Flow Matching and Diffusion Models

Open DMQA Seminar
2025.02.28

조한샘

고려대학교
KOREA UNIVERSITY

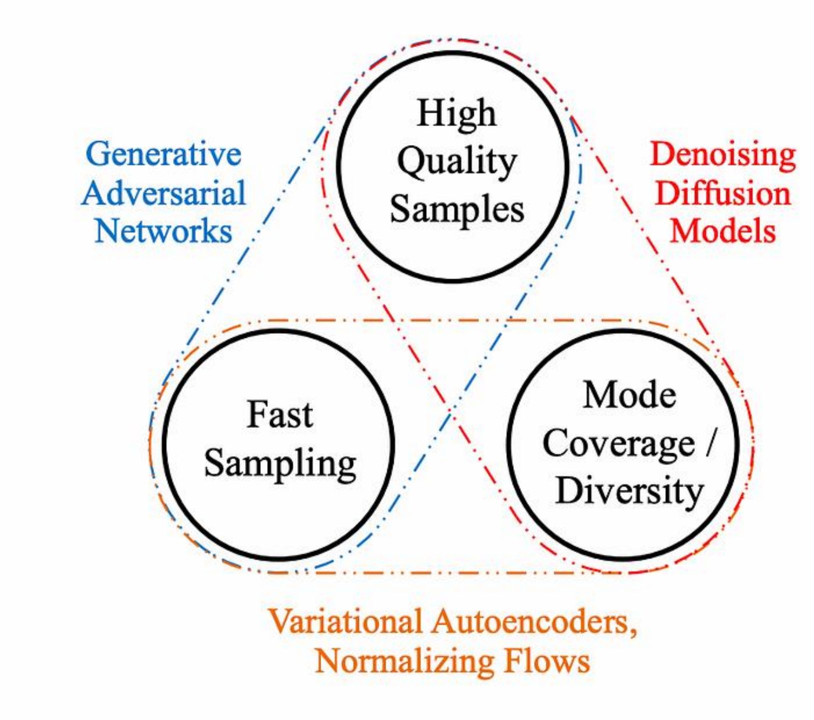
발표자 소개

- **조한샘**
  - ✓ Data Mining & Quality Analytics Lab
  - ✓ 석·박통합과정 (2020.09~)
- **관심 연구 분야**
  - ✓ Visual Generative Models
  - ✓ Controllable Generation
- **Contact**
  - ✓ chosam95@korea.ac.kr

Introduction

Generative Learning Trilemma

- 생성모델의 3가지 요소
- Diffusion model은 다양한 고품질의 데이터를 생성할 수 있지만 생성 속도가 느림

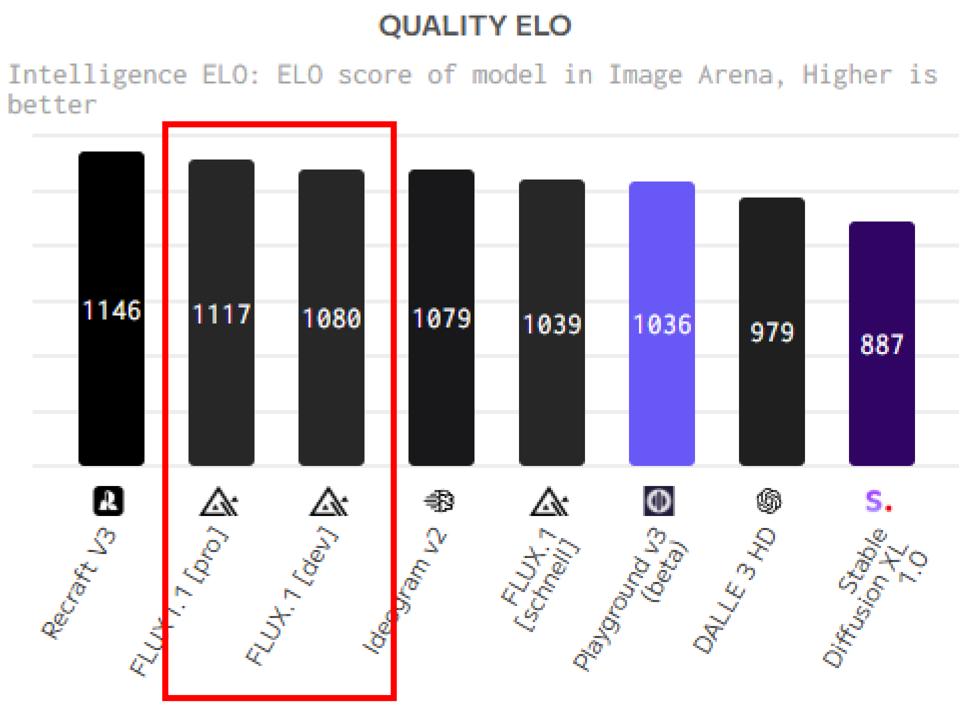


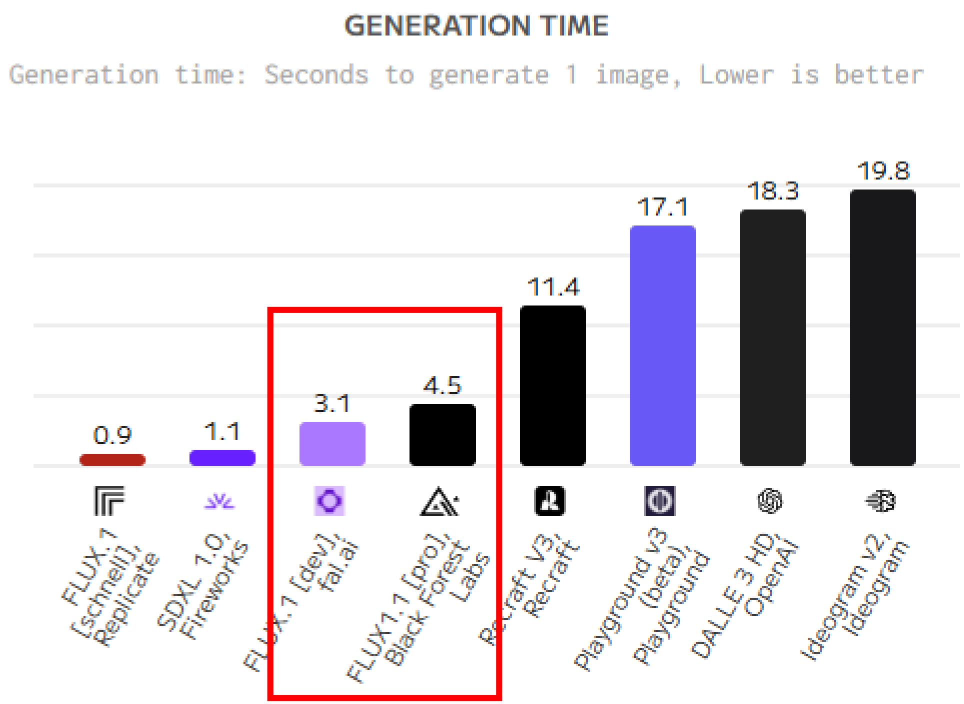
Xiao, Z., Kreis, K., & Vahdat, A. (2022). Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. In International Conference on Learning Representations (ICLR).

