Introduction to Flow Matching

서울대학교 통계학과 변희준

2026.03.23.

1. Basics

Generative Models

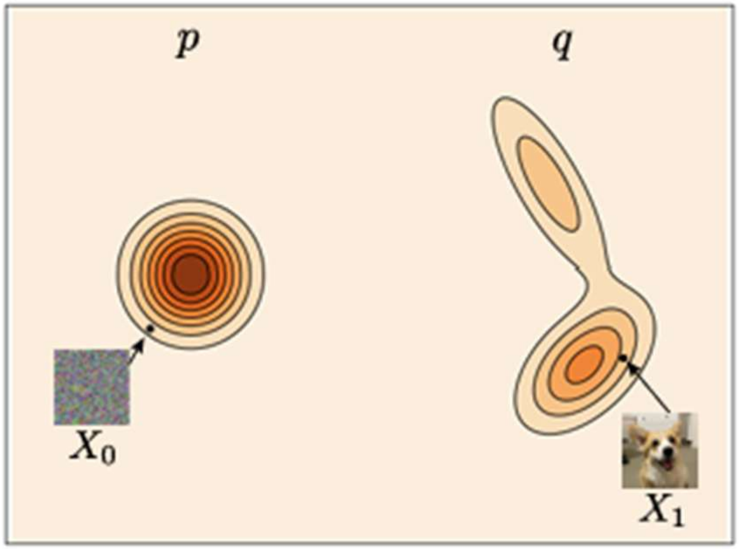
Q. What are **generative models**?

A. Algorithms that return (approximate) samples from a **target distribution** q:

- If we have partial information (e.g. $q = \frac{\pi}{Z}$; Z unknown) about the analytic form…
  - Use MCMC, neural samplers, etc.

- If we only have samples $z_1, \dots z_n \sim q$…
  - Use optimal transport maps, deep generative models (Normalizing Flows, VAEs, GANs, diffusion models, flow matching models …)

Idea 1: Sampling as Transporting



Objective: Find a map $T: \mathbb{R}^d \to \mathbb{R}^d$ such that $x \sim p \Longrightarrow T(x) \sim q$

⇒ **Monge Problem**

Q. Does such a map T always exist?

Idea 1: Sampling as Transporting

Brenier's Theorem

Let $\mu, \nu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$ be two probability measures such that μ has a density and let $X \sim \mu$. If $\bar{\gamma}$ is an optimal coupling, i.e., if

$$\int \|x-y\|^2 \bar{\gamma}(dx, dy) = \min_{\gamma \in \Gamma_{\mu,\nu}} \int \|x-y\|^2 \gamma(dx, dy) = W_2^2(\mu, \nu),$$

then there exists a convex function $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\big(X, \nabla\varphi(X)\big) \sim \bar{\gamma} \in \Gamma_{\mu,\nu}$.

e.g. if μ & ν are 1-dimensional distributions, and if μ has a density, then

$$X \sim \mu \Rightarrow \left(F_\nu^{-1} \circ F_\mu\right)(X) \sim \nu$$

Q. But how do we learn $T = \nabla\varphi$ (or some other suboptimal T)?
A. We minimize the empirical risk over samples $z_1, \ldots, z_n \sim q$!

Generative Paradigm Shift

1st Generation: Endpoint Loss (Ex) GANs, VAEs, Normalizing Flows

- **Objective**: learn a single map T so that

$$x \sim p \Rightarrow T(x) \sim q.$$

- Loss is computed only on the dataset (representatives of the final output)

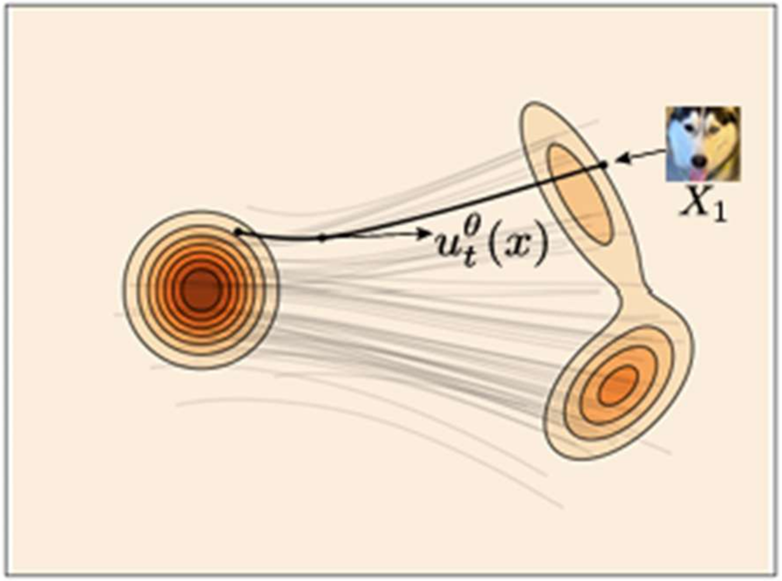
2nd Generation: Trajectory Loss (Ex) Diffusion, Flow Matching, Autoregressive

