Introduction to Flow Matching

서울대학교 통계학과 변희준

2026.03.23.

1. Basics

Generative Models

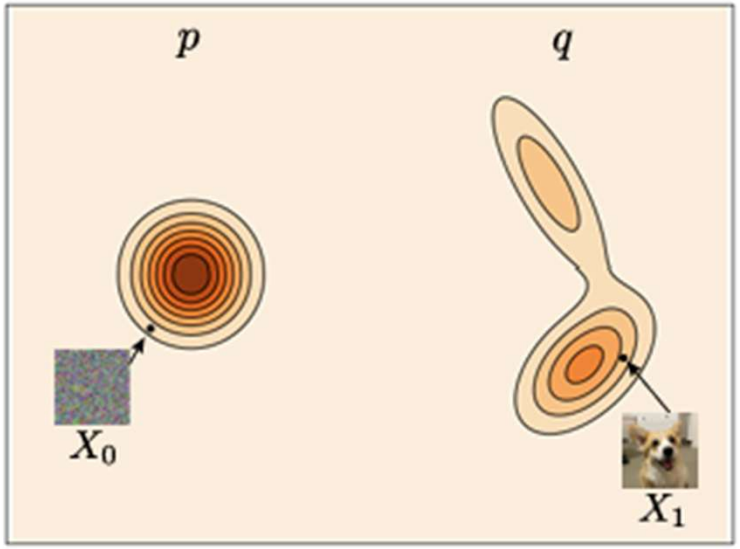
Q. What are **generative models**?

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 $z_1, \ldots z_n \sim q$…
  - 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 $x \sim p \Longrightarrow T(x) \sim q$

$\Rightarrow$ **Monge Problem**

Q. Does such a map T always exist?

Idea 1: Sampling as Transporting

Brenier's Theorem[7]

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 $\varphi: \mathbb{R}^d \to \mathbb{R}$ such that $\left(X, \nabla\varphi(X)\right) \sim \bar{\gamma} \in \Gamma_{\mu,\nu}$.

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)

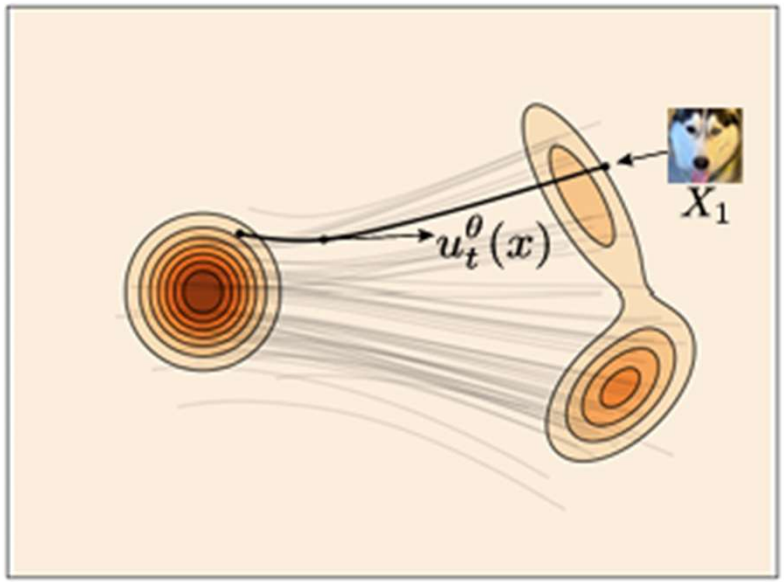
2nd Generation: Trajectory Loss (Ex) Diffusion, Flow Matching, Autoregressive

- **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[7]

