COMP547

DEEP UNSUPERVISED LEARNING

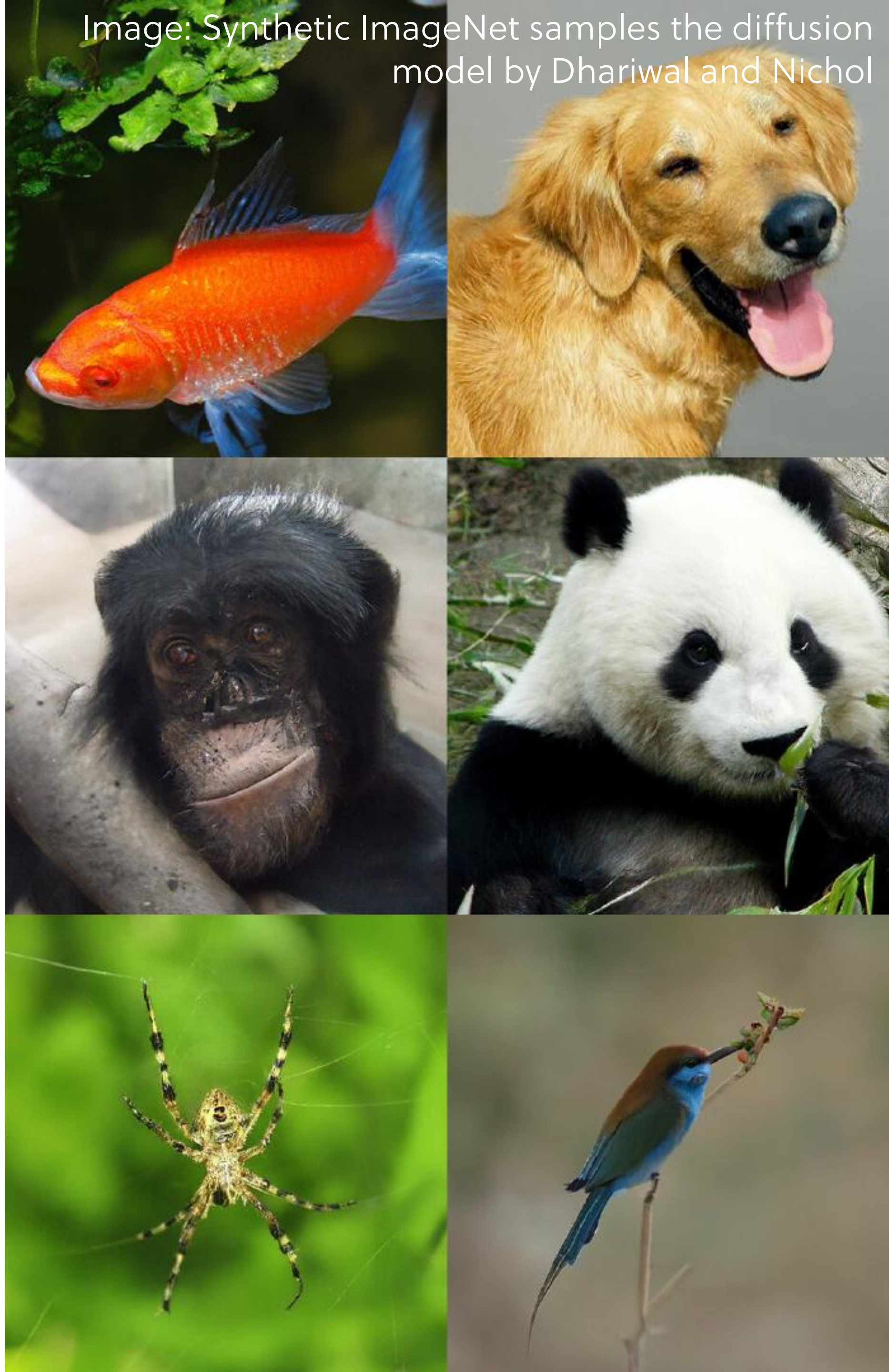
Lecture #11 – Flow Matching

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2026

Previously on COMP547

- Basics of Diffusion Models
- Deeper Dive into Diffusion Models



Lecture overview

- Notations
- Continuous Normalizing Flow and Flow Matching
- Conditional Flow Matching
- Reflow / k-Rectified Flow
- Applications