- **Objective**: learn a collection of maps $\{T_i : i \in I\}$ so that

$$x \sim p \Rightarrow T_N \circ \cdots \circ T_1(x) \sim q.$$

- Loss is also computed for the intermediate steps (for each T_i)

Idea 2: Iterative Transport via ODE



Objective: Find a vector field u: $[0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$\begin{cases} X_0 \sim p_0 \coloneqq p \\ \dot{X}_t = u_t(X_t) \end{cases} \Longrightarrow X_1 \sim p_1 \coloneqq q$$

Note that u implicitly defines a flow map $\psi_{0,1}(X_0) = \psi_{t_{N-1},1} \circ \cdots \circ \psi_{0,t_1}(X_0)$

Q. Does such a vector field u always exist?

Idea 2: Iterative Transport via ODE

Benamou-Brenier's Theorem

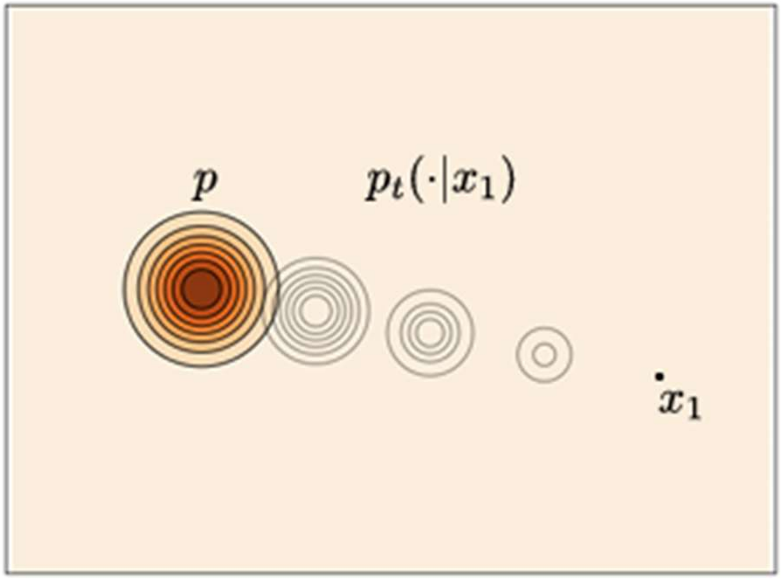
Let $\mu_0, \mu_1 \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$. Then there exists a unique optimal path $(\mu_t)_{t\in[0,1]}$ described by $X_t \sim \mu_t$, where $X_t = (1 - t)X_0 + t\, X_1$ and $(X_0, X_1) \sim \bar{\gamma} \in \Gamma_{\mu_0,\mu_1}$ with $\bar{\gamma}$ being an optimal coupling.

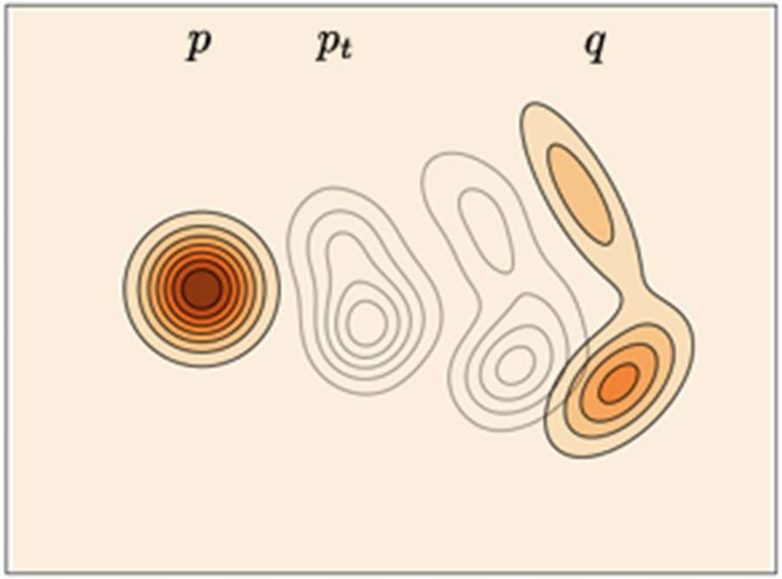
Moreover, $\underline{X_t}$ satisfies the equation $\underline{\dot{X}_t = v_t(X_t) = X_1 - X_0}$ where $v_t := (\nabla\varphi - \mathrm{id}) \circ \nabla\varphi_t^{-1},\ \nabla\varphi_t(x) := (1 - t)x + t\nabla\varphi(x),$ and $\nabla\varphi(X_0) = X_1$.

