}(\widetilde{X}_t) \;\approx\; s_\theta(\widetilde{X}_t, 1-t)$$

[10] Song et al., ICLR 2021.

Sampling: One Positive-Time Euler Step

Start with noise. Evaluate the score and take a small generation step.

$$\tau = 1 - t, \qquad h > 0, \qquad z \sim \mathcal{N}(0, I)$$

$$\widetilde{X}_{t+h} = \widetilde{X}_t + \left[-f_\tau(\widetilde{X}_t) + g_\tau^2 s_\theta(\widetilde{X}_t, \tau) \right] h + g_\tau \sqrt{h}\, z$$

1. Convert generation time to the forward noise level.
2. Evaluate the time-conditioned score network.
3. Apply the drift correction and fresh Gaussian noise.

Smaller steps improve the numerical approximation but require more evaluations.

[10] Song et al., ICLR 2021. [11] Yang Song, score-based modeling tutorial, 2021.

Diffusion Training and Generation

Training: independent noisy examples supply regression targets.



Generation: sequential dynamics reuse the learned score.



The next connection concerns a deterministic velocity for the same distribution path.

[10] Song et al., ICLR 2021.

Outline

- Flows and flow matching

  Velocity fields, conditional paths, and regression

- Diffusion and score SDEs

  Denoising, scores, and reverse-time sampling

- **Connections and practice**

  Probability-flow ODEs, fast sampling, and conditioning

The Probability-Flow ODE

The PF-ODE moves samples and reproduces the SDE's evolving density.

$$\frac{dx_\tau}{d\tau} = b_\tau(x_\tau), \qquad b_\tau(x) = f_\tau(x) - \frac{1}{2} g_\tau^2 \nabla_x \log q_\tau(x)$$

The resulting density obeys a continuity equation:

$$\partial_\tau q_\tau = -\nabla \cdot (q_\tau b_\tau) = -\nabla \cdot (f_\tau q_\tau) + \frac{1}{2} g_\tau^2 \Delta q_\tau$$

This is the SDE's Fokker–Planck equation. Equal densities allow different sample paths.

For generation, integrate backward in corruption time:

$$t = 1 - \tau, \qquad \widetilde{x}_t = x_{1-t}, \qquad \frac{d\widetilde{x}_t}{dt} = -b_{1-t}(\widetilde{x}_t)$$

Exact score and matching initialization. Derivation: Appendix A12.

[10] Song et al., ICLR 2021.

Probability Path, Target, and Sampler

The path and the learned target define the training problem.

Flow matching

$$X_t = (1-t)Z + tY, \qquad v_\theta(X_t, t) \longleftarrow Y - Z$$

Score matching

$$X_\tau = a_\tau X_0 + \sigma_\tau \epsilon, \qquad s_\theta(X_\tau, \tau) \longleftarrow -\epsilon/\sigma_\tau$$

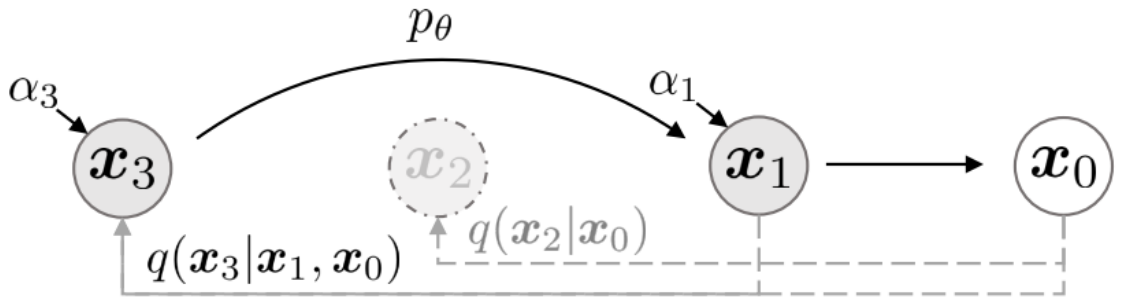
FM samples with an ODE. Diffusion can sample with an SDE or an ODE.