Introduction

Flow matching

- Flow matching을 기반으로 하는 FLUX같은 모델이 이미지 퀄리티가 높고 속도가 빠름

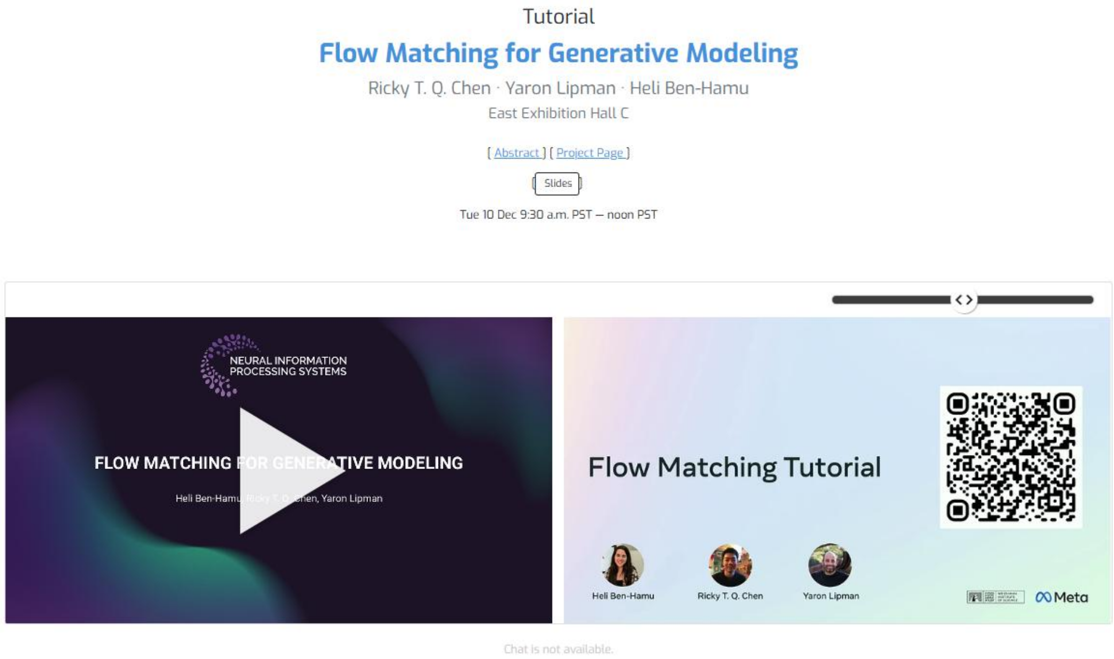




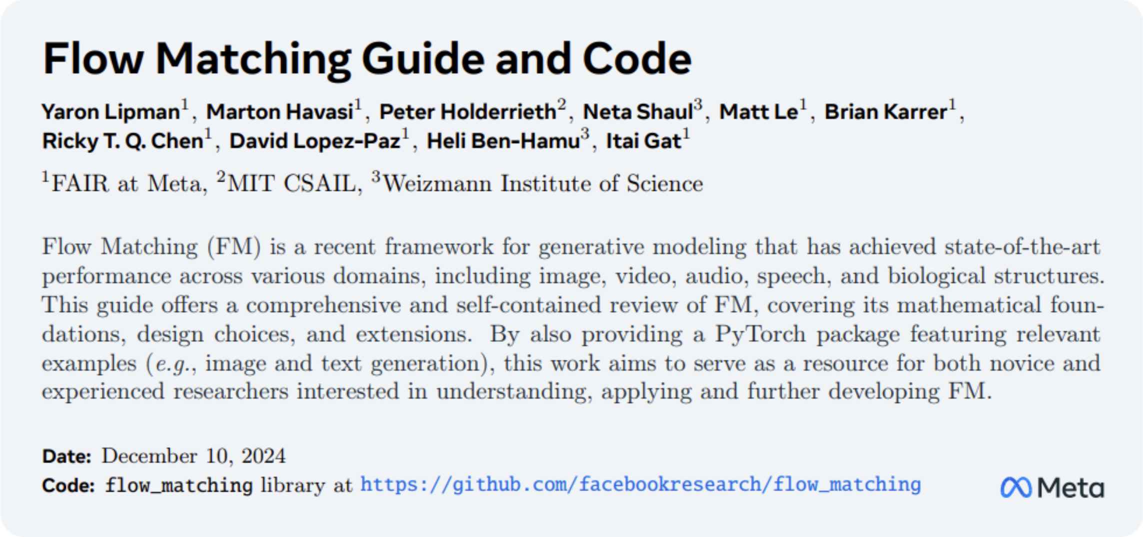
https://artificialanalysis.ai/text-to-image

Flow Models

Reference: NeurIPS 2024 Tutorial



https://neurips.cc/virtual/2024/tutorial/99531



Flow Matching Guide and Code

Yaron Lipman[1], Marton Havasi[1], Peter Holderrieth[2], Neta Shaul[3], Matt Le[1], Brian Karrer[1], Ricky T. Q. Chen[1], David Lopez-Paz[1], Heli Ben-Hamu[3], Itai Gat[1]

[1]FAIR at Meta, [2]MIT CSAIL, [3]Weizmann Institute of Science

Flow Matching (FM) is a recent framework for generative modeling that has achieved state-of-the-art performance across various domains, including image, video, audio, speech, and biological structures. This guide offers a comprehensive and self-contained review of FM, covering its mathematical foundations, design choices, and extensions. By also providing a PyTorch package featuring relevant examples (*e.g.*, image and text generation), this work aims to serve as a resource for both novice and experienced researchers interested in understanding, applying and further developing FM.

Date: December 10, 2024
Code: `flow_matching` library at https://github.com/facebookresearch/flow_matching

Meta

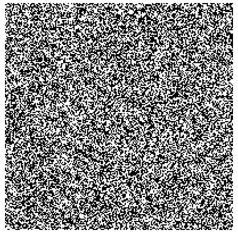
https://arxiv.org/abs/2412.06264

Flow Models

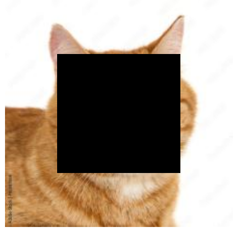
Generative Models

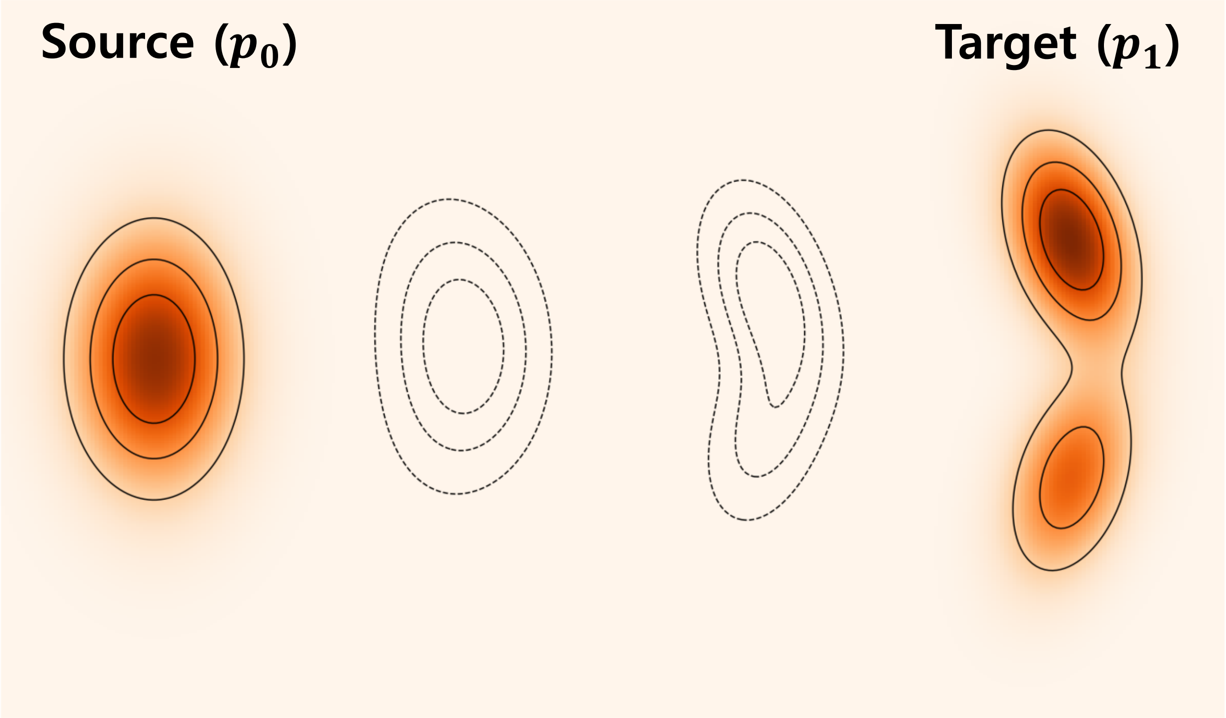
- 생성모델: 데이터의 분포를 학습
- **Source 분포** (input, p_0) / **Target 분포(=데이터 분포)** (output, p_1)
- **<u>Source 분포에서 target 분포로 변환되는 과정</u>**을 학습함으로써 데이터 분포 학습

