Flow and Diffusion Models

Lecture 04

Qiang Sun

Roadmap

1. Flow models: ODEs, flows, simulation, sampling algorithm
2. Diffusion models: SDEs, Brownian motion, Euler–Maruyama, sampling algorithm
3. Constructing training targets: probability paths, continuity / Fokker–Planck

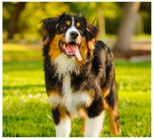
Flow and Diffusion Models

Turn noise into data

Initial distribution: $\boxed{p_{\text{init}}}$ ***Default:*** $\boxed{p_{\text{init}} = \mathcal{N}(0, I_d)}$

A generative model converts samples from a initial distribution (e.g. Gaussian) into samples from the data distribution:

$\boxed{x \sim p_{\text{init}}}$ ⇒ **Generative Model** ⇒ $\boxed{z \sim p_{\text{data}}}$



Generative modeling viewpoint

Sampling goal:

$$X_1 \sim p_{\text{data}} \quad \text{by transforming} \quad X_0 \sim p_{\text{init}} \;\; (\text{e.g. } \mathcal{N}(0, I_d)).$$

Key idea: obtain the transformation by simulating

- an **ODE** (flow models), or
- an **SDE** (diffusion models).

ODEs and vector fields

A trajectory is a map

$$X : [0,1] \to \mathbb{R}^d, \qquad t \mapsto X_t.$$

A (time-dependent) vector field:

$$u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d, \qquad (x,t) \mapsto u_t(x).$$

ODE with initial condition:

$$\frac{d}{dt} X_t = u_t(X_t), \qquad X_0 = x_0.$$

Flows

The *flow* map answers: where do we go starting from x_0?

$$\phi : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d, \qquad (x_0, t) \mapsto \phi_t(x_0),$$

defined by

$$\frac{d}{dt}\phi_t(x_0) = u_t(\phi_t(x_0)), \qquad \phi_0(x_0) = x_0.$$

Trajectory recovery: $X_t = \phi_t(X_0)$.

Existence and uniqueness

Theorem (Flow existence and uniqueness, Picard–Lindelöf Theorem)

If $u_t : \mathbb{R}^d \to \mathbb{R}^d$ is continuously differentiable with bounded derivative, then the ODE has a unique solution given by a flow ϕ_t. Moreover, ϕ_t is a diffeomorphism for all t.

Remark

Key takeaway: In the cases of practical interest for machine learning, unique solutions to ODE/flows exist.

Example: linear vector fields

Example (Linear vector fields)

Let $u_t(x) = -\theta x$ with $\theta > 0$, and the ODE be given:

$$\frac{d}{dt}\phi_t(x_0) = u_t(\phi_t(x_0)), \qquad \phi_0(x_0) = x_0.$$

Then the flow is given by:

$$\phi_t(x_0) = e^{-\theta t} x_0.$$

Simulating an ODE: Euler method

If ϕ_t is not available in closed form, simulate:

$$X_{t+h} = X_t + h\, u_t(X_t), \qquad t = 0, h, 2h, \ldots, 1-h, \qquad h = \frac{1}{n}.$$

(Heun's method is a simple higher-order alternative.)

Key Idea

A *flow model* uses a neural vector field:

$$X_0 \sim p_{\mathrm{init}}, \qquad \frac{d}{dt} X_t = u_t^\theta(X_t), \qquad u^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d.$$

Training goal (later): choose θ so that

$$X_1 \sim p_{\mathrm{data}}.$$

Sampling from a flow model with Euler method

Algorithm 1: Sampling from a flow model with Euler method.

Input: Neural network vector field u_t^θ, number of steps n

Set $t = 0$;

Set step size $h = \frac{1}{n}$;

Draw a sample $X_0 \sim p_{\text{init}}$;

for $i = 1, \ldots, n$ **do**

  $X_{t+h} \leftarrow X_t + h\, u_t^\theta(X_t)$;

  Update $t \leftarrow t + h$;

return X_1;