Diffusion probability paths can also be used for flow-matching training.

[5] Lipman et al., ICLR 2023. [10] Song et al., ICLR 2021.

DDIM: Fewer Denoiser Evaluations

Use the same noise predictor on a shorter sampling grid.



Source figure notation:

$$x_0^{\text{src}} \leftrightarrow X_0, \qquad \alpha_k^{\text{src}} = \bar{\alpha}_k$$

$$\widehat{X}_{0,k} = \frac{X_k - \sqrt{1-\bar{\alpha}_k}\,\epsilon_\theta(X_k, k)}{\sqrt{\bar{\alpha}_k}}$$

$$X_j = \sqrt{\bar{\alpha}_j}\,\widehat{X}_{0,k} + \sqrt{1-\bar{\alpha}_j}\,\epsilon_\theta(X_k, k), \quad j < k$$

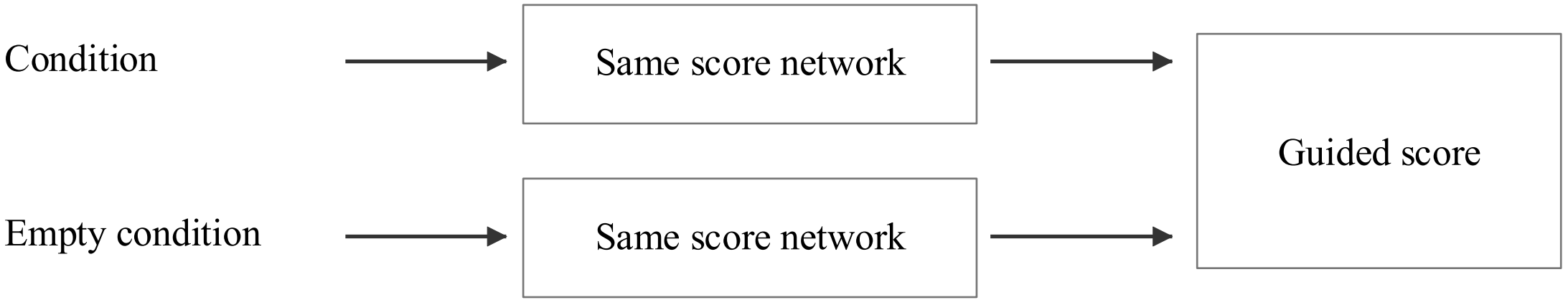
Deterministic update: reuse the estimated noise direction without fresh noise.

[12] Song, Meng, and Ermon, ICLR 2021.

Classifier-Free Guidance

A condition can be a class label or text. Drop it sometimes during training.

Both calls use the same noisy sample, time, and network parameters.



$$\widetilde{s}_\theta = s_\theta(x, \tau, \varnothing) + \omega\left[s_\theta(x, \tau, c) - s_\theta(x, \tau, \varnothing)\right]$$

Ordinary conditioning uses unit guidance. Larger scales often reduce diversity.

Two network predictions are commonly needed per sampling step.

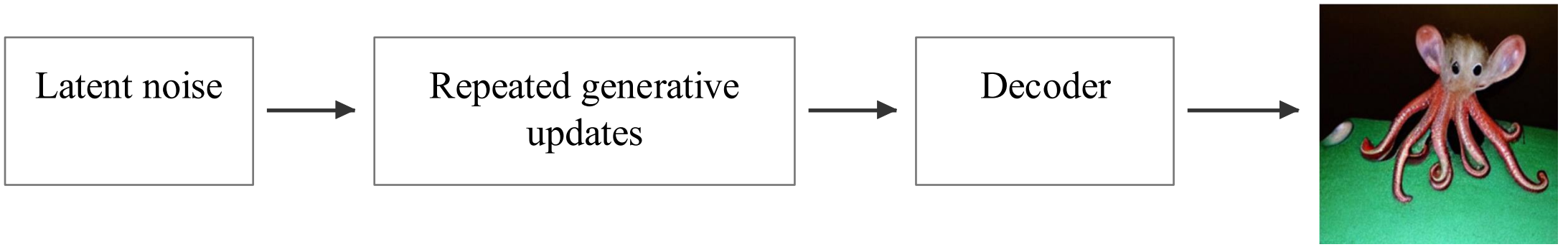
[13] Ho and Salimans, 2022.