Image synthesis



Inpainting

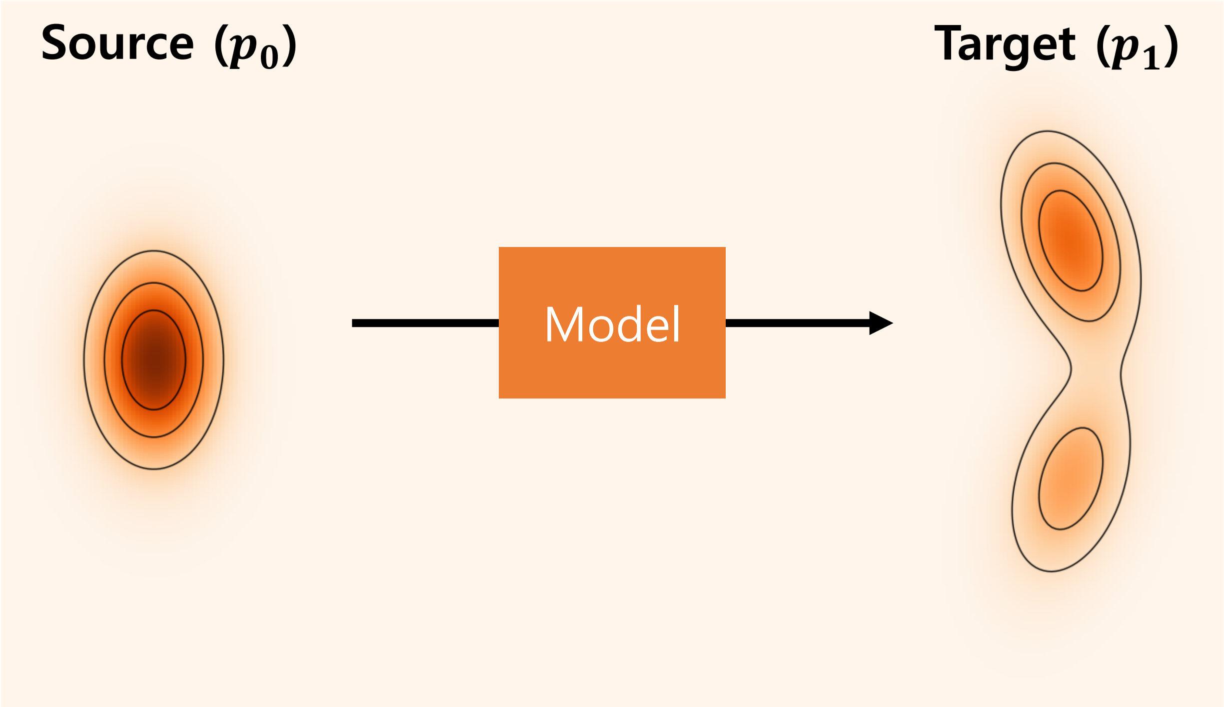




Flow Models

Generative Models

- GAN*, VAE** 같은 모델은 source 분포에서 target 분포를 한번에 mapping
- **장점**: 생성 속도가 빠름
- **단점**: 데이터 분포 학습이 어려움

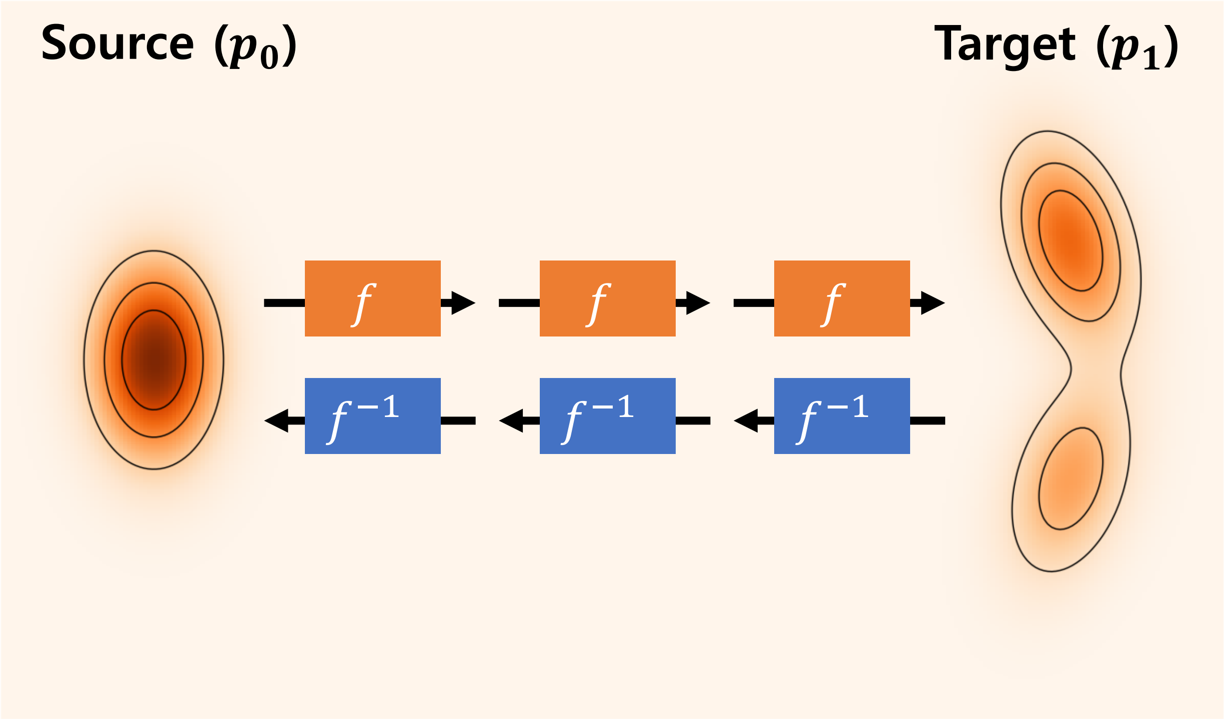


*Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
**Kingma, D. P., & Welling, M. (2013, December). Auto-encoding variational bayes.