Q. But how do we learn or even obtain targets for v_t?

A. We can easily represent v_t given knowledge of the destination (X_1)!

Idea 3: Conditional Paths

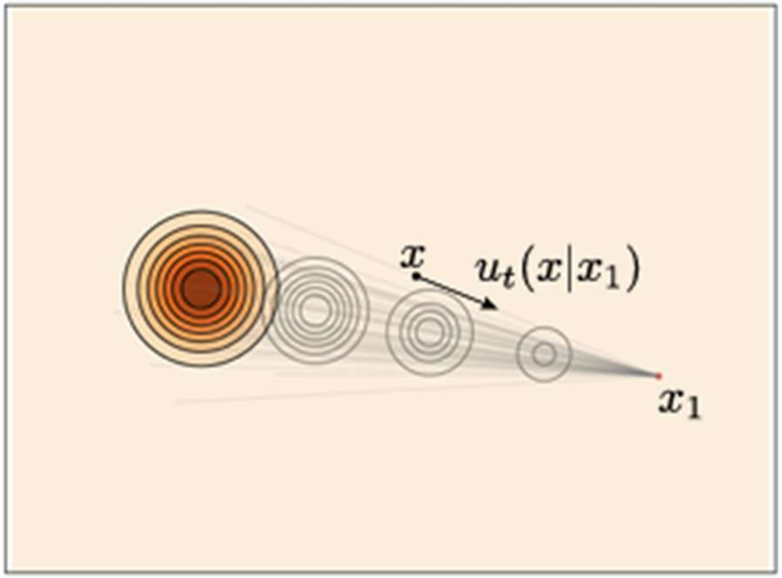


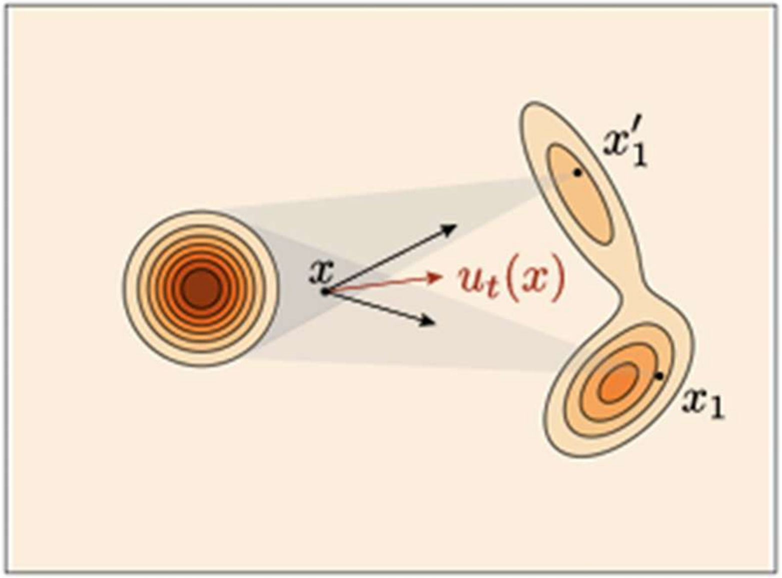


- Define a **conditional probability path** $p_t(\cdot\,|x_1)$ such that
  - $p_0(x|x_1) \equiv p(x)$
  - $p_1(x|x_1) = \delta_{x_1}(x)$

- Define the **marginal probability path** $p_t(x) = \int p_t(x|x_1)q(x_1)\,dx_1$. Then
  - $p_0(x) = \int p(x)q(x_1)\,dx_1 = p(x)$
  - $p_1(x) = \int \delta_{x_1}(x)q(x_1)\,dx_1 = q(x)$

Idea 3: Conditional Paths





- Suppose that a **conditional vector field** $u_t(\cdot\,|x_1)$ **generates** the probability path $p_t(\cdot\,|x_1)$, i.e.,
  - $X_0 \sim p_0(\cdot\,|x_1),\ \dot{X}_t = u_t(X_t|x_1) \implies X_t \sim p_t(\cdot\,|x_1)$

- Then the **marginal vector field**

$$u_t(x) \coloneqq \int u_t(x|x_1) \frac{p_t(x|x_1)q(x_1)}{p_t(x)}\, dx_1 = \mathbb{E}[u_t(x|X_1)|X_t = x]$$

generates the probability path $p_t(x) = \int p_t(x|x_1)q(x_1)\, dx_1$.
  - Why?

Continuity Equation

Suppose that $X_0 \sim \mu_0$, and that $(X_t)_{t\geq 0}$ evolves according to the dynamics $\dot{X}_t = v_t(X_t)$ where v_t is Lipschitz continuous in x, uniformly in t, and is also uniformly bounded. Let μ_t denote the law of X_t for all $t \geq 0$. Then, $(\mu_t)_{t\geq 0}$ satisfies the following **continuity equation**:

$$\partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0$$

Moreover, every solution μ_t of the same equation that admits a density for every t is necessarily obtained from the dynamics $\dot{X}_t = v_t(X_t)$ (i.e., generated by v_t).