Generation in Latent Space

Encode training images into a smaller spatial feature grid.

$$h = E(X_0), \qquad \widehat{X}_0 = D(h)$$

At generation time



The repeated model works on fewer spatial values. Decode the final latent once.

Compression reduces cost but limits recoverable detail.

[14] Rombach et al., CVPR 2022.

Main Takeaways

Flow matching

Regress velocity from simple conditional paths, then integrate an ODE.

Score-based diffusion

Learn denoising scores, then reverse a known noising process.

The connection

A diffusion score also defines a probability-flow velocity field.

What changes when a diffusion-trained model uses an ODE sampler?

[5] Lipman et al., ICLR 2023. [10] Song et al., ICLR 2021.

Questions?

Why can noisy per-pair targets learn the correct marginal velocity?

They average to the conditional mean, which transports the marginal path.

Why is the probability-flow ODE not just the reverse SDE with noise removed?

Its drift uses half the score correction to preserve the same marginals.

References

[1] DSL-Lab. Understanding Normalizing Flows. 2024. Original density-transformation illustration.

[2] Dinh, Sohl-Dickstein, and Bengio. Density Estimation Using Real NVP. ICLR 2017.

[3] Chen, Rubanova, Bettencourt, and Duvenaud. Neural Ordinary Differential Equations. NeurIPS 2018.

[4] CERN Indico. Lecture material on flows. 2024. Source of the continuous-flow density illustration.

[5] Lipman, Chen, Ben-Hamu, Nickel, and Le. Flow Matching for Generative Modeling. ICLR 2023.

[6] Lipman, Y. Flow-matching lecture slides. Source of the flow, density, and conditional-path animations in the original course material.

[7] Tong et al. Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport. TMLR 2024.

[8] Ho, Jain, and Abbeel. Denoising Diffusion Probabilistic Models. NeurIPS 2020.

[9] Song, J., Meng, C., and Vahdat, A. Denoising Diffusion Models: A Generative Learning Big Bang. CVPR 2023 tutorial. Source of original chain/noising figures.

References (continued)

[10] Song, Sohl-Dickstein, Kingma, Kumar, Ermon, and Poole. Score-Based Generative Modeling through Stochastic Differential Equations. ICLR 2021.

[11] Song, Y. Generative Modeling by Estimating Gradients of the Data Distribution. 2021. Source of score and Langevin illustrations.

[12] Song, J., Meng, C., and Ermon, S. Denoising Diffusion Implicit Models. ICLR 2021.

[13] Ho and Salimans. Classifier-Free Diffusion Guidance. 2022.

[14] Rombach, Blattmann, Lorenz, Esser, and Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.

[15] Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015.

[16] Grathwohl, Chen, Bettencourt, Sutskever, and Duvenaud. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models. ICLR 2019.

[17] Lipman et al. Flow Matching Guide and Code. 2024.

Appendix

Derivations and implementation details

A1–A6: flows and conditional flow matching

A7–A12: denoising, DDPM, and score SDEs

A13–A17: parameterizations, sampling, and variational objectives

A1. Normalizing Flows: Change of Variables

An invertible differentiable map transports a simple random variable into data space.

$$Z \sim p_0, \qquad \widehat{Y} = F_\theta(Z), \qquad F_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

$$p_\theta(y) = p_0(F_\theta^{-1}(y)) \left|\det DF_\theta^{-1}(y)\right|$$

$$\log p_\theta(y) = \log p_0(z) - \log \left|\det DF_\theta(z)\right|, \qquad z = F_\theta^{-1}(y)$$

The Jacobian corrects for volume expansion or compression. Likelihood evaluation uses the inverse map.