Flow Models

Generative Models

- Normalizing flow* 모델은 source 분포에서 target 분포로 단계적으로 변화함
- **장점**: likelihood 계산 가능
- **단점**: 역변환이 가능한 모델 구조가 필요, 학습이 비효율적

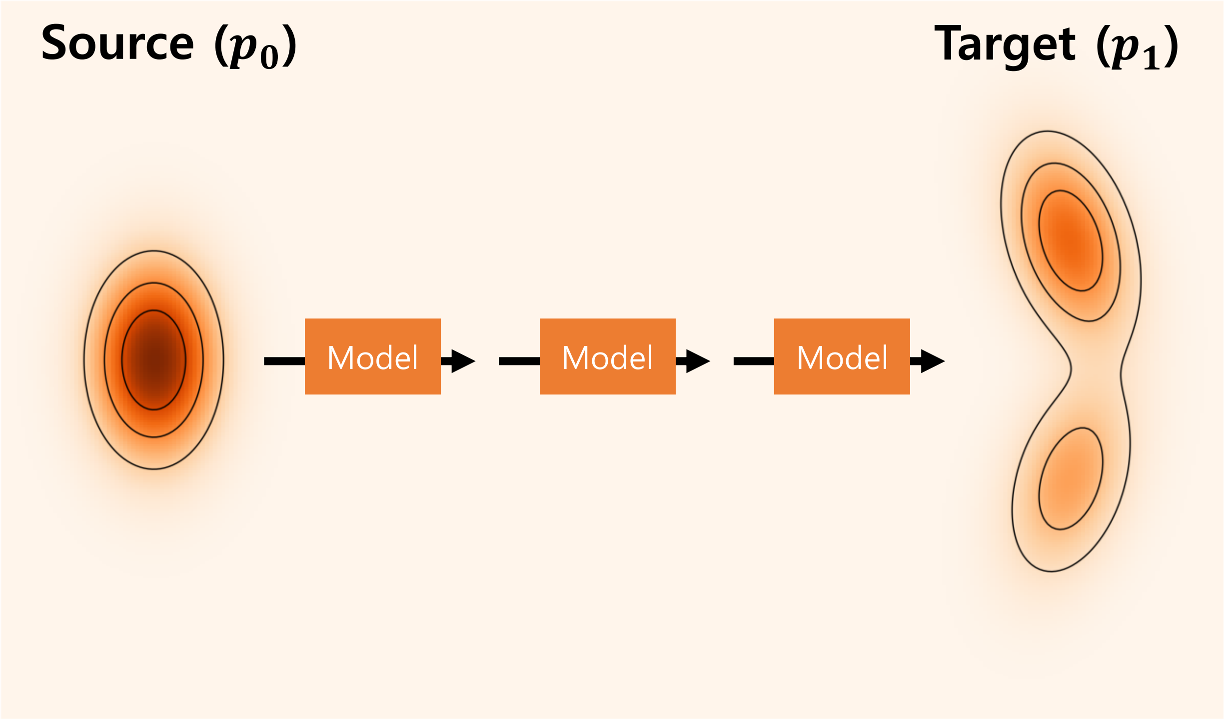


*Rezende, D., & Mohamed, S. (2015, June). Variational inference with normalizing flows. In International conference on machine learning (pp. 1530-1538). PMLR.

Flow Models

Generative Models

- Diffusion models*은 source 분포에서 target 분포로 단계적으로 변화함
- **장점**: 복잡한 데이터 분포 학습 가능, 학습이 효율적
- **단점**: 생성속도가 느림



*Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851.

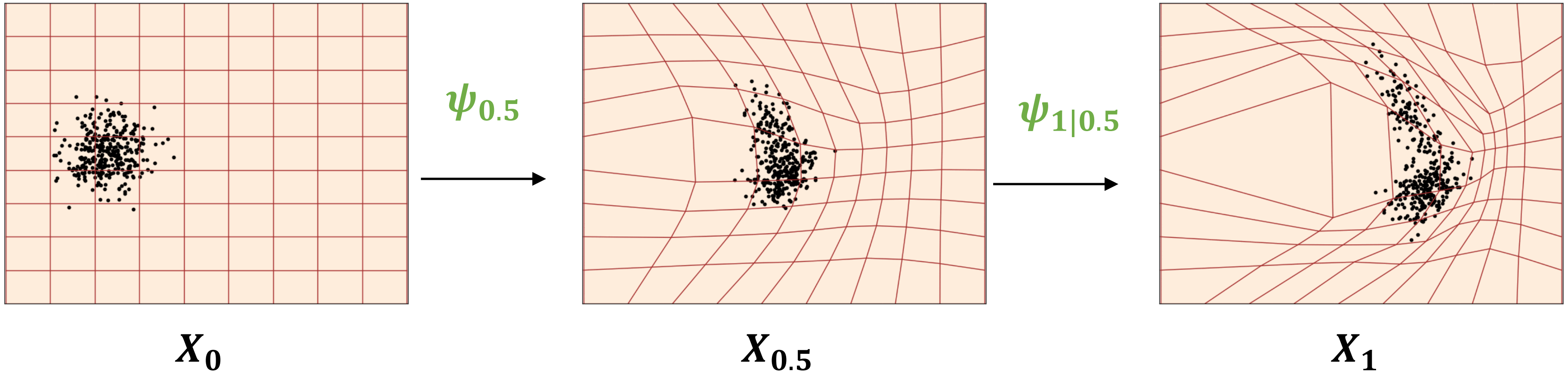
Flow Models

Flow Model

- **Flow model**: source 분포 p_0를 target 분포 p_1으로 변환해주는 **flow(ψ_t)**를 찾는 모델
- Flow(ψ_t): X_0를 X_t 로 mapping해주는 함수 (diffeomorphism / 미분가능 / 역함수 존재)
- 하지만 continuous time 상황에서 flow를 직접적으로 학습하기는 어려움

$$\boldsymbol{X_t = \psi_t(X_0), \qquad where\ X_0 \sim p_0}$$

$$\boldsymbol{\psi_{s|t} = \psi_s \circ \psi_t^{-1}}$$



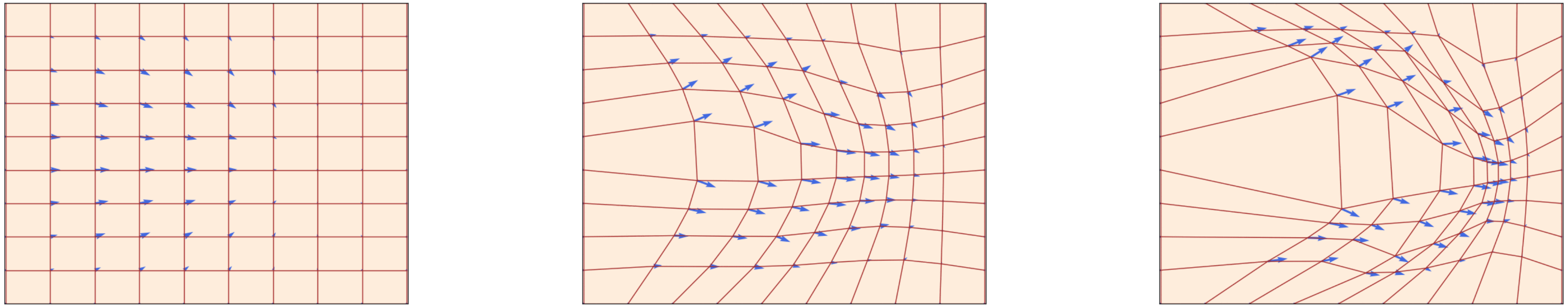
Flow Models

Velocity Field

- Velocity field를 통해 flow를 간접적으로 계산
- **Velocity field(u_t)**: flow의 미분으로 정의, t시점 X_t의 변화량인 ODE로 표현 가능