Proof (first part). For all compactly supported and smooth test functions $\varphi : \mathbb{R}^d \to \mathbb{R}$, it holds that

$$\int \varphi\, \partial_t \mu_t = \partial_t \int \varphi\, d\mu_t = \int \langle \nabla\varphi, v_t \rangle \mu_t = -\int \varphi\, \nabla \cdot (\mu_t v_t)$$

- It is easy to show that if $\big(u_t(\cdot\,|x_1), p_t(\cdot\,|x_1)\big)$ satisfies the continuity equation, then so does $(u_t(\cdot), p_t(\cdot))$.

Example: Gaussian Probability Paths

Let $\alpha_t, \beta_t \in \mathbb{R}$ be continuously differentiable, monotonic functions that we choose such that $\alpha_0 = \beta_1 = 0$ & $\alpha_1 = \beta_0 = 1$ (e.g. $\alpha_t = t, \beta_t = 1 - t$).

We can easily check that the **Gaussian probability path**

$$p_t(x|x_1) = \mathcal{N}\left(x; \alpha_t x_1, \beta_t^2 I_d\right) \Leftrightarrow x = \alpha_t x_1 + \beta_t \varepsilon, \varepsilon \sim \mathcal{N}(0, I_d)$$

satisfies the boundary conditions required for a conditional probability path.

The corresponding conditional vector field is

$$u_t(x|x_1) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) x_1 + \frac{\dot{\beta}_t}{\beta_t} x$$

(e.g. $x = t x_1 + (1 - t)\varepsilon;\ u_t(x|x_1) = \frac{x_1 - x}{1 - t}$)

Idea 4: Conditional Flow Matching Loss

Objective: learn $u_t^\theta \approx u_t$.

- A natural approach would be to try minimizing the MSE:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\pi, x\sim p_t(\cdot|x_1)}\left[\left\|u_t^\theta(x) - u_t(x)\right\|^2\right]$$

: intractable because u_t is unknown.

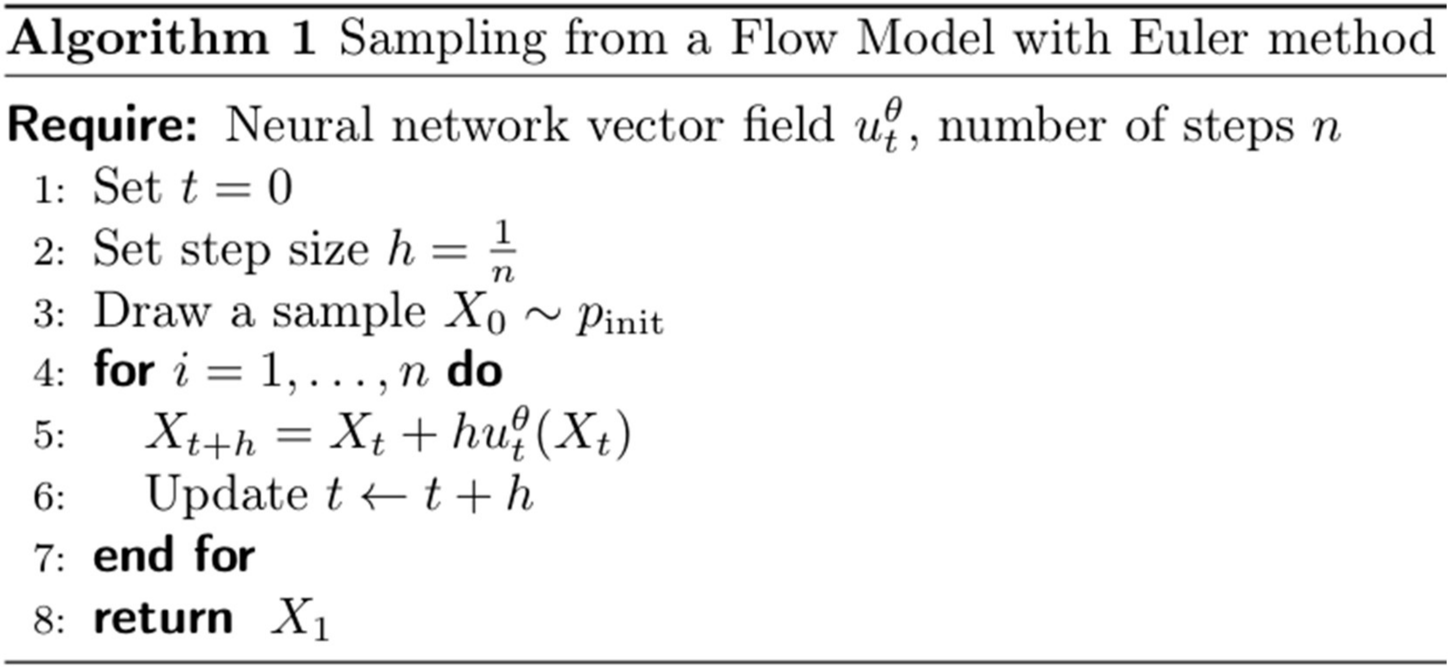
- The following tractable **conditional flow matching loss** is equivalent to the above loss up to a constant:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[\left\|u_t^\theta(x) - u_t(x|x_1)\right\|^2\right]$$