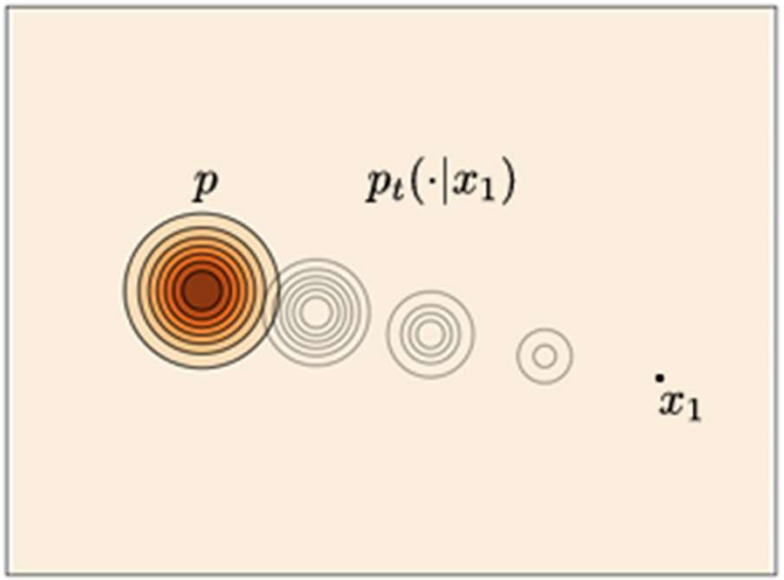
Let $\mu_0, \mu_1 \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$. Then there exists a unique optimal path $(\mu_t)_{t\in[0,1]}$ described by $X_t \sim \mu_t$, where $X_t = (1-t)X_0 + t\,X_1$ and $(X_0, X_1) \sim \bar{\gamma} \in \Gamma_{\mu_0,\mu_1}$ with $\bar{\gamma}$ being an optimal coupling.

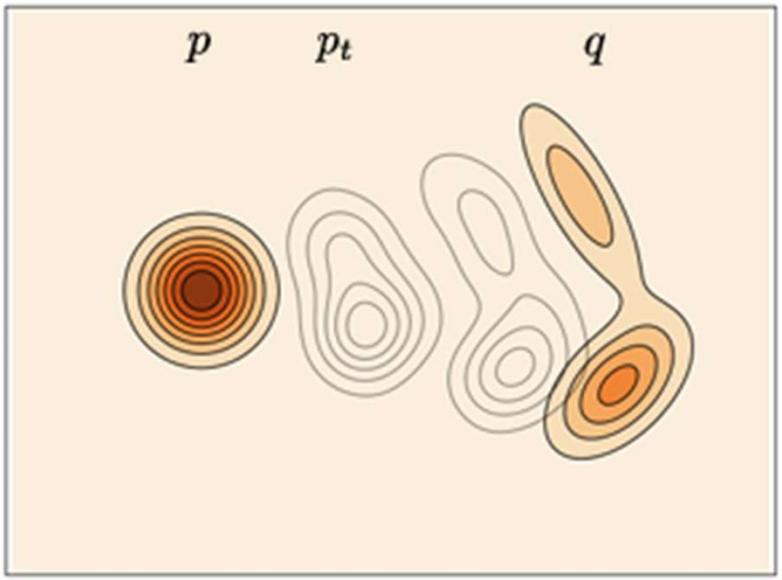
Moreover, X_t satisfies the equation $\dot{X}_t = v_t(X_t) = X_1 - X_0$ where $v_t := (\nabla\varphi - \mathrm{id}) \circ \nabla\varphi_t^{-1}$, $\nabla\varphi_t(x) := (1-t)x + t\nabla\varphi(x)$, and $\nabla\varphi(X_0) = X_1$.