[15] Rezende and Mohamed, ICML 2015. [2] Dinh, Sohl-Dickstein, and Bengio, ICLR 2017.

A2. Probability Conservation

For any fixed region, mass can change only through its boundary. The normal points outward.

$$\frac{d}{dt}\int_R p_t(x)\,dx = -\int_{\partial R} p_t(x)v_t(x)\cdot n(x)\,dS$$

$$\int_R \left[\partial_t p_t(x) + \nabla\cdot(p_t(x)v_t(x))\right]dx = 0$$

$$\partial_t p_t(x) + \nabla\cdot(p_t(x)v_t(x)) = 0$$

The last equation holds locally because the integral identity holds for every region. The time derivative fixes the spatial location.

[5] Lipman et al., ICLR 2023. [7] Tong et al., TMLR 2024.

A3. Continuous Flows and Log Density

A neural velocity field defines a continuous normalizing flow. Follow a particle and apply the continuity equation.

$$\frac{dz_t}{dt} = v_\theta(z_t, t), \qquad z_0 \sim p_0$$

$$\frac{d}{dt} \log p_t(z_t) = -\nabla \cdot v_\theta(z_t, t) = -\operatorname{tr} D_x v_\theta(z_t, t)$$

$$\log p_1(z_1) = \log p_0(z_0) - \int_0^1 \nabla \cdot v_\theta(z_t, t)\, dt$$

Likelihood training integrates the trajectory and its density change. Flow matching instead supplies direct regression targets during training.

[3] Chen et al., NeurIPS 2018. [16] Grathwohl et al., ICLR 2019.

A4. The Marginal Velocity Field

Differentiate the expectation of any smooth, compactly supported test function.

$$X_t = (1-t)Z + tY, \qquad U = Y - Z, \qquad u_t(x) = \mathbb{E}[U \mid X_t = x]$$

$$\begin{aligned}
\frac{d}{dt}\int \varphi(x)p_t(x)\,dx &= \mathbb{E}[\nabla\varphi(X_t)\cdot U] \\
&= \int \nabla\varphi(x)\cdot u_t(x)p_t(x)\,dx \\
&= -\int \varphi(x)\nabla\cdot(p_t(x)u_t(x))\,dx
\end{aligned}$$

$$\partial_t p_t(x) + \nabla\cdot(p_t(x)u_t(x)) = 0$$

The identity holds for every test function, giving the continuity equation. A unique, well-defined ODE flow then reproduces these marginals.

[5] Lipman et al., ICLR 2023. [7] Tong et al., TMLR 2024.

A5. The Conditional Regression Identity

Evaluate all fields at the sampled state and time. The target residual has zero conditional mean.

$$u_t(X_t) = \mathbb{E}[U \mid X_t, t]$$

$$\begin{aligned}\mathbb{E}[(v_\theta - u_t) \cdot (U - u_t)] &= \mathbb{E}[(v_\theta - u_t) \cdot \mathbb{E}[U - u_t \mid X_t, t]] \\ &= 0\end{aligned}$$

$$\begin{aligned}\mathbb{E}\|v_\theta(X_t, t) - U\|^2 &= \mathbb{E}\|v_\theta(X_t, t) - u_t(X_t)\|^2 + C, \\ C &= \mathbb{E}\|U - u_t(X_t)\|^2.\end{aligned}$$

$$\nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta)$$

The fixed path distribution makes the residual constant independent of network parameters. Equality concerns expected gradients.

[5] Lipman et al., ICLR 2023. [7] Tong et al., TMLR 2024.