A toy example

Toy example

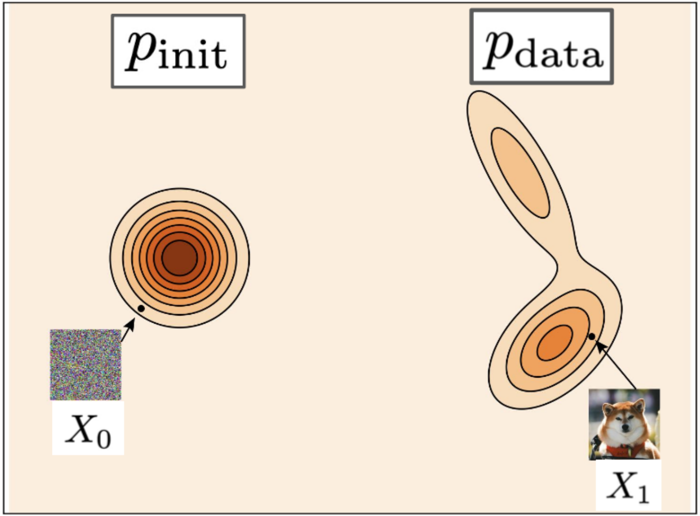


Figure credit: Yaron Lipman

SDEs and stochastic trajectories

An SDE produces a stochastic process $(X_t)_{0 \leq t \leq 1}$:

$$X_t \text{ is random for each } t.$$

Different simulations yield different sample paths.

Brownian motion

A Brownian motion W_t satisfies:

- $W_0 = 0$ and paths are continuous;
- **Normal increments:** $W_t - W_s \sim \mathcal{N}\big(0, (t-s)I_d\big)$;
- **Independent increments.**

Discrete simulation ($h = 1/n$):

$$W_{t+h} = W_t + \sqrt{h}\,\varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, I_d).$$

From ODEs to SDEs

Heuristic increment form:

$$X_{t+h} = X_t + h\, u_t(X_t) + \sigma_t \big(W_{t+h} - W_t\big) + hR_t(h).$$

Symbolic SDE notation:

$$dX_t = u_t(X_t)\, dt + \sigma_t\, dW_t, \qquad X_0 = x_0.$$

SDE existence and uniqueness

Theorem (SDE solution existence and uniqueness)

If u_t is continuously differentiable with bounded derivative and σ_t is continuous, then

$$dX_t = u_t(X_t)\,dt + \sigma_t\,dW_t, \qquad X_0 = x_0$$

has a unique solution process $(X_t)_{0 \leq t \leq 1}$.

Remark

Key takeaway: In the cases of practical interest for machine learning, unique solutions to SDEs exist.

Example: Ornstein–Uhlenbeck

Example (Ornstein–Uhlenbeck process)

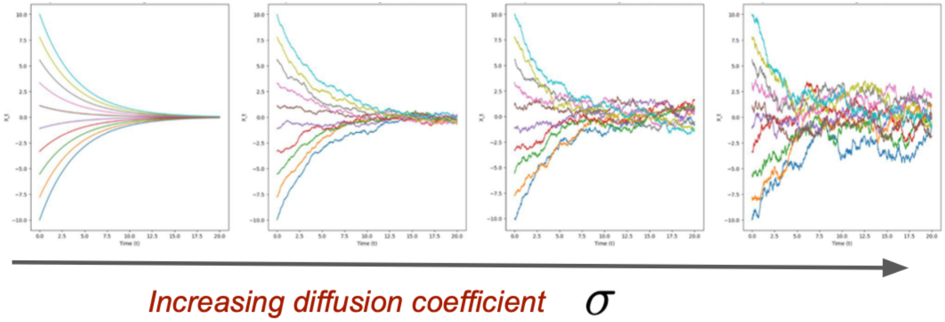
For $\theta > 0$ and constant $\sigma \geq 0$:

$$dX_t = -\theta X_t \, dt + \sigma \, dW_t.$$

Drift pulls toward 0; diffusion injects noise. For $\sigma = 0$, recovers the deterministic linear flow.

Ornstein–Uhlenbeck Processes: Sample Paths

$$\mathrm{d}X_t = -\theta X_t \mathrm{d}t + \sigma dW_t$$



Simulating an SDE: Euler–Maruyama

Euler–Maruyama update ($h = 1/n$):

$$X_{t+h} = X_t + h\, u_t(X_t) + \sigma_t \sqrt{h}\, \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, I_d).$$

Key Idea

A *diffusion model*:

$$X_0 \sim p_{\text{init}}, \qquad dX_t = u_t^\theta(X_t)\, dt + \sigma_t\, dW_t,$$

where u_t^θ is neural and σ_t is a fixed schedule. Goal: $X_1 \sim p_{\text{data}}$ after training.