The equivalence originates from the following identity:

$$\mathbb{E}\left[\left\|u_t^\theta(x) - u_t(x|x_1)\right\|^2\right] = \mathbb{E}\left[\left\|u_t^\theta(x) - u_t(x)\right\|^2\right] + \mathbb{E}[\|u_t(x) - u_t(x|x_1)\|^2]$$

Sampling via Flow Matching



Algorithm 1 Sampling from a Flow Model with Euler method

```
Require: Neural network vector field $u_t^\theta$, number of steps $n$
Set $t = 0$
Set step size $h = \frac{1}{n}$
Draw a sample $X_0 \sim p_{\text{init}}$
for $i = 1, \dots, n$ do
    $X_{t+h} = X_t + h u_t^\theta(X_t)$
    Update $t \leftarrow t + h$
end for
return $X_1$
```

2. Advanced

Alternative Parametrizations

Recall the Gaussian probability path

$$x = \alpha_t x_1 + \beta_t \varepsilon, \varepsilon \sim \mathcal{N}(0, I_d)$$

with the conditional vector field

$$u_t(x|x_1) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) x_1 + \frac{\dot{\beta}_t}{\beta_t} x$$

Then the marginal vector field can be represented as

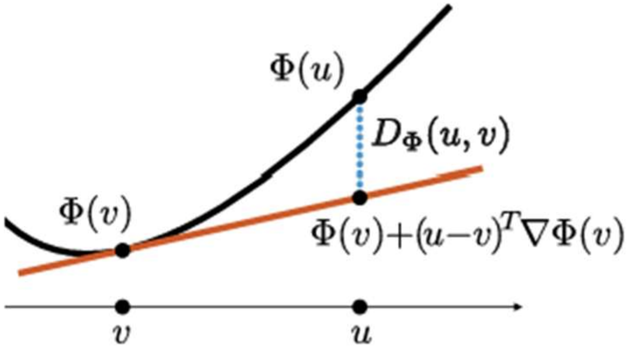
$$u_t(x) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) \mathbb{E}[X_1 | X_t = x] + \frac{\dot{\beta}_t}{\beta_t} x = \frac{\dot{\alpha}_t}{\alpha_t} x + \left(\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t} \beta_t \right) \mathbb{E}[\varepsilon | X_t = x]$$

Which can be approximated via different parametrizations:

- **Velocity(v)-prediction**: $u_t^\theta(x) \approx u_t(x)$ / Loss: $\mathbb{E}\left[\left\| u_t^\theta(x) - u_t(x|x_1) \right\|^2\right]$
- **Data(x)-prediction**: $x_{1|t}^\theta(x) \approx \mathbb{E}[X_1 | X_t = x]$ / Loss: $\mathbb{E}\left[\left\| x_{1|t}^\theta(x) - x_1 \right\|^2\right]$
- **Noise(ε)-prediction**: $\varepsilon_t^\theta(x) \approx \mathbb{E}[\varepsilon | X_t = x]$ / Loss: $\mathbb{E}\left[\left\| \varepsilon_t^\theta(x) - \varepsilon \right\|^2\right]$

Alternative Loss Functions

Bregman Divergence: $D_\Phi(u, v) \coloneqq \Phi(u) - [\Phi(v) + \langle \nabla\Phi(v), u - v \rangle]$
Where $\Phi: \mathbb{R}^d \to \mathbb{R}$ is a strictly convex function defined on a convex set $\Omega \subset \mathbb{R}^d$



- Note that $\nabla_v D_\Phi(u, v) = -(\nabla^2\Phi(v))(u - v)$ is affine in u.
- Therefore, the gradient of

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\pi,\, x\sim p_t(\cdot)}\left[D_\Phi\left(u_t(x), u_t^\theta(x)\right)\right]$$

is equal to the gradient of the **conditional flow matching loss**

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[D_\Phi\left(u_t(x|x_1), u_t^\theta(x)\right)\right]$$

Alternative Samplers

- There is also an equation for SDEs (known as the **Fokker-Planck equation**) that is analogous to the continuity equation:

$$\begin{cases} X_0 \sim p_0 \coloneqq p \\ dX_t = v_t(X_t)dt + \sigma_t dW_t \end{cases} \Leftrightarrow \partial_t\, \mu_t \; + \nabla \cdot (\mu_t v_t) = \frac{\sigma_t^2}{2} \Delta p_t$$