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

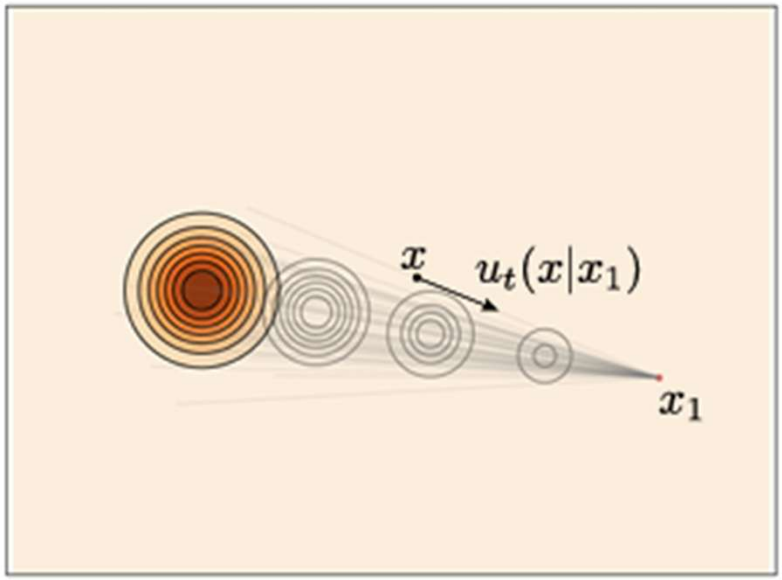


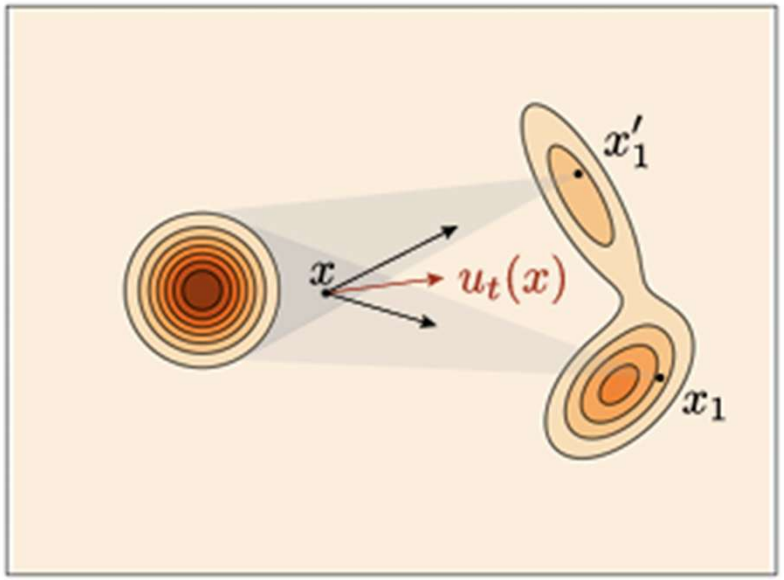


- Define a **conditional probability path** $p_t(\cdot\,|x_1)$ such that
  - $p_0(x|x_1) \equiv p(x)$
  - $p_1(x|x_1) = \delta_{x_1}(x)$

- 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).[28]

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. $x = tx_1 + (1-t)\varepsilon;\ u_t(x|x_1) = \frac{x_1 - x}{1-t}$)

Idea 4: Conditional Flow Matching Loss

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

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

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

: intractable because u_t is unknown.

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

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

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^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**

  $X_{t+h} = X_t + h u_t^\theta(X_t)$

  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

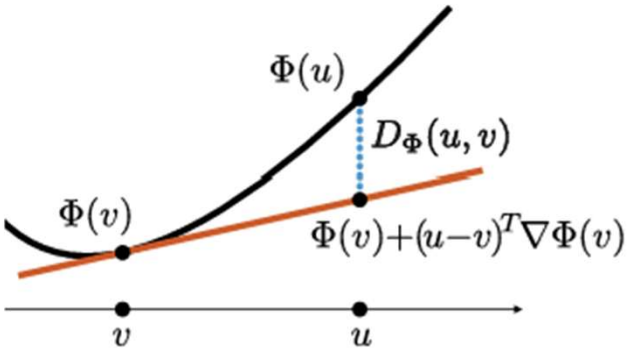
$$u_t(x) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) \mathbb{E}[X_1|X_t = x] + \frac{\dot{\beta}_t}{\beta_t} x = \frac{\dot{\alpha}_t}{\alpha_t} x + \left(\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t} \beta_t \right) \mathbb{E}[\varepsilon|X_t = x]$$

Which can be approximated via different parametrizations[20]:

- **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

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

is equal to the gradient of the **conditional flow matching loss**[14, 19, 20]

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

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**:

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

- 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^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 $\epsilon \sim \mathcal{N}(0, I_d)$
    $X_{t+h} = X_t + h u_t^\theta(X_t) + \sigma_t \sqrt{h} \epsilon$
    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 0 to ∞ (as opposed to ranging from 0 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 vector field** $u_t^*(x)$ that minimizes the CFM loss is given by[3, 13]

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

Which induces the marginal probability path

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

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 boundary condition for the marginal probability path is

$$p_1(x) = \int \mathcal{N}(x_1, \sigma_{\min}^2 I_d) q(x_1)\, dx_1 = q * \mathcal{N}(0, \sigma_{\min}^2 I_d)$$

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

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

Which can be learned by minimizing the KL divergence:

$$\mathcal{L}_{\text{VFM}}(\theta) = \mathbb{E}_{t\sim\pi}\left[D_{KL}\left(p_{1,t}(x_1, x_t)\|p_{1,t}^\theta(x_1, x_t)\right)\right]$$
$$= -\mathbb{E}_{t\sim\pi,\, x_1\sim q,\sim p_t(\cdot|x_1)}\left[\log p_{1|t}^\theta(x_1|x)\right] + \text{const}$$

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

$$p_{1|t}^\theta(x_1|x) = \prod_{i=1}^{d} p_{1|t}^\theta(x_1^i|x)$$

Variational Flow Matching

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

- Categorical Distributions[10, 23]: $p^{\theta}_{1|t}(x^i_1|x) = Cat\left(x^i_1; \pi^{\theta,i}_t(x)\right)$
- Exponential Families[17]: $p^{\theta}_{1|t}(x^i_1|x) = h(x^i_1)\exp\left(\eta^{\theta,i}_t(x)T(x^i_1) - A\left(\eta^{\theta,i}_t(x)\right)\right)$
- 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)[5]:

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

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)[12]

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)[12]

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

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

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

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

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

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[8, 27]
    - transition matching[29, 30]
- One (or few) -step samplers
    - mean flow[15, 16, 32]
    - flow map matching[4, 9]
    - terminal velocity matching[33]

Theoretical Perspective

- Conditional probability paths with other conditions
  - Stochastic interpolants[1, 2, 4]
- Relationship with optimal transport
  - Rectified flow[21, 22, 24]
- Extending to other modalities
  - Non-Euclidean flow matching[6, 9]
  - Discrete flow matching[14]
  - Diffusion language models (Masked diffusion models)[25, 26]
  - Generator matching[19, 20]

Thank you!

A. Appendix

Bregman Divergences as Likelihoods

Let's take a closer look at the Exponential Family parameterization:

$$p^{\theta}_{1|t}(x^i_1|x) = h(x^i_1)\exp\left(\eta^{\theta,i}_t(x)T(x^i_1) - A\left(\eta^{\theta,i}_t(x)\right)\right)$$

- We can derive the following identities (A^*: convex conjugate of A):

$$\mu^{\theta}_{1|t}(x^i_1|x) \coloneqq \mathbb{E}_{p^{\theta}_{1|t}(x_1|x)}[x_1] = \nabla A\left(\eta^{\theta,i}_t(x)\right)$$
$$H\left(p^{\theta}_{1|t}(x^i_1|x)\right) = -\log h(x^i_1) - A^*\left(\mu^{\theta}_{1|t}(x^i_1|x)\right)$$

- $-\mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[\log p^{\theta}_{1|t}(x_1|x)\right]$
  $= \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[-\eta^{\theta,i}_t(x)\left(T(x^i_1) - \mu^{\theta}_{1|t}(x^i_1|x)\right) - A^*\left(\mu^{\theta}_{1|t}(x^i_1|x)\right)\right] + const.$
  $= \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[D_{A^*}\left(T(x^i_1), \mu^{\theta}_{1|t}(x^i_1|x)\right)\right] + const.$
  $= \mathbb{E}_{t\sim\pi,\, x_1\sim q, x\sim p_t(\cdot|x_1)}\left[D_{A^*}\left(\mu_{1|t}(x^i_1|x), \mu^{\theta}_{1|t}(x^i_1|x)\right)\right] + const.$