Sampling from a diffusion model with Euler–Maruyama method

Algorithm 2: Sampling from a Diffusion Model with Euler–Maruyama method.

Input: Neural network u_t^θ, number of steps n, diffusion coefficient σ_t

Set $t = 0$;

Set step size $h = \frac{1}{n}$;

Draw a sample $X_0 \sim p_{\text{init}}$;

for $i = 1, \ldots, n$ **do**

- Draw a sample $\varepsilon \sim \mathcal{N}(0, I_d)$;
- $X_{t+h} \leftarrow X_t + h\, u_t^\theta(X_t) + \sigma_t \sqrt{h}\, \varepsilon$;
- Update $t \leftarrow t + h$;

return X_1;

SDE generative model summary

Summary (Diffusion vs. flow)

A diffusion model consists of:

$$u^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d, \qquad \sigma : [0,1] \to [0,\infty).$$

Sampling:

- $X_0 \sim p_{init}$
- *simulate* $dX_t = u_t^\theta(X_t)\,dt + \sigma_t\,dW_t$
- *hope (after training)* $X_1 \sim p_{data}$

If $\sigma_t \equiv 0$*, the diffusion model reduces to a flow model.*

Constructing the Training Target

Why we need a training target

With random θ, simulating gives nonsense. Train via (e.g.) MSE:

$$L(\theta) = \left\| u_t^{\theta}(x) - u_t^{\text{target}}(x) \right\|^2.$$

Two steps:

1. choose a probability path $(p_t)_{t\in[0,1]}$ from p_{init} to p_{data};
2. derive u_t^{target} that realizes this path.

Conditional and marginal probability paths

A conditional path $p_t(x \mid z)$ satisfies:

$$p_0(\cdot \mid z) = p_{\text{init}}, \qquad p_1(\cdot \mid z) = \delta_z.$$

Induced marginal path:

$$p_t(x) = \int p_t(x \mid z)\, p_{\text{data}}(z)\, dz,$$

so

$$p_0 = p_{\text{init}}, \qquad p_1 = p_{\text{data}}.$$

Gaussian conditional probability path

Example (Gaussian conditional probability path)

Schedulers α_t, β_t with $\alpha_0 = 0, \alpha_1 = 1, \beta_0 = 1, \beta_1 = 0$:

$$p_t(\cdot \mid z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d).$$

Sampling from the marginal:

$$z \sim p_{\text{data}},\ \varepsilon \sim \mathcal{N}(0, I_d) \implies x = \alpha_t z + \beta_t \varepsilon \sim p_t.$$

Marginalization trick (flow target)

Theorem (Marginalization trick)

Assume conditional drift $u_t^{target}(\cdot \mid z)$ yields

$$X_0 \sim p_{init}, \quad \frac{d}{dt}X_t = u_t^{target}(X_t \mid z) \implies X_t \sim p_t(\cdot \mid z).$$

Define

$$u_t^{target}(x) = \int u_t^{target}(x \mid z)\, \frac{p_t(x \mid z)p_{data}(z)}{p_t(x)}\, dz.$$

Then

$$X_0 \sim p_{init}, \quad \frac{d}{dt}X_t = u_t^{target}(X_t) \implies X_t \sim p_t,$$

hence $X_1 \sim p_{data}$.

Example (Target ODE for Gaussian probability paths)

For $p_t(\cdot \mid z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$ with $\dot{\alpha}_t = \partial_t \alpha_t$, $\dot{\beta}_t = \partial_t \beta_t$:

$$u_t^{\text{target}}(x \mid z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right) z + \frac{\dot{\beta}_t}{\beta_t} x.$$

Also, conditional flow map:

$$\phi_t^{\text{target}}(x \mid z) = \alpha_t z + \beta_t x.$$

Continuity equation

Define divergence:

$$\mathrm{div}(v_t)(x) = \sum_{i=1}^{d} \frac{\partial}{\partial x_i}(v_t(x))_i.$$