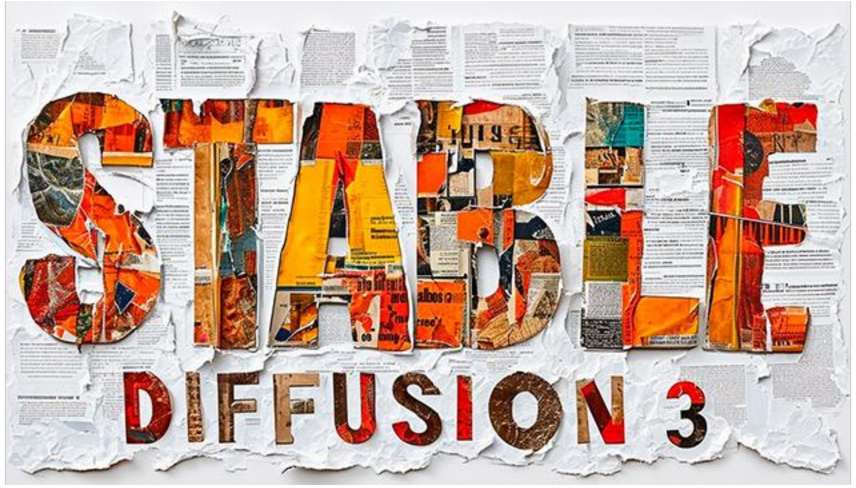
Disclaimer: Much of the material and slides for this lecture were borrowed from

—Minhyuk Sung's tutorial on Diffusion Models for Image & Video Generation: From Foundations to Emerging Directions

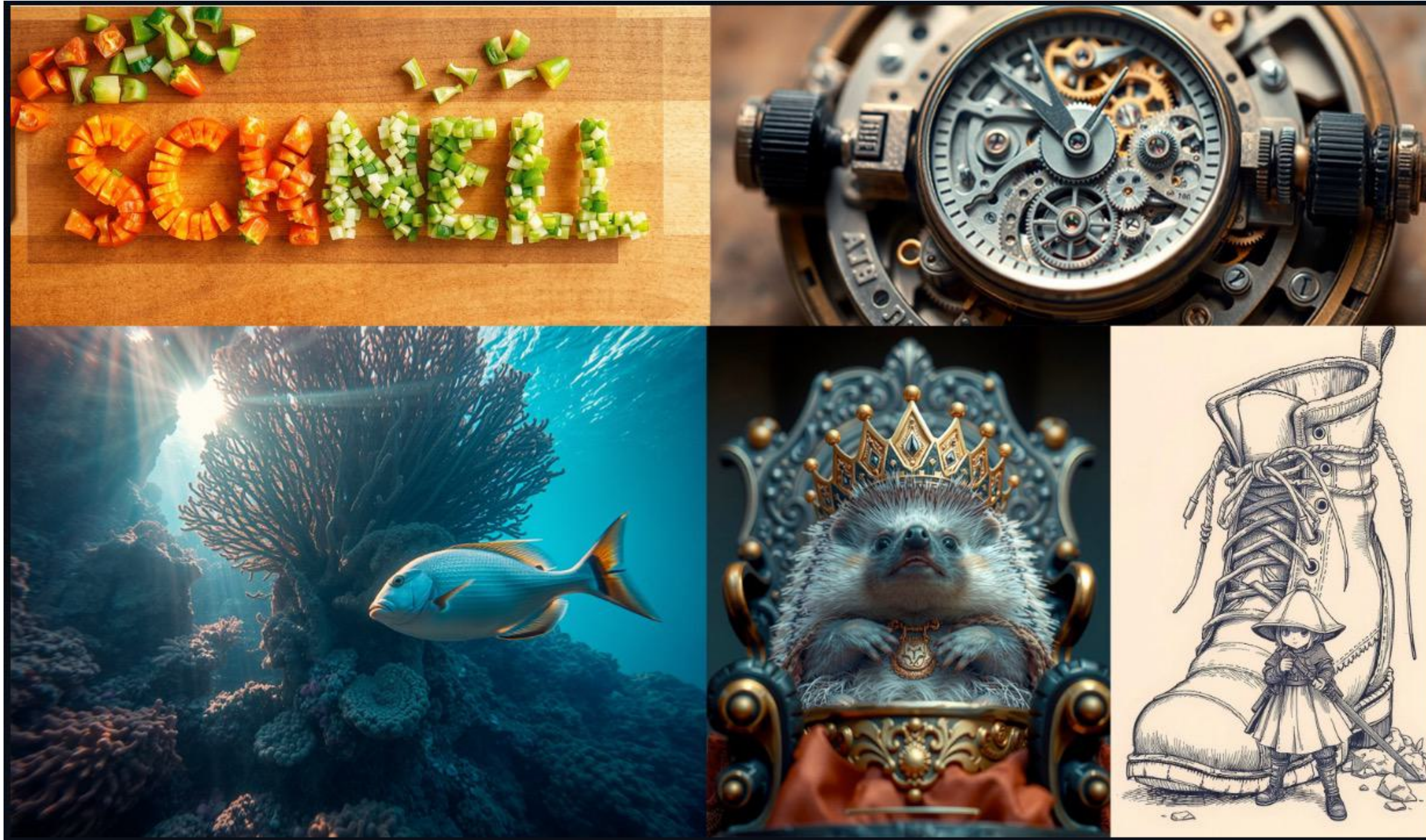
Lecture overview

- **Notations**
- Continuous Normalizing Flow and Flow Matching
- Conditional Flow Matching
- Reflow / k-Rectified Flow
- Applications