- Hence, the following dynamics have the same marginals:

$$\begin{cases} \dot{X}_t = u_t(X_t) \\ dX_t = \left[u_t(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right] dt + \sigma_t dW_t \end{cases}$$

- The **score function** $\nabla \log p_t(X_t)$ can be approximated by minimizing the following **conditional score matching loss**:

$$\mathcal{L}_{\mathrm{CSM}}(\theta) = \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[D_\Phi(\nabla \log p_t(x|x_1)\,, s_t^\theta(x)) \right]$$

- Here, we used the fact that $\nabla \log p_t(x) = \int \nabla \log p_t(x|x_1) \frac{p_t(x|x_1) q(x_1)}{p_t(x)}\, dx_1$.

SDE & Gaussian Probability Paths

Recall the Gaussian probability path

$$p_t(x|x_1) = \mathcal{N}\left(x; \alpha_t x_1, \beta_t^2 I_d\right) \Leftrightarrow x = \alpha_t x_1 + \beta_t \varepsilon, \varepsilon \sim \mathcal{N}(0, I_d)$$

With the corresponding conditional vector field being

$$u_t(x|x_1) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t\right)x_1 + \frac{\dot{\beta}_t}{\beta_t}x$$

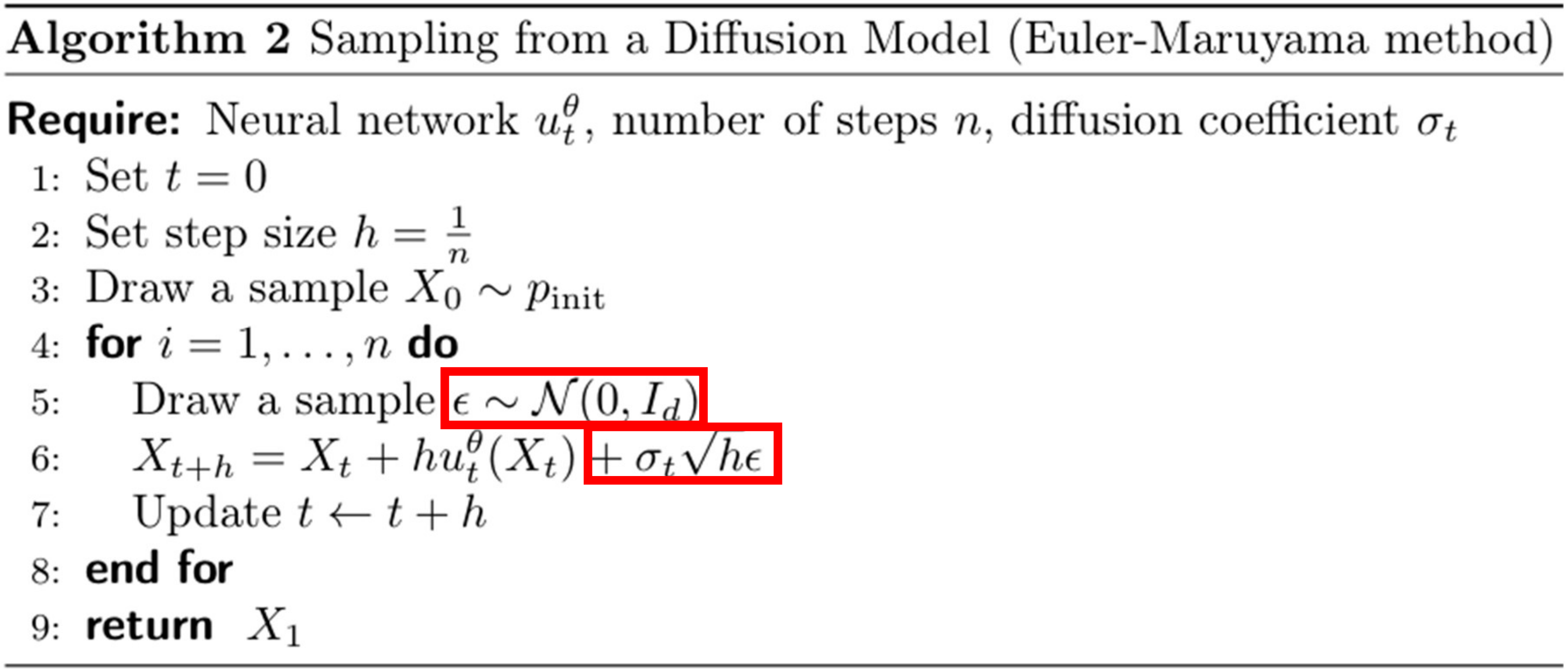
The **conditional score** can also be easily calculated:

$$\nabla \log p_t(x|x_1) = -\frac{x - \alpha_t x_1}{\beta_t^2} = \frac{\alpha_t u_t(x|x_1) - \dot{\alpha}_t x}{\left(\dot{\alpha}_t \beta_t - \alpha_t \dot{\beta}_t\right)\beta_t}$$

Thus, we may approximate $\nabla \log p_t(x) \approx \frac{\alpha_t u_t^\theta(x) - \dot{\alpha}_t x}{(\dot{\alpha}_t \beta_t - \alpha_t \dot{\beta}_t)\beta_t}$ without having to train a separate score approximator.