$$\frac{d}{dt}\psi_t(X_0) = u_t(\psi_t(X_0)) \quad \Rightarrow \quad \frac{dX_t}{dt} = u_t(X_t)$$

u_0 $\quad$ $u_{0.5}$ $\quad$ u_1



X_0 $\quad$ $X_{0.5}$ $\quad$ X_1

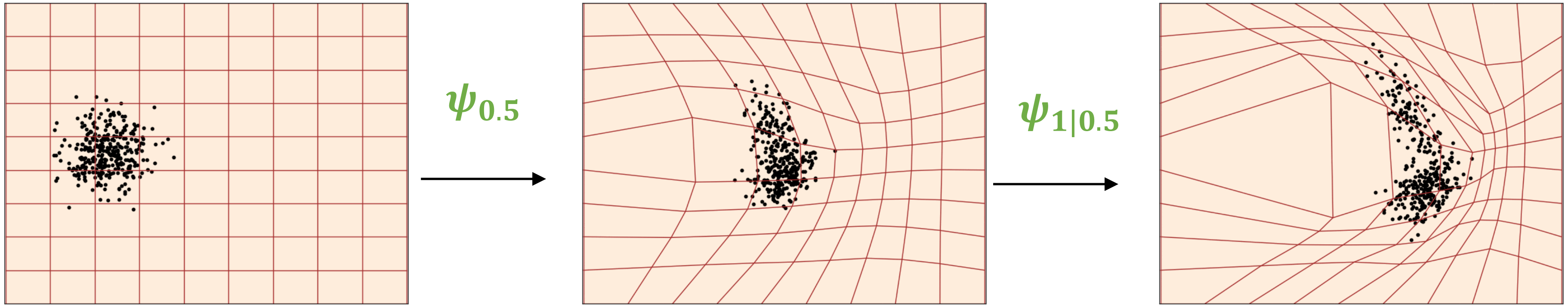
Flow Models

Flow and Velocity field

- Velocity field를 계산한 후 ODE를 풀면 flow를 계산할 수 있음
- Euler method와 같은 ODE solver을 통해서

$$\text{Flow}(\psi_t) \underset{\text{Solve ODE}}{\overset{\text{미분}}{\rightleftarrows}} \text{Velocity field}(u_t)$$

$$\textit{Euler Solever}: \quad X_{t+h} = X_t + h \cdot u_t(X_t)$$

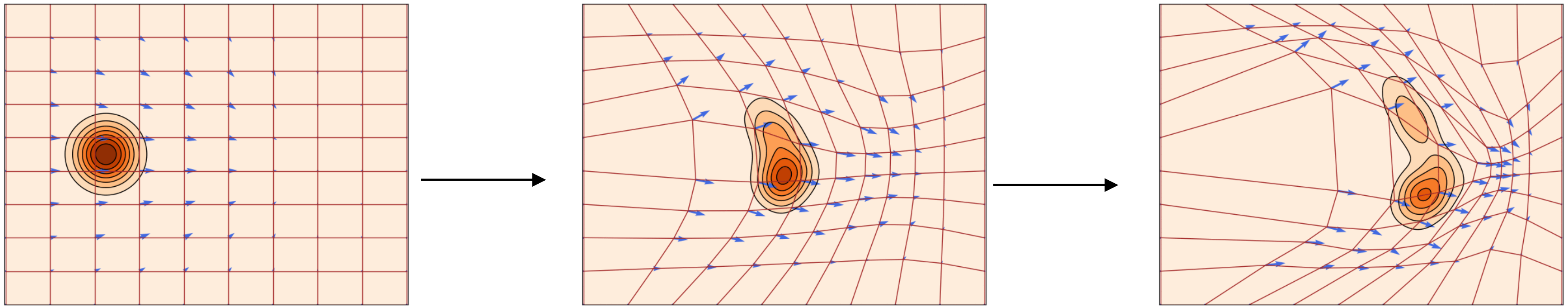


X_0 $\qquad X_{0.5} = X_0 + 0.5u_0(X_0)$ $\qquad X_1 = X_{0.5} + 0.5u_{0.5}(X_{0.5})$

Flow Models

Probability Paths

- **Probability path(p_t)**: source 분포 p_0를 target 분포 p_1으로 변환되는 과정의 중간 t시점 분포



$X_0 \sim p_0$ $X_{0.5} \sim p_{0.5}$ $X_1 \sim p_1$

Flow Models

Flow Matching (FM)

- **Flow model**: source 분포 p_0를 target 분포 p_1으로 변환해주는 **flow(ψ_t)**를 찾는 모델
- Flow는 velocity field를 통해서 계산됨
- Flow model은 **velocity field를 학습함**으로써 source 분포에서 target 분포로의 변환을 수행 할 수 있음
- 하지만 velocity field를 직접적으로 계산하기 어려움

Flow(ψ_t)

$$X_t = \psi_t(X_0), \quad where\ X_0 \sim p_0$$

Velocity field(u_t)

$$\frac{dX_t}{dt} = u_t(X_t) \xrightarrow{\text{Solve ODE}} X_{t+h} = X_t + h \cdot u_t(X_t)$$

$$L_{FM}(\theta) = E_{t\sim U(0,1), X_t \sim p_t} \left\| \underbrace{u_t^{\theta}(X_t)}_{\text{모델}} - \underbrace{u_t(X_t)}_{\text{타겟}} \right\|_2^2$$

Flow Models

Conditional Flow Matching (CFM)

- Target 분포 **X_1에 대한 condition**을 통해서 conditional velocity field를 학습
- 두 loss의 gradient는 동일 = **CFM loss로 학습하더라도 FM loss로 학습된 모델을 얻을 수 있음**

모델 타겟

$$L_{FM}(\theta) = E_{t\sim U(0,1),\, X_t\sim p_t}\left\| u_t^{\theta}(X_t) - u_t(X_t) \right\|_2^2$$

↓

$$L_{CFM}(\theta) = E_{t\sim U(0,1),\, X_t\sim p_t,\, X_1\sim p_1}\left\| u_t^{\theta}(X_t) - u_t(X_t|X_1) \right\|_2^2$$