A6. General Gaussian Conditional Flow Matching

Condition on a clean sample. Choose a differentiable mean and a positive scalar standard deviation.

$$X_t = \mu_t(Y) + \sigma_t(Y)Z, \qquad p_t(x \mid Y) = \mathcal{N}(x; \mu_t(Y), \sigma_t(Y)^2 I)$$

$$\dot{X}_t = \dot{\mu}_t(Y) + \dot{\sigma}_t(Y)Z$$

$$u_t(x \mid Y) = \dot{\mu}_t(Y) + \frac{\dot{\sigma}_t(Y)}{\sigma_t(Y)}\big(x - \mu_t(Y)\big)$$

Substitute the standardized residual for the noise. Both path parameters depend on the conditioning sample, not the evaluation location.

[5] Lipman et al., ICLR 2023.

A7. Denoising and the Marginal Score

Use forward diffusion time. The noisy marginal is a mixture over unknown clean samples.

$$X_\tau = a_\tau X_0 + \sigma_\tau Z, \qquad q_\tau(u) = \int q_\tau(u \mid x_0) p_{\text{data}}(x_0)\, dx_0$$

$$\begin{aligned} s_\tau(u) &= \nabla_u \log q_\tau(u) = \mathbb{E}[\nabla_u \log q_\tau(u \mid X_0) \mid X_\tau = u], \\ &= \mathbb{E}\left[-\frac{Z}{\sigma_\tau} \mid X_\tau = u \right]. \end{aligned}$$

$$\mathcal{L}_{\text{DSM}}(\theta) = \mathbb{E}\left[\lambda(\tau) \left\| s_\theta(X_\tau, \tau) + \frac{Z}{\sigma_\tau} \right\|^2 \right]$$

A noisy conditional score is a valid regression target for the marginal score. The clean endpoint is needed only to construct training examples.

[10] Song et al., ICLR 2021. [8] Ho et al., NeurIPS 2020.

A8. DDPM: The Tractable Forward Posterior

Forward step indices increase toward noise. Conditioning on the clean sample makes the one-step posterior Gaussian.

$$\alpha_k = 1 - \beta_k, \quad \bar{\alpha}_k = \prod_{j=1}^{k} \alpha_j, \quad X_k = \sqrt{\bar{\alpha}_k} X_0 + \sqrt{1 - \bar{\alpha}_k} Z$$

$$q(X_{k-1} \mid X_k, X_0) = \mathcal{N}(X_{k-1}; \widetilde{\mu}_k(X_k, X_0), \widetilde{\beta}_k I)$$

$$\widetilde{\mu}_k(u, x_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1 - \bar{\alpha}_k} x_0 + \frac{\sqrt{\alpha_k}(1 - \bar{\alpha}_{k-1})}{1 - \bar{\alpha}_k} u,$$
$$\widetilde{\beta}_k = \frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \beta_k.$$

Generation does not know the clean sample. A learned reverse transition approximates the posterior after that sample is marginalized out.

[8] Ho et al., NeurIPS 2020.

A9. DDPM: Likelihood Terms and Noise Regression

For fixed reverse variance, predicting noise parameterizes the learned Gaussian mean.

$$p_\theta(X_{k-1} \mid X_k) = \mathcal{N}(X_{k-1}; \mu_\theta(X_k, k), c_k^2 I),$$

$$\mu_\theta(u, k) = \frac{1}{\sqrt{\alpha_k}} \left(u - \frac{\beta_k}{\sqrt{1 - \bar{\alpha}_k}} \epsilon_\theta(u, k) \right).$$

$$\ell_k(\theta) = w_k \, \mathbb{E} \| Z - \epsilon_\theta(X_k, k) \|^2 + C_k,$$

$$w_k = \frac{\beta_k^2}{2c_k^2 \alpha_k (1 - \bar{\alpha}_k)}.$$

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{k, X_0, Z} \| Z - \epsilon_\theta(X_k, k) \|^2$$

The simplified objective drops the likelihood-derived weights. It changes the emphasis across noise levels.