Theorem (Continuity equation)

An ODE with drift u_t follows $X_t \sim p_t$ iff

$$\partial_t p_t(x) = -\,\mathrm{div}\left(p_t u_t\right)(x).$$

Proof idea: marginalization trick (one-slide sketch)

- Start from $p_t(x) = \int p_t(x \mid z) p_{\text{data}}(z)\, dz$.
- Differentiate in t and use conditional continuity:

$$\partial_t p_t(x \mid z) = -\operatorname{div}\big(p_t(\cdot \mid z)\, u_t^{\text{target}}(\cdot \mid z)\big)(x).$$

- Swap integral and divergence, then factor $p_t(x)$ to obtain

$$\partial_t p_t(x) = -\operatorname{div}\big(p_t(x)\, u_t^{\text{target}}(x)\big),$$

with u_t^{target} given by Theorem 6.

SDE extension trick (diffusion target)

Define the marginal score $\nabla \log p_t(x)$.

Theorem (SDE extension trick)

For diffusion coefficient $\sigma_t \geq 0$, the SDE

$$dX_t = \left(u_t^{target}(X_t) + \frac{\sigma_t^2}{2}\nabla \log p_t(X_t)\right) dt + \sigma_t \, dW_t, \quad X_0 \sim p_{init},$$

follows the same marginal path $X_t \sim p_t$ (hence $X_1 \sim p_{data}$).

Score marginalization + Gaussian score

Score marginalization identity:

$$\nabla \log p_t(x) = \int \nabla \log p_t(x \mid z) \, \frac{p_t(x \mid z) p_{\text{data}}(z)}{p_t(x)} \, dz.$$

Example (Gaussian score)

If $p_t(x \mid z) = \mathcal{N}(x; \alpha_t z, \beta_t^2 I_d)$, then

$$\nabla \log p_t(x \mid z) = -\frac{x - \alpha_t z}{\beta_t^2}.$$

Define Laplacian:

$$\Delta w_t(x) = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2} w_t(x) = \mathrm{div}(\nabla w_t)(x).$$

Theorem (Fokker–Planck equation)

For

$$dX_t = u_t(X_t)\,dt + \sigma_t\,dW_t, \quad X_0 \sim p_{init},$$

we have $X_t \sim p_t$ *iff*

$$\partial_t p_t(x) = -\,\mathrm{div}\left(p_t u_t\right)(x) + \frac{\sigma_t^2}{2}\Delta p_t(x).$$

Proof idea: SDE extension trick (one-slide sketch)

- Start from continuity equation for the flow target:

$$\partial_t p_t = -\operatorname{div}(p_t u_t^{\text{target}}).$$

- Add/subtract $\frac{\sigma_t^2}{2}\Delta p_t$ and rewrite

$$\Delta p_t = \operatorname{div}(\nabla p_t) = \operatorname{div}\left(p_t \nabla \log p_t\right).$$

- Conclude Fokker–Planck holds with drift

$$u_t = u_t^{\text{target}} + \frac{\sigma_t^2}{2}\nabla \log p_t,$$

which proves Theorem 9.

Remark: Langevin dynamics

Remark (Langevin dynamics)

For a static target $p_t = p$ and $u_t^{\text{target}} = 0$:

$$dX_t = \frac{\sigma_t^2}{2} \nabla \log p(X_t)\, dt + \sigma_t\, dW_t.$$

Under mild conditions, p is stationary and the dynamics converge to p from broad initializations.

Final summary: training target derivation

Summary (Derivation of the training target)

Flow target:

$$u_t^{target}(x) = \int u_t^{target}(x \mid z) \, \frac{p_t(x \mid z) p_{data}(z)}{p_t(x)} \, dz.$$

Diffusion extension:

$$dX_t = \left(u_t^{target}(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right) dt + \sigma_t \, dW_t.$$

Gaussian path (key formulas):

$$p_t(x \mid z) = \mathcal{N}(x; \alpha_t z, \beta_t^2 I_d), \quad u_t^{target}(x \mid z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) z + \frac{\dot{\beta}_t}{\beta_t} x,$$

$$\nabla \log p_t(x \mid z) = -\frac{x - \alpha_t z}{\beta_t^2}.$$