$$\nabla_\theta L_{FM}(\theta) = \nabla_\theta L_{CFM}(\theta)$$

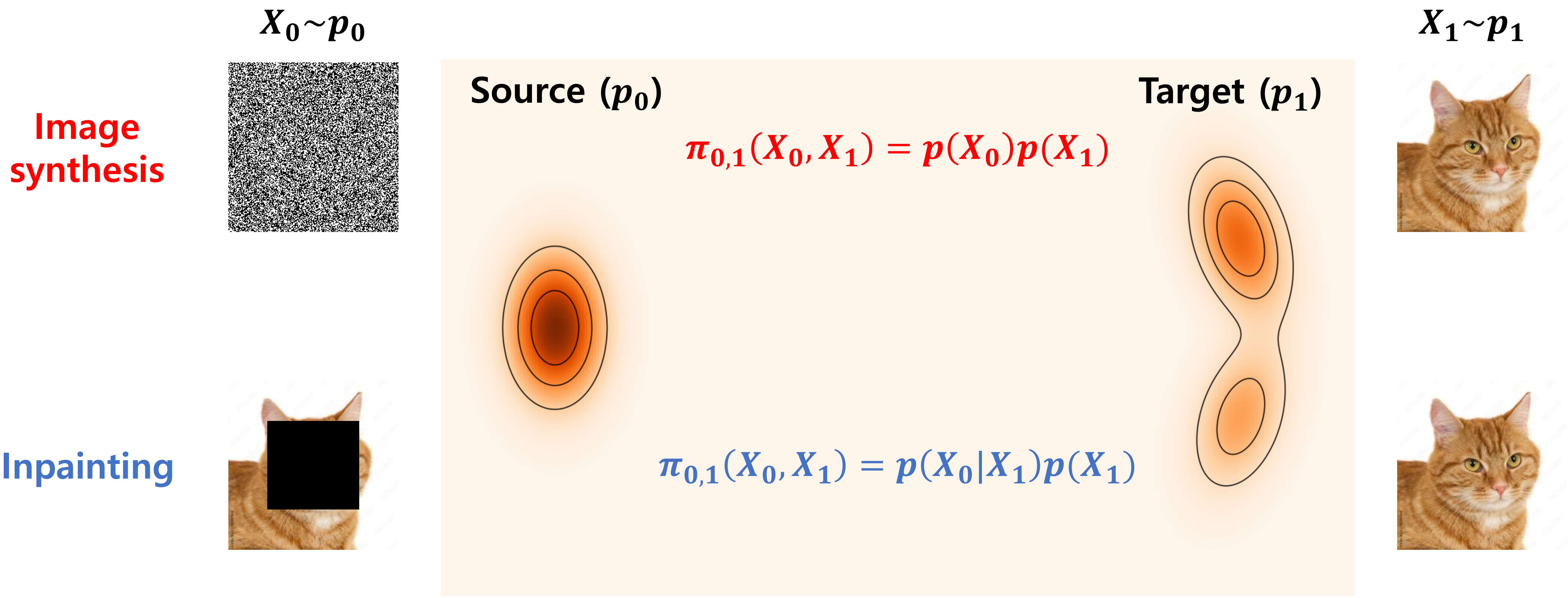
(Flow Matching*, ICLR 2023)

*Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., & Le, M. Flow Matching for Generative Modeling. In The Eleventh International Conference on Learning Representations.

Flow Models

Training – 1. Data

- Source와 target 분포의 데이터 수집 및 관계 정의
- **Image synthesis**: source와 target이 독립 / **Inpainting**: source가 target에 의존적



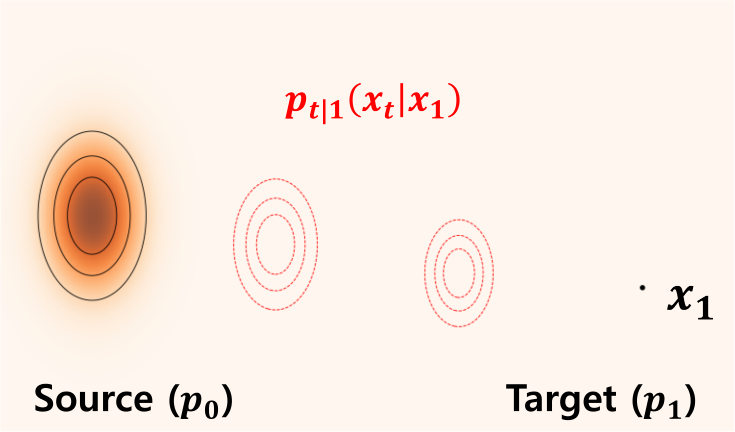
Flow Models

Training – 2. Probability Paths

- **<u>Source와 target 분포를 연결하는 방법 정의</u>**
- 직접 probability path($\boldsymbol{p_t}$)를 정의하기는 어려움
- Target 분포 X_1을 condition으로 활용해서 **<u>conditional probability path($\boldsymbol{p_{t|1}}$)를 정의</u>**
- Conditional probability path를 모든 target sample x_1에 대해서 더해줌으로써 probability path를 정의

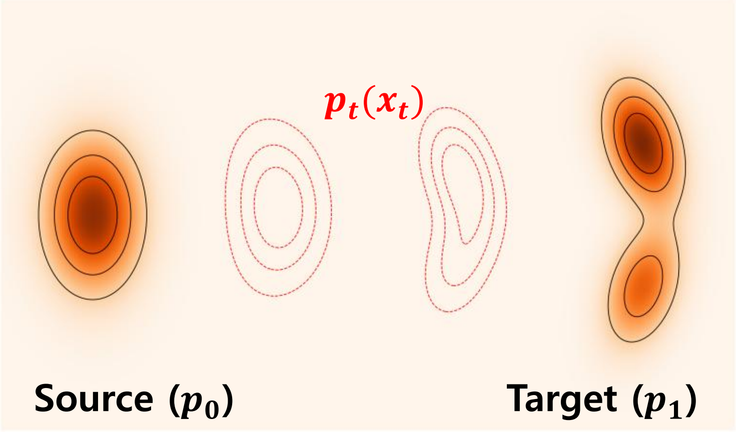
Conditional Probability path

$$\boldsymbol{p_{t|1}(x_t|x_1) = N(x_t; tx_1, (1-t)^2 I)}$$



Probability path

$$\boldsymbol{p_t(x_t) = \int p_{t|1}(x_t|x_1) p_1(x_1) dx_1}$$



Flow Models

Training – 3. Velocity Field

- Conditional velocity field 계산

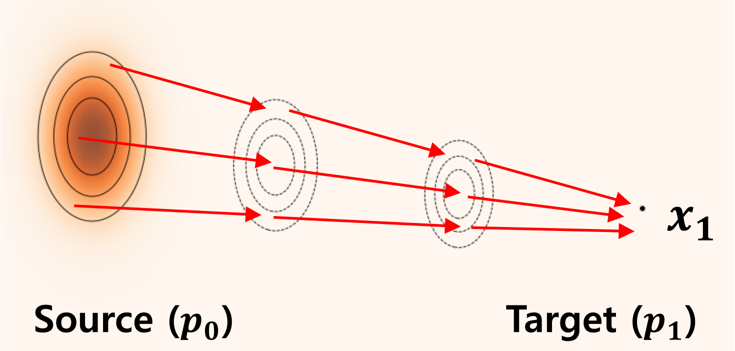
Conditional Probability path

$$p_{t|1}(x|x_1) = N(x; tx_1, (1-t)^2 I) \longrightarrow X_{t|1} = tx_1 + (1-t)X_0$$

Conditional Velocity Field

$$\frac{dX_{t|1}}{dt} = u_t(X_{t|1}) = u(X_t|x_1)$$

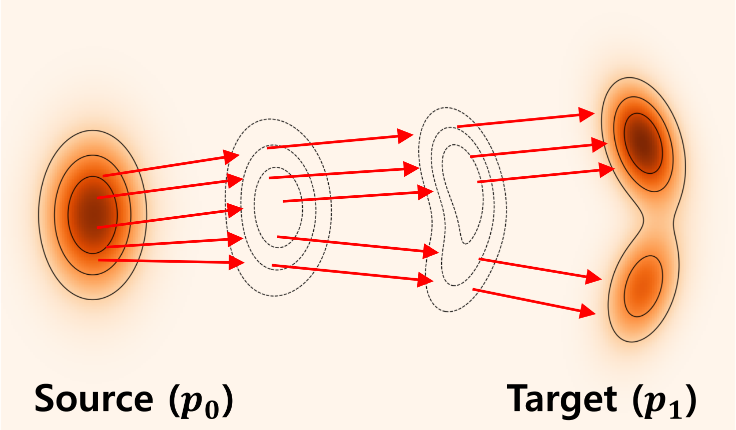
$$\begin{aligned} u_t(x_t|x_1) &= x_1 - X_0 \\ &= x_1 - \frac{x_t - tx_1}{1-t} \\ &= \frac{x_1 - x_t}{1-t} \end{aligned}$$