[8] Ho et al., NeurIPS 2020.

A10. The Variance-Preserving SDE

In forward time, linear drift reduces the clean signal while Brownian noise supplies the remaining variance.

$$dX_\tau = -\frac{1}{2}\beta(\tau)X_\tau\,d\tau + \sqrt{\beta(\tau)}\,dW_\tau, \qquad X_0 \sim p_{\text{data}}$$

$$a_\tau = \exp\left(-\frac{1}{2}\int_0^\tau \beta(r)\,dr\right), \qquad \sigma_\tau^2 = 1 - a_\tau^2$$

$$X_\tau \mid X_0 \sim \mathcal{N}(a_\tau X_0, \sigma_\tau^2 I)$$

This gives direct noisy training samples at any time. A finite terminal schedule usually gives approximately Gaussian noise.

[10] Song et al., ICLR 2021.

A11. The Reverse-Time Clock

Assume isotropic diffusion with a scalar noise coefficient depending only on time. Define the forward noisy marginal score.

$$dX_\tau = f_\tau(X_\tau)\,d\tau + g_\tau\,dW_\tau, \qquad s_\tau(x) = \nabla_x \log q_\tau(x)$$

$$t = 1 - \tau, \qquad \widetilde{X}_t = X_{1-t}, \qquad p_t = q_{1-t}$$

$$d\widetilde{X}_t = \left[-f_{1-t}(\widetilde{X}_t) + g_{1-t}^2 s_{1-t}(\widetilde{X}_t)\right] dt + g_{1-t}\,dB_t$$

Here generation time increases and its increments are positive. Reversing the clock changes the drift sign and the score-correction sign together.

[10] Song et al., ICLR 2021.

A12. The Probability-Flow Half Factor

Rewrite the forward Fokker–Planck equation as a continuity equation. The scalar diffusion coefficient has no spatial dependence.

$$\partial_\tau q_\tau = -\nabla \cdot (f_\tau q_\tau) + \frac{1}{2} g_\tau^2 \Delta q_\tau$$

$$\partial_\tau q_\tau = -\nabla \cdot \left[\left(f_\tau - \frac{1}{2} g_\tau^2 s_\tau \right) q_\tau \right], \qquad q_\tau s_\tau = \nabla q_\tau$$

$$t = 1 - \tau, \qquad \frac{d\widetilde{x}_t}{dt} = -f_{1-t}(\widetilde{x}_t) + \frac{1}{2} g_{1-t}^2 s_{1-t}(\widetilde{x}_t)$$

With an exact score, this ODE and the reverse SDE share one-time marginals. Their trajectories and path distributions differ.

[10] Song et al., ICLR 2021.

A13. Gaussian Paths: Scores and Velocities

Return to generation time and a Gaussian interpolation. The score determines posterior means of both endpoints.

$$X_t = \alpha_t Y + \sigma_t Z, \qquad r_t(x) = \nabla_x \log p_t(x)$$

$$\mathbb{E}[Z \mid X_t = x] = -\sigma_t r_t(x), \qquad \mathbb{E}[Y \mid X_t = x] = \frac{x + \sigma_t^2 r_t(x)}{\alpha_t}$$

$$u_t(x) = \frac{\dot{\alpha}_t}{\alpha_t} x + \left(\frac{\dot{\alpha}_t \sigma_t^2}{\alpha_t} - \dot{\sigma}_t \sigma_t \right) r_t(x)$$

Average the conditional derivative using these posterior means. Velocity and score parameterizations are related through the chosen schedule.

[5] Lipman et al., ICLR 2023. [10] Song et al., ICLR 2021. [17] Lipman et al., 2024.