- The last two equations correspond to the conditional & marginal loss for x-prediction! (We can derive parallel results for v-prediction & ε-prediction)

Sample Likelihood Calculation

It is possible to simultaneously calculate the log likelihood of the generated samples given the starting point $x_0 \sim p$ by solving the following system of ODEs and concluding that $\log q(X_1) = \ell_1$:

$$\frac{d}{dt}\begin{bmatrix} X_t \\ \ell_t \end{bmatrix} = \begin{bmatrix} u_t(X_t) \\ -\nabla \cdot u_t(X_t) \end{bmatrix}, \qquad \begin{bmatrix} X_0 \\ \ell_0 \end{bmatrix} = \begin{bmatrix} x_0 \\ \log p(x_0) \end{bmatrix}$$

Alternatively, we could instead utilize our final generated samples x_1 and solve the above system of ODEs backwards in time with the initial conditions

$$\begin{bmatrix} X_1 \\ \ell_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$$

and conclude that $\log q(x_1) = \log p(X_0) - \ell_0$.

Since u_t is unknown in practice, we need to introduce the approximation $u_t(x) \approx u_t^\theta(x)$ which results in an estimate $\log p_1^\theta(X_1) \approx \log q(X_1)$.
We may also approximate $\nabla \cdot u_t^\theta(x) \approx \mathrm{tr}\left(Z^T \partial_x u_t^\theta(x) Z\right)$ (where $\mathbb{E}[Z] = 0$ and $\mathrm{Var}(Z) = I_d$), since the right-hand side is an unbiased estimator of the left-hand side (known as the **Hutchinson's trace estimator**).

References

1. Albergo, M. S., Boffi, N. M., & Vanden-Eijnden, E. (2023). Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv*. https://doi.org/10.48550/arXiv.2303.08797
2. Albergo, M. S., & Vanden-Eijnden, E. (2023). Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2209.15571
3. Bertrand, Q., Gagneux, A., Massias, M., & Emonet, R. (2025). On the closed-form of flow matching: Generalization does not arise from target stochasticity. *arXiv*. https://doi.org/10.48550/arXiv.2506.03719
4. Boffi, N. M., Albergo, M. S., & Vanden-Eijnden, E. (2025). Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*. https://doi.org/10.48550/arXiv.2406.07507
5. Chen, H., Zhang, K., Tan, H., Xu, Z., Luan, F., Guibas, L., Wetzstein, G., & Bi, S. (2025). Gaussian mixture flow matching models. In *Proceedings of the Forty-Second International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2504.05304
6. Chen, R. T. Q., & Lipman, Y. (2024). Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2302.03660
7. Chewi, S., Niles-Weed, J., & Rigollet, P. (2025). *Statistical optimal transport*. Springer. https://doi.org/10.1007/978-3-031-85160-5
8. Dao, Q., Phung, H., Nguyen, B., & Tran, A. (2023). Flow matching in latent space. *arXiv*. https://doi.org/10.48550/arXiv.2307.08698
9. Davis, O., Albergo, M. S., Boffi, N. M., Bronstein, M. M., & Bose, A. J. (2025). Generalised flow maps for few-step generative modelling on Riemannian manifolds. *arXiv*. https://arxiv.org/abs/2510.21608
10. Eijkelboom, F., Bartosh, G., Naesseth, C. A., Welling, M., & van de Meent, J.-W. (2024). Variational flow matching for graph generation. *Advances in Neural Information Processing Systems*, *37*. https://doi.org/10.48550/arXiv.2406.04843