Velocity Field

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

$$u_t(x_t) = \int u_t(x_t|x_1) p_{1|t}(x_1|x_t) dx_1$$



Flow Models

Training – 4. Conditional Flow Matching

- Conditional velocity field를 활용해 학습

모델 타겟

$$\boldsymbol{L_{CFM}(\theta) = E_{t\sim U(0,1),\, X_t\sim p_t,\, X_1\sim p_1} \left\| {\color{red} u_t^{\theta}}(X_t) - {\color{blue} u_t(X_t|X_1)} \right\|_2^2}$$

Conditional Probability path

$$\boldsymbol{p_{t|1}(x_t|x_1) = N(x_t; t x_1, (1-t)^2 I)}$$

Conditional Velocity field

$$\boldsymbol{u_t(x_t|x_1) = x_1 - X_0}$$

$$\boldsymbol{L_{CFM}(\theta) = E_{t\sim U(0,1),\, X_t\sim p_t,\, X_1\sim p_1} \left\| {\color{red} u_t^{\theta}}(X_t) - {\color{blue} (X_1 - X_0)} \right\|_2^2}$$

Flow Models

Code

- Training

```
25 flow = Flow()
26 optimizer = torch.optim.Adam(flow.parameters(), 1e-2)
27 loss_fn = nn.MSELoss()
28
29 for _ in range(10000):
30     x_1  = Tensor(make_moons(256, noise=0.05)[0])
31     x_0  = torch.randn_like(x_1)
32     t    = torch.rand(len(x_1), 1)
33     x_t  = (1 - t) * x_0 + t * x_1
34     dx_t = x_1 - x_0
35     optimizer.zero_grad()
36     loss_fn(flow(x_t, t), dx_t).backward()
37     optimizer.step()
```

Conditional Probability path	$p_{t\|1}(x_t\|x_1) = N(x_t; tx_1, (1-t)^2 I)$
Conditional Velocity field	$u_t(x_t\|x_1) = x_1 - X_0$

30: target 분포 p_1에서 x_1샘플링
31: source 분포 p_0에서 x_0샘플링
32: t를 U(0,1)에서 샘플링
33: $x_t = (1-t)x_0 + tx_1$ (**conditional probability path**)
34: **conditional velocity filed** $x_1 - x_0$계산
35~37: CFM loss 계산 후 모델 업데이트

Flow Models

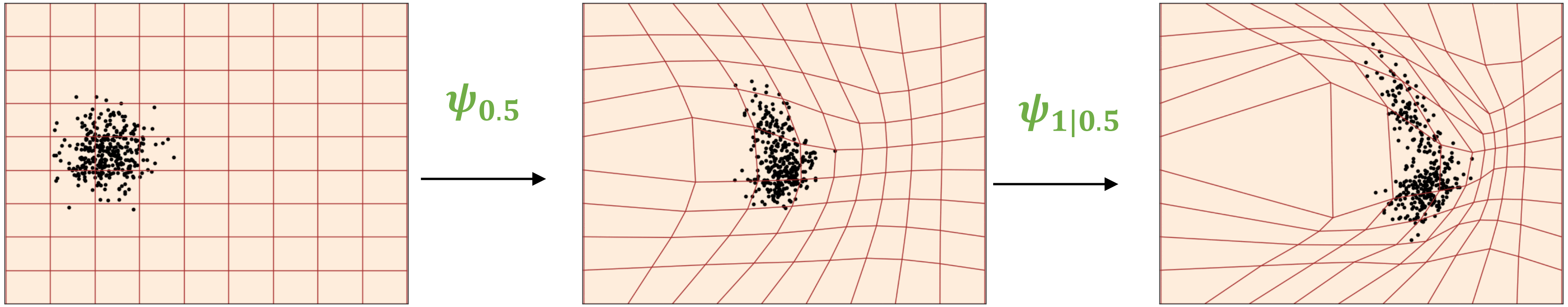
Code

- Sampling
- Step은 다양한 ODE solver 적용가능 (Euler, Heun)

```
50  for i in range(n_steps):
51    x = flow.step(x, time_steps[i], time_steps[i + 1])
```

51: 학습된 모델을 통해 정해진 step 수 만큼 진행

$$\boldsymbol{Euler\ Solever}: \quad X_{t+h} = X_t + h \cdot u_t(X_t)$$



X_0 $\quad$ $X_{0.5} = X_0 + 0.5u_0(X_0)$ $\quad$ $X_1 = X_{0.5} + 0.5u_{0.5}(X_{0.5})$

Flow Matching and Diffusion Models

Design space

- **Data coupling**: $\pi_{0,1}(X_0, X_1) = p(X_0)p(X_1)$
- **Probability path**: $p_{t|1}(x|x_1) = N(x; \mu_t(x_1), \sigma_t(x_1)^2\, I)$
- **Velocity field**: $u_t(x|x_1)$

DDPM

$$\mu_t(x_1) = \alpha_{1-t}x_1, \qquad \sigma_t(x_1) = \sqrt{1-\alpha_{1-t}^2}$$

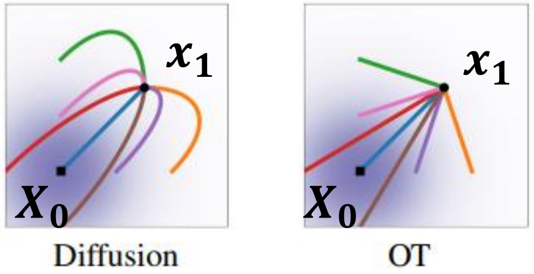
$$\downarrow$$

$$X_{t|1} = \alpha_{1-t}x_1 + \sqrt{1-\alpha_{1-t}^2}X_0$$

$$\downarrow$$

$$u_t(x_t|x_1) = \frac{\alpha'_{1-t}}{1-\alpha_{1-t}^2}(\alpha_{1-t}x_t - x_1)$$

$$where \quad a' = \frac{d}{dt}\alpha$$



Flow matching

$$\mu_t(x_1) = tx_1, \qquad \sigma_t(x_1) = 1-t$$

$$\downarrow$$

$$X_{t|1} = tx_1 + (1-t)X_0$$

$$\downarrow$$

$$u_t(x_t|x_1) = \frac{x_1 - x_t}{1-t}$$

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., & Le, M. Flow Matching for Generative Modeling. In The Eleventh International Conference on Learning Representations.

Flow Matching and Diffusion Models

Flow matching loss

- Velocity를 parameterize하는 방식의 차이
- **Flow matching** → velocity field에 대해서 학습
- **DDPM** → velocity field를 x_0 prediction (noise prediction)으로 변형한 후 학습 (Tutorial* Section 4.8.1 참고)

$$
\begin{aligned}
u_t(x) &= \dot{\alpha}_t \mathbb{E}\left[X_1 | X_t = x\right] + \dot{\sigma}_t \mathbb{E}\left[X_0 | X_t = x\right] && (4.55)\\
&= \frac{\dot{\sigma}_t}{\sigma_t} x + \left[\dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}\right] \mathbb{E}\left[X_1 | X_t = x\right] \quad \boldsymbol{x_1}\textbf{(image) prediction} && (4.56)\\
&= \frac{\dot{\alpha}_t}{\alpha_t} x + \left[\dot{\sigma}_t - \sigma_t \frac{\dot{\alpha}_t}{\alpha_t}\right] \mathbb{E}\left[X_0 | X_t = x\right], \quad \boldsymbol{x_0}\textbf{(noise) prediction} && (4.57)
\end{aligned}
$$

*Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., ... & Gat, I. (2024). Flow matching guide and code. arXiv preprint arXiv:2412.06264.