A14. DDIM: A General Reverse Step

Use the DDPM forward schedule. A sampling step moves from a noisy index to an earlier, cleaner index.

$$j < k, \qquad \widehat{Z}_k = \epsilon_\theta(X_k, k), \qquad \widehat{X}_{0,k} = \frac{X_k - \sqrt{1-\bar{\alpha}_k}\,\widehat{Z}_k}{\sqrt{\bar{\alpha}_k}}$$

$$\varsigma_{k,j} = \eta \sqrt{\frac{1-\bar{\alpha}_j}{1-\bar{\alpha}_k}} \sqrt{1 - \frac{\bar{\alpha}_k}{\bar{\alpha}_j}}, \qquad 0 \le \eta \le 1$$

$$X_j = \sqrt{\bar{\alpha}_j}\,\widehat{X}_{0,k} + \sqrt{1 - \bar{\alpha}_j - \varsigma_{k,j}^2}\,\widehat{Z}_k + \varsigma_{k,j}\xi, \qquad \xi \sim \mathcal{N}(0, I)$$

Zero stochasticity gives a deterministic update after the initial noise draw. Skipped indices reduce network evaluations.

[12] Song, Meng, and Ermon, ICLR 2021.

A15. Minibatch Optimal Transport

Keep the sampled noise and data batches. Change which endpoints are paired by minimizing their total squared displacement.

$$\pi^\star \in \arg\min_{\pi \in S_B} \sum_{i=1}^{B} \|Z_i - Y_{\pi(i)}\|^2$$

$$X_t^{(i)} = (1-t)Z_i + tY_{\pi^\star(i)}, \qquad V_t^{(i)} = Y_{\pi^\star(i)} - Z_i$$

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} \|v_\theta(X_t^{(i)}, t) - V_t^{(i)}\|^2\right]$$

A permutation preserves both batch marginals while changing their joint pairing. Finite minibatch matching is not exact global transport.

[7] Tong et al., TMLR 2024.

A16. The DDPM Variational Bound

Fix one clean observation. Average its bound over the training distribution.

$$\begin{aligned}\mathcal{E}_\theta(x_0) &:= \mathbb{E}_{q(X_{1:K}|x_0)}[\log p_\theta(x_0, X_{1:K}) - \log q(X_{1:K} \mid x_0)] \\ &\leq \log p_\theta(x_0).\end{aligned}$$

$$\begin{aligned}-\mathcal{E}_\theta(x_0) = D_{\mathrm{KL}}(q(X_K \mid x_0) \| p(X_K)) \\ + \sum_{k=2}^{K} \mathbb{E}_{q(X_k|x_0)} D_{\mathrm{KL}}(q(X_{k-1} \mid X_k, x_0) \| p_\theta(X_{k-1} \mid X_k)) \\ - \mathbb{E}_{q(X_1|x_0)} \log p_\theta(x_0 \mid X_1).\end{aligned}$$

$$\mathcal{L}_{\mathrm{VLB}}(\theta) = -\mathbb{E}_{X_0 \sim p_{\mathrm{data}}} \mathcal{E}_\theta(X_0)$$

Interior KL terms give the weighted noise MSE in Appendix A9.

[8] Ho et al., NeurIPS 2020.

A17. Clean and Noise Prediction: Loss Weights

For positive signal and noise scales, the outputs are linearly convertible.

$$X_k = a_k X_0 + \sigma_k \epsilon, \qquad a_k = \sqrt{\bar{\alpha}_k}, \quad \sigma_k = \sqrt{1 - \bar{\alpha}_k}$$

$$\widehat{X}_{0,\theta}(x, k) = \frac{x - \sigma_k \epsilon_\theta(x, k)}{a_k}, \qquad \epsilon_\theta(x, k) = \frac{x - a_k \widehat{X}_{0,\theta}(x, k)}{\sigma_k}$$

$$\|\epsilon_\theta - \epsilon\|^2 = \frac{a_k^2}{\sigma_k^2} \|\widehat{X}_{0,\theta} - X_0\|^2$$

The identity assumes the outputs obey this conversion. Unweighted clean-data and noise losses assign different importance to time steps.

[8] Ho et al., NeurIPS 2020.