Example: Stable Diffusion 3



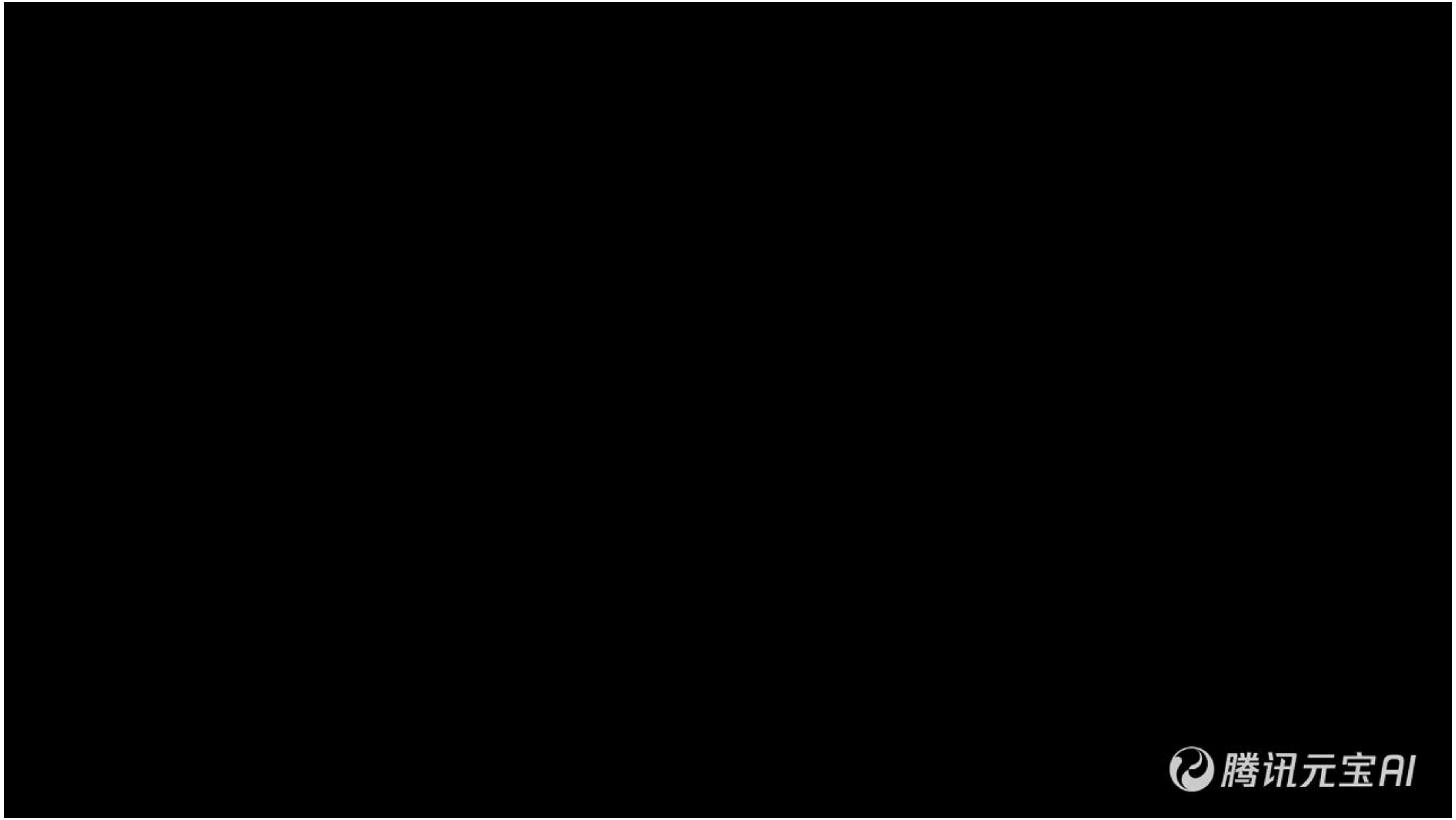
Example: Flux.1 Schnell



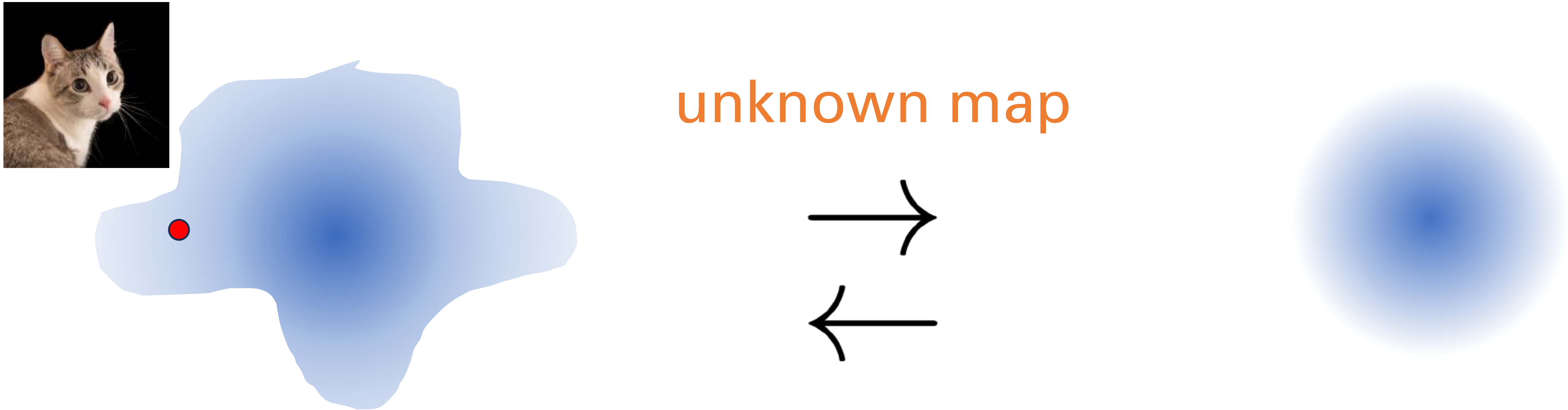
`FLUX.1 [schnell]` is a 12 billion parameter rectified flow transformer capable of generating images from text descriptions. For more information, please read our blog post.

Example: Meta MovieGen

Example: HunyuanVideo



Recap: Diffusion Models

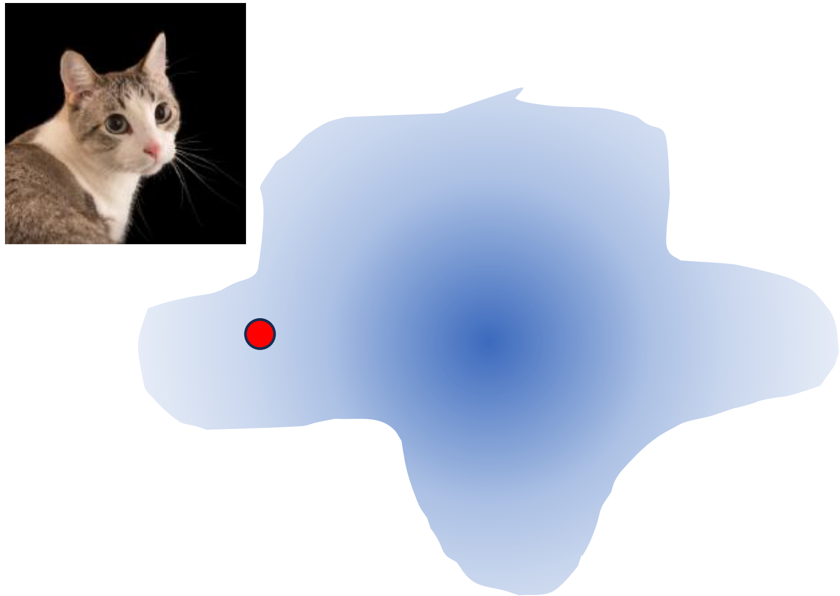


(unknown) data distribution
$p_0(\boldsymbol{x}_0)$

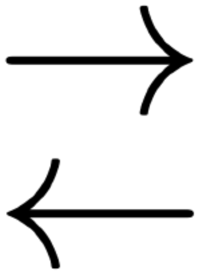
known distribution
$p_T(\boldsymbol{x}_T)$

Sampling $\Leftrightarrow$ (Unconditional generation)

Flow Models



unknown map



(unknown) data distribution

$p_1(\boldsymbol{x}_1)$

known distribution

$p_0(\boldsymbol{x}_0)$

Sampling $\Leftrightarrow$ *(Unconditional generation)*

Notations

Diffusion Models [Generation: $T \rightarrow 0$]

- $\boldsymbol{x}_0$: A sample from the data distribution.
- $\boldsymbol{x}_T$: A sample from the reference (base) distribution.

Flow Matching [Generation: $0 \rightarrow 1$]

