Flow Matching for Generative Modeling

IDEA Seminar

Presenter
: Youngmin Ahn

April 15, 2026

CONTENTS

1 Continuous Normalizing Flows **2**

2 Flow Matching **4**

2.1 Method 4

2.2 Designing Conditional probability paths and vector fields 6

2.3 Experiments 9

Continuous Normalizing Flows

Let $\boldsymbol{x}^* \in \mathbb{R}^d$ be a random vector with density q.
Goal : Transform a simple distribution p_0 into a distribution p_1 that approximates q.

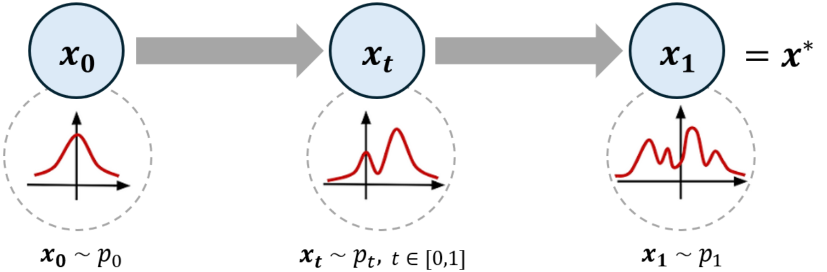

Figure. Overview of continuous normalizing flow

- Flow $\phi_t(\boldsymbol{x}) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, $\phi_0(\boldsymbol{x}_0) = \boldsymbol{x}_0$, $\phi_t(\boldsymbol{x}_0) = \boldsymbol{x}_t$
- Vector field $\boldsymbol{v}_t(\boldsymbol{x}) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, $\frac{d}{dt}\phi_t(\boldsymbol{x}) = \boldsymbol{v}_t(\phi_t(\boldsymbol{x}))$
- Model $\boldsymbol{v}_t(\boldsymbol{x})$ with a neural network $\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^p$ are learnable parameters.

Continuous Normalizing Flows

Learn $\boldsymbol{\theta}$ by minimizing the negative log-likelihood.

$$\begin{aligned}\mathcal{L}_{CNF}(\boldsymbol{\theta}) &= \mathbb{E}\left[-\log p_1(\boldsymbol{x}_1)\right] \\ &= \mathbb{E}\left[-\log p_0(\boldsymbol{x}_0) + \int_0^1 \operatorname{Tr}\left(\frac{\partial \boldsymbol{v}_t}{\partial \boldsymbol{x}}(\boldsymbol{x}_t; \boldsymbol{\theta})\right) dt\right]\end{aligned}$$

- Requires solving the following ordinary differential equation (ODE) to obtain $\boldsymbol{x}_1$ from $\boldsymbol{x}_0$:

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{v}_t(\boldsymbol{x}_t).$$

- Requires iterative computation while updating $\boldsymbol{\theta}$.
- Training is computationally expensive.

Flow Matching

Method

- Regression task by specifying a vector field $\boldsymbol{u}_t(\mathbf{x})$ and the corresponding probability path $p_t(\mathbf{x})$.

$$\mathcal{L}_{FM}(\boldsymbol{\theta}) = \mathbb{E}_{t,p_t(\boldsymbol{x})}||\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{u}_t(\boldsymbol{x})||^2, \text{ where } t \sim U[0,1].$$

- p_t and $\boldsymbol{u}_t$ can be constructed using probability paths and vector fields defined for each sample of $\boldsymbol{x}_1$.

$$p_t(\boldsymbol{x}) = \int p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) q(\boldsymbol{x}_1) d\boldsymbol{x}_1 \tag{1}$$

$$\boldsymbol{u}_t(\boldsymbol{x}) = \int \boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \frac{p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) q(\boldsymbol{x}_1)}{p_t(\boldsymbol{x})} d\boldsymbol{x}_1 \tag{2}$$

Flow Matching
Method

- Then minimizing $\mathcal{L}_{FM}$ is equal with minimizing

$$\mathcal{L}_{CFM}(\boldsymbol{\theta}) = \mathbb{E}_{t,q(\boldsymbol{x_1}),p_t(\boldsymbol{x}|\boldsymbol{x}_1)}||\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1)||^2, \text{ where } t \sim U[0,1].$$

Theorem 1

Assuming that $p_t(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and $t \in [0,1]$, then, up to a constant independent of $\boldsymbol{\theta}$, $\mathcal{L}_{CFM}$ and $\mathcal{L}_{FM}$ are equal. Hence, $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{FM} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{CFM}$.

- Therefore, Now we need to design suitable $\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_1)$ and $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$.