Flow Matching and Diffusion Models

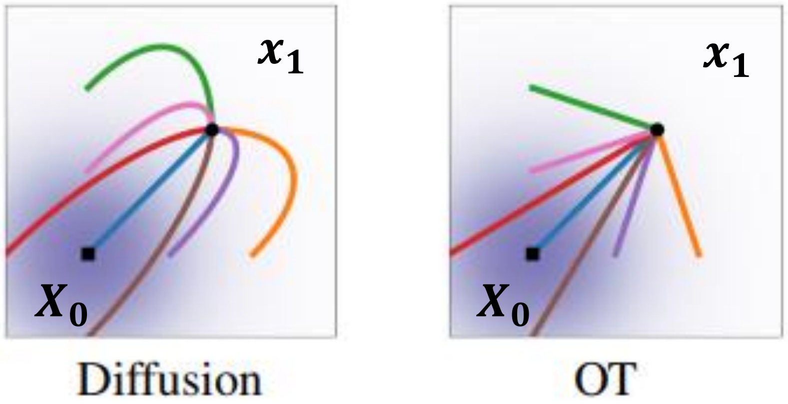
Flow Models as Generative Models

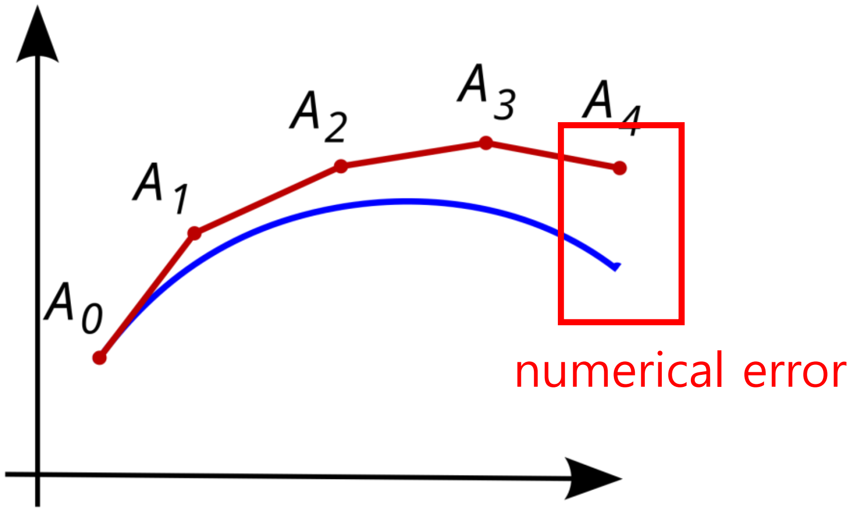
- **Data coupling**: $\pi_{0,1}(X_0, X_1)$ / 임의의 source 분포에서 target 분포로의 변환 과정
- **Probability paths**: $p_{t|1}(x|x_1)$ → **Velocity field**: $u_t(x|x_1)$
- **Velocity parameterization**: velocity, x_0-prediction, x_1-prediction

Flow Matching and Diffusion Models

Why Fast?

- ODE는 path가 직선에 가까울수록 numerical error가 감소
- Diffusion path보다는 OT path가 직선의 path를 만듦



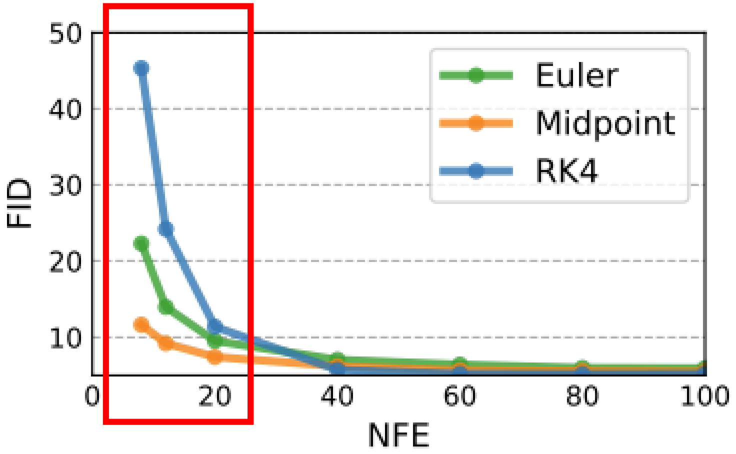


Euler Method

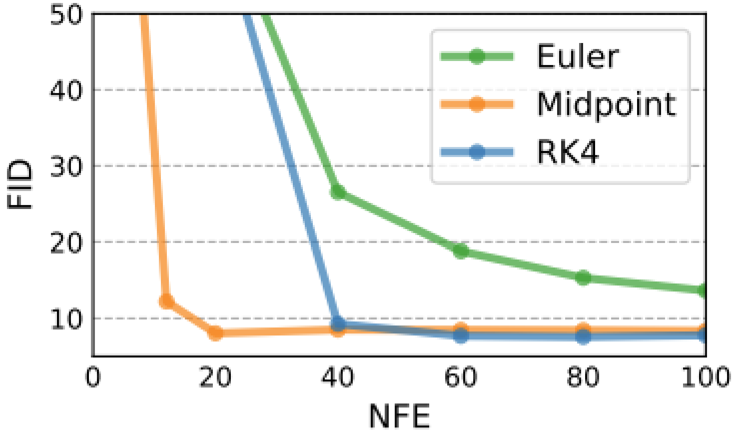
Flow Matching and Diffusion Models

Why Fast?

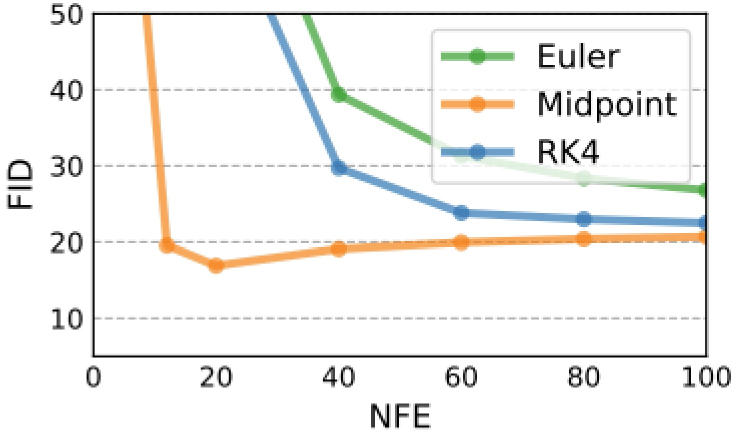
- Diffusion path를 사용하면 sampling step이 작은 경우 이미지 퀄리티가 좋지 않음



Flow matching $^{w}/$ OT



Flow matching $^{w}/$ Diffusion



Score matching $^{w}/$ Diffusion

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., & Le, M. Flow Matching for Generative Modeling. In The Eleventh International Conference on Learning Representations.

Conclusion

Flow Models

- **Flow model**: source 분포 p_0를 target 분포 p_1으로 변환해주는 **flow($\boldsymbol{\psi_t}$)**를 찾는 모델
- **Velocity field**를 통해서 flow를 계산 → **flow model은 velocity field를 학습**
- Diffusion model도 flow model의 한 종류
- 유연한 design space를 가지고 있어 다양한 상황에 적합한 모델 설계 가능