- Note that the above equation blows up as $t \to 1$ particularly because $\beta_t \to 0$. Therefore it is recommended to let $\sigma_t^2 = \left(\dot{\alpha}_t \beta_t - \alpha_t \dot{\beta}_t\right)\beta_t \gamma_t$ where γ_t is well-defined on $[0,1]$.

Sampling via SDEs



Algorithm 2 Sampling from a Diffusion Model (Euler-Maruyama method)

```
Require: Neural network $u_t^\theta$, number of steps $n$, diffusion coefficient $\sigma_t$
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 $\boxed{\epsilon \sim \mathcal{N}(0, I_d)}$
    $X_{t+h} = X_t + hu_t^\theta(X_t) \boxed{+ \sigma_t \sqrt{h}\epsilon}$
    Update $t \leftarrow t + h$
end for
return $X_1$
```

The main difference with Langevin MCMC or classical diffusion models is that they use a time convention ranging from $\underline{0 \text{ to } \infty}$ (as opposed to ranging from $\underline{0 \text{ to } 1}$). We may convert between the two using time reparameterizations.

Memorization & Overfitting

Let $p_0 = p \sim \mathcal{N}(0, I_d)$, and suppose $p_1 = q$ is an empirical distribution over a set of points $\{y^i : 1 \leq i \leq N\} \subset \mathbb{R}^d$, given by

$$q \sim \frac{1}{N}\sum_{i=1}^{N} \delta_{y^i}$$

Assume that we use a Gaussian probability path, which implies that the conditional velocity field is given as

$$u_t(x|x_1) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t}\alpha_t \right) x_1 + \frac{\dot{\beta}_t}{\beta_t} x.$$

Then the **optimal velocity field** $u_t^*(x)$ that minimizes the CFM loss is given by

$$u_t^*(x) = \sum_{i=1}^{N} u_t(x|y^i) \frac{\exp\left(-\frac{\left\|x - \alpha_t y^i\right\|^2}{2\beta_t^2}\right)}{\sum_{j=1}^{N} \exp\left(-\frac{\|x - \alpha_t y^j\|^2}{2\beta_t^2}\right)}$$

Which induces the marginal probability path

$$p_t(x) = \frac{1}{N}\sum_{i=1}^{N} p_t(x|y^i) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}\left(x; \alpha_t y^i, \beta_t^2 I_d\right)$$

Mitigating Memorization

- Use a simpler/regularized model
- Use Sde samplers
- Use a conditional probability path with a relaxed boundary condition, e.g.,

$$p_1(x|x_1) = \mathcal{N}(x_1, \sigma_{\min}^2 I_d)$$

The corresponding marginal probability path is

$$p_t(x) = \int \mathcal{N}(x_1, \sigma_{\min}^2 I_d) q(x_1)\, dx_1 = q * \mathcal{N}(0, \sigma_{\min}^2 I_d)$$

Which is analogous to a KDE estimator if q is an empirical distribution.

Variational Flow Matching

Recall that

$$u_t(x) = \mathbb{E}[u_t(x|x_1)|x_t = x] = \mathbb{E}_{p_{1|t}(x_1|x)}[u_t(x|x_1)]$$

This formulation suggests the approximation

$$u_t^\theta(x) = \mathbb{E}_{p^\theta_{1|t}(x_1|x)}[u_t(x|x_1)] \approx u_t(x)$$

Which can be learned by minimizing the KL divergence:

$$\begin{aligned}\mathcal{L}_{\mathrm{VF}}\quad(\theta) &= \mathbb{E}_{t\sim\pi}\left[D_{KL}\left(p_{1,t}(x_1,x_t)\middle\|p^\theta_{1,t}(x_1,x_t)\right)\right]\\ &= -\mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[\log p^\theta_{1|t}(x_1|x)\right] + \mathrm{const}\end{aligned}$$

If $u_t(x|x_1)$ is affine in x_1, we may write $u_t^\theta(x) = u_t\left(x\middle|\mathbb{E}_{p^\theta_{1|t}(x_1|x)}[x_1]\right)$. Thus, we may use a **mean-field** parametrization without loss of generality:

$$p^\theta_{1|t}(x_1|x) = \prod_{i=1}^{d} p^\theta_{1|t}\left(x_1^i|x\right)$$

Variational Flow Matching

Possible choices for $p^{\theta}_{1|t}(x^i_1|x)$ include:

- Categorical Distributions: $p^{\theta}_{1|t}(x^i_1|x) = Cat\left(x^i_1; \pi^{\theta,i}_t(x)\right)$
- Exponential Families: $p^{\theta}_{1|t}(x^i_1|x) = h(x^i_1)\exp\left(\eta^{\theta,i}_t(x)T(x^i_1) - A\left(\eta^{\theta,i}_t(x)\right)\right)$
- Gaussian Mixtures:

$$p^{\theta}_{1|t}(x^i_1|x) = \sum_{k=1}^{K} A^{\theta,i}_{t,k}(x)\mathcal{N}\left(x^i_1; \mu^{\theta,i}_{t,k}(x), \left[\sigma^{\theta,i}_{t,k}(x)\right]^2\right)$$

- Gaussian Mixtures (w/o mean field approximation):

$$p^{\theta}_{1|t}(x_1|x) = \sum_{k=1}^{K} A^{\theta}_{t,k}(x)\mathcal{N}\left(x_1; \mu^{\theta}_{t,k}(x), \Sigma^{\theta}_{t,k}(x)\right)$$

Although only x-prediction ($x^{\theta}_{1|t}(x) \approx \mathbb{E}[X_1|X_t = x]$) is introduced here, one can easily extend the VFM framework to v-prediction ($u^{\theta}_t(x) \approx u_t(x)$) or ε-prediction ($\varepsilon^{\theta}_t(x) \approx \mathbb{E}[\varepsilon|X_t = x]$)

Conditional Generation

In order to condition the generation on some additional information $y \in \mathcal{Y}$, one may simply extend the approximator from

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

to

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

And, assuming that the conditional probability path is independent of y, minimize the **guided conditional flow matching loss:**

$$\mathcal{L}_{\text{CFM}}^{\text{guided}}(\theta) = \mathbb{E}_{t\sim\pi,\,(x_1,\boldsymbol{y})\sim q,x\sim p_t(\cdot|x_1)}\left[D_\Phi(u_t(x|x_1), u_t^\theta(x|\boldsymbol{y}))\right]$$

Classifier Free Guidance (CFG)

It was soon empirically realized that generating high dimensional samples (e.g. images) with this procedure did not fit well enough to the desired label y.

A heuristic that works well is to artificially reinforce the condition y:
We substitute $\nabla \log p_t(x|y) = \nabla \log p_t(x) + \nabla \log p_t(y|x)$with

$$\widetilde{s}_t(x|y) = \nabla \log p_t(x) + w_t \nabla \log p_t(y|x) = w_t \nabla \log p_t(x|y) + (1 - w_t) \nabla \log p_t(x)$$

Where the **guidance scale** $w_t > 1$.

If we approximate $s_t^\theta(x|y) \approx \nabla \log p_t(x|y)$, we can augment the label set $\mathcal{Y}$ with a **default label** $\emptyset$ and use the same model to approximate $s_t^\theta(x|\emptyset) \approx \nabla \log p_t(x)$.

Classifier Free Guidance (CFG)

We can also use the same construction for the flow matching vector field:

$$\widetilde{u_t}(x|y) = w_t {\color{red}u_t(x|y)} + (1 - w_t){\color{green}u_t(x)} \approx w_t u_t^\theta(x|y) + (1 - w_t)u_t^\theta(x|\emptyset) =: \widetilde{u_t^\theta}(x|y)$$

To train the approximator, we minimized the following modified loss:

$$\mathcal{L}_{\text{CFM}}^{\text{CFG}}(\theta) = \mathbb{E}\left[D_\Phi\left(u_t(x|x_1), u_t^\theta(x|\boldsymbol{y}')\right)\right]$$

Where the expectation is over $t \sim \pi, \;\; (x_1, \boldsymbol{y}) \sim {\color{red}q}, \;\; x \sim p_t(\cdot\,|x_1), \;\; y' = \begin{cases} \emptyset & w.p.\,\eta \\ y & w.p.\,(1 - \eta) \end{cases}$

After training, we may generate samples using the ODE (or an equivalent Langevin SDE) defined by the vector field

$$\widetilde{u_t^\theta}(x|y) = w_t u_t^\theta(x|y) + (1 - w_t)u_t^\theta(x|\emptyset).$$

3. Extensions

Practical Perspective

- Flow matching in latent space
  - latent flow matching
  - transition matching
- One (or few) -step samplers
  - mean flow
  - flow map matching
  - terminal velocity matching

Theoretical Perspective

- Conditional probability paths with other conditions
    - Stochastic interpolants
- Relationship with optimal transport
    - Rectified flow
- Extending to other modalities
    - Discrete flow matching
    - Diffusion language models (Masked diffusion models)
    - Generator matching

Thank you!