Flow Matching

Designing Conditional probability paths and vector fields

- Consider Gaussian conditional probability paths.

$$p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_t(\boldsymbol{x}_1), \sigma_t(\boldsymbol{x}_1)^2 I), \tag{3}$$

  where $\boldsymbol{\mu} : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, $\sigma : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^+$, and I denote the identity matrix.
- Set $\boldsymbol{\mu}_0(\boldsymbol{x}_1) = 0$ and $\sigma_0(\boldsymbol{x}_1) = 1$, to make $p_0(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid 0, I)$.
- Consider the flow

$$\phi_t(\boldsymbol{x}) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x} + \boldsymbol{\mu}_t(\boldsymbol{x}_1). \tag{4}$$

- Then the corresponding vector field is

$$\frac{d}{dt}\phi_t(\boldsymbol{x}) = \boldsymbol{u}_t(\phi_t(\boldsymbol{x}) \mid \boldsymbol{x}_1).$$

Theorem 2

Let $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ be a Gaussian probability path as in equation (3), and ϕ_t its corresponding flow map as in equation (4). Then, the unique vector field that defines ϕ_t has the form:

$$\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \frac{\sigma_t'(\boldsymbol{x}_1)}{\sigma_t(\boldsymbol{x}_1)}(\boldsymbol{x} - \boldsymbol{\mu}_t(\boldsymbol{x}_1)) + \boldsymbol{\mu}_t'(\boldsymbol{x}_1),$$

where $f' = \frac{d}{dt}f$. Consequently, $\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ generates the Gaussian path $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$.

Flow Matching

Designing Conditional probability paths and vector fields

Example 1: Diffusion conditional vector fields

Use diffusion paths as conditional probability paths in Flow Matching.

- **Flow Matching recovers diffusion dynamics.**
- Reversed (noise $\to$ data) variance exploding diffusion path

$$p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}_1, s_{1-t}^2 I),$$

  where s_t is an increasing function, $s_0 = 0$, and $s_1 \gg 1$.

$$\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = -\frac{s_{1-t}'}{s_{1-t}}(\boldsymbol{x} - \boldsymbol{x}_1).$$

- Reversed (noise $\to$ data) variance preserving diffusion path

$$p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x} \mid \alpha_{1-t}\boldsymbol{x}_1, (1 - \alpha_{1-t}^2) I),$$

  where $\alpha_t = e^{-\frac{1}{2}T(t)}$, $T(t) = \int_0^t \beta$, and β is the noise scale function.

$$\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \frac{\alpha_{1-t}'}{1 - \alpha_{1-t}^2}(\alpha_{1-t}\boldsymbol{x} - \boldsymbol{x}_1).$$

Example 2: Optimal Transport (OT) conditional vector fields

Define the mean and the standard deviation to simply change linearly in time.

$$\boldsymbol{\mu}_t(\boldsymbol{x}) = t\boldsymbol{x}_1, \quad \sigma_t(\boldsymbol{x}) = 1 - (1 - \sigma_{min})t,$$

where σ_{min} is a small constant representing the minimum noise level.

$$\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = \frac{\boldsymbol{x}_1 - (1 - \sigma_{min})\boldsymbol{x}}{1 - (1 - \sigma_{min})t}.$$

- **$\boldsymbol{x}_t$ moves in straight line with constant speed, leading to more efficient computation.**

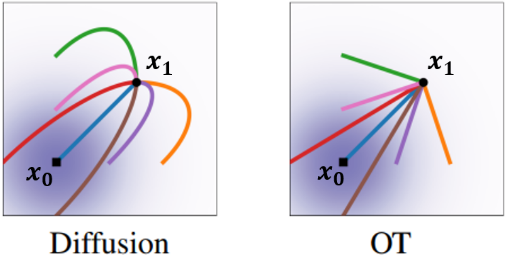

Figure. Diffusion and OT paths

Flow Matching

Experiments

1. Density modeling and sample quality
2. Sampling efficiency
3. Conditional sampling from low-resolution images

Density modeling and sample quality

Model	CIFAR-10			ImageNet 32×32			ImageNet 64×64		
	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓
Ablations									
DDPM	3.12	7.48	274	3.54	6.99	262	3.32	17.36	264
Score Matching	3.16	19.94	242	3.56	5.68	178	3.40	19.74	441
ScoreFlow	3.09	20.78	428	3.55	14.14	195	3.36	24.95	601
Ours									
FM w/ Diffusion	3.10	8.06	183	3.54	6.37	193	3.33	16.88	187
FM w/ OT	**2.99**	**6.35**	**142**	**3.53**	**5.02**	**122**	**3.31**	**14.45**	**138**

Model	ImageNet 128×128	
	NLL↓	FID↓
MGAN (Hoang et al., 2018)	–	58.9
PacGAN2 (Lin et al., 2018)	–	57.5
Logo-GAN-AE (Sage et al., 2018)	–	50.9
Self-cond. GAN (Lučić et al., 2019)	–	41.7
Uncond. BigGAN (Lučić et al., 2019)	–	25.3
PGMGAN (Armandpour et al., 2021)	–	21.7
FM w/ OT	**2.90**	**20.9**

Figure. Likelihood (BPD), quality of generated samples (FID), and evaluation time (NFE) for the same model trained with different methods.