References

11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., & Rombach, R. (2024). Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the Forty-First International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2403.03206
12. Feng, R., Wu, T., Yu, C., Deng, W., & Hu, P. (2025). On the guidance of flow matching. *arXiv*. https://doi.org/10.48550/arXiv.2502.02150
13. Gao, W., & Li, M. (2024). How do flow matching models memorize and generalize in sample data subspaces? *arXiv*. https://doi.org/10.48550/arXiv.2410.23594
14. Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T. Q., Synnaeve, G., Adi, Y., & Lipman, Y. (2024). Discrete flow matching. *Advances in Neural Information Processing Systems*, *37*. https://doi.org/10.48550/arXiv.2407.15595
15. Geng, Z., Deng, M., Bai, X., Kolter, J. Z., & He, K. (2025). Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, *38*. https://doi.org/10.48550/arXiv.2505.13447
16. Geng, Z., Lu, Y., Wu, Z., Shechtman, E., Kolter, J. Z., & He, K. (2025). Improved mean flows: On the challenges of fastforward generative models. *arXiv*. https://doi.org/10.48550/arXiv.2512.02012
17. Guzmán-Cordero, A., Eijkelboom, F., & van de Meent, J.-W. (2025). Exponential family variational flow matching for tabular data generation. In *Proceedings of the Forty-Second International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2506.05940
18. Holderrieth, P., & Erives, E. (2025). An introduction to flow matching and diffusion models. *arXiv*. https://doi.org/10.48550/arXiv.2506.02070
19. Holderrieth, P., Havasi, M., Yim, J., Shaul, N., Gat, I., Jaakkola, T., Karrer, B., Chen, R. T. Q., & Lipman, Y. (2025). Generator matching: Generative modeling with arbitrary Markov processes. In *The Thirteenth International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2410.20587
20. Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T. Q., Lopez-Paz, D., Ben-Hamu, H., & Gat, I. (2024). Flow matching guide and code. *arXiv*. https://doi.org/10.48550/arXiv.2412.06264

References

21. Liu, X. (2022). Rectified flow: A marginal preserving approach to optimal transport. *arXiv*. https://doi.org/10.48550/arXiv.2209.14577
22. Liu, X., Gong, C., & Liu, Q. (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2209.03003
23. Matișan, R.-A., Hu, V. T., Bartosh, G., Ommer, B., Snoek, C. G. M., Welling, M., van de Meent, J.-W., Derakhshani, M. M., & Eijkelboom, F. (2025). Purrception: Variational flow matching for vector-quantized image generation. *arXiv*. https://doi.org/10.48550/arXiv.2510.01478
24. Mena, G., Kuchibhotla, A. K., & Wasserman, L. (2025). Statistical properties of rectified flow. *arXiv*. https://doi.org/10.48550/arXiv.2511.03193
25. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., & Li, C. (2025). Large language diffusion models. *arXiv*. https://doi.org/10.48550/arXiv.2502.09992
26. Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., & Kuleshov, V. (2024). Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, *37*. https://doi.org/10.48550/arXiv.2406.07524
27. Samaddar, A., Sun, Y., Nilsson, V., & Madireddy, S. (2025). Efficient flow matching using latent variables. *arXiv*. https://doi.org/10.48550/arXiv.2505.04486
28. Santambrogio, F. (2015). *Optimal transport for applied mathematicians: Calculus of variations, PDEs, and modeling*. Birkhäuser. https://doi.org/10.1007/978-3-319-20828-2
29. Shaul, N., Singer, U., Gat, I., & Lipman, Y. (2025). Transition matching: Scalable and flexible generative modeling. *Advances in Neural Information Processing Systems*, *38*. https://doi.org/10.48550/arXiv.2506.23589
30. Singer, U., & Lipman, Y. (2025). Exploring the design space of transition matching. *arXiv*. https://doi.org/10.48550/arXiv.2512.12465
31. Zaghen, O., Eijkelboom, F., Pouplin, A., Liu, C., Welling, M., van de Meent, J.-W., & Bekkers, E. J. (2025). Riemannian variational flow matching for material and protein design. *arXiv*. https://doi.org/10.48550/arXiv.2502.12981
32. Zhang, H., Siarohin, A., Menapace, W., Vasilkovsky, M., Tulyakov, S., Qu, Q., & Skorokhodov, I. (2025). AlphaFlow: Understanding and improving MeanFlow models. *arXiv*. https://doi.org/10.48550/arXiv.2510.20771
33. Zhou, L., Parger, M., Haque, A., & Song, J. (2025). Terminal velocity matching. *arXiv*. https://doi.org/10.48550/arXiv.2511.19797