Sampling efficiency

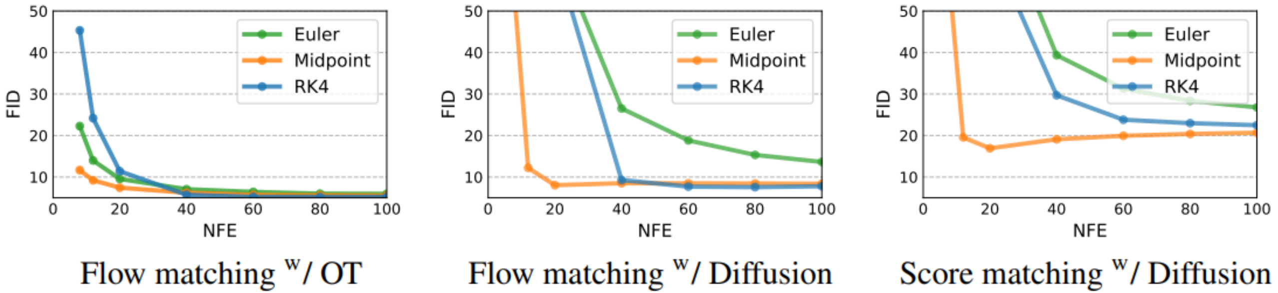

Figure. FID score and evaluation time for the models trained on ImageNet 32 $\times$ 32.

- Flow Matching, especially when using OT paths, allows us to use fewer evaluations for sampling while retaining similar sample quality.

Sampling efficiency

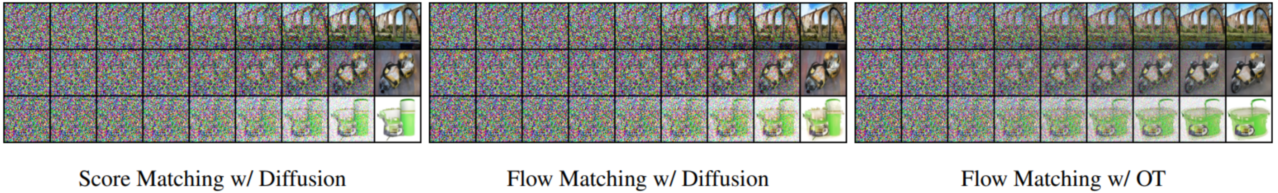

Figure. Sample paths from the same initial noise with models trained on ImageNet 64×64.

- The OT path reduces noise roughly linearly, while diffusion paths visibly remove noise only towards the end of the path.
- Note also the differences between the generated images.

Flow Matching

Experiments

Conditional sampling from low-resolution images

Model	FID↓	IS↑	PSNR↑	SSIM↑
Reference	1.9	240.8	–	–
Regression	15.2	121.1	**27.9**	**0.801**
SR3 (Saharia et al., 2022)	5.2	180.1	26.4	0.762
FM w/ OT	**3.4**	**200.8**	24.7	0.747

Figure. Image super-resolution on the ImageNet validation set.

Figure. Conditional generation $64 \times 64 \rightarrow 256 \times 256$. Flow Matching OT upsampled images from validation set.

Thank You

Inducing Negative Log-likelihood

Let $J_t = \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0}$ be the Jacobian. The log-density is $\log p_t(\mathbf{x}_t) = \log p_0(\mathbf{x}_0) - \log \det J_t$.

By the chain rule,

$$\frac{dJ_t}{dt} = \frac{\partial}{\partial \mathbf{x}_0}\left(\frac{d\mathbf{x}_t}{dt}\right) = \frac{\partial \mathbf{v}_t(\mathbf{x}_t)}{\partial \mathbf{x}_0} = \frac{\partial \mathbf{v}_t(\mathbf{x}_t)}{\partial \mathbf{x}_t}\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0} = \left(\frac{\partial \mathbf{v}_t}{\partial \mathbf{x}_t}\right) J_t$$

Since the derivative of a log-determinant is given by $\frac{d}{dt}\log\det A_t = \mathrm{Tr}\left(A_t^{-1}\frac{dA_t}{dt}\right)$.
Substituting J_t for A_t,

$$\frac{d}{dt}\log\det J_t = \mathrm{Tr}\left(J_t^{-1}\frac{dJ_t}{dt}\right) = \mathrm{Tr}\left(J_t^{-1}\left(\frac{\partial \mathbf{v}_t}{\partial \mathbf{x}_t}\right) J_t\right) = \mathrm{Tr}\left(\frac{\partial \mathbf{v}_t}{\partial \mathbf{x}_t}\right),$$

where the last step uses the cyclic property of the Trace ($\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA)$).

Therefore,

$$\frac{d}{dt}\log p_t(\mathbf{x}_t) = -\mathrm{Tr}\left(\frac{\partial \mathbf{v}_t}{\partial \mathbf{x}_t}(\mathbf{x}_t)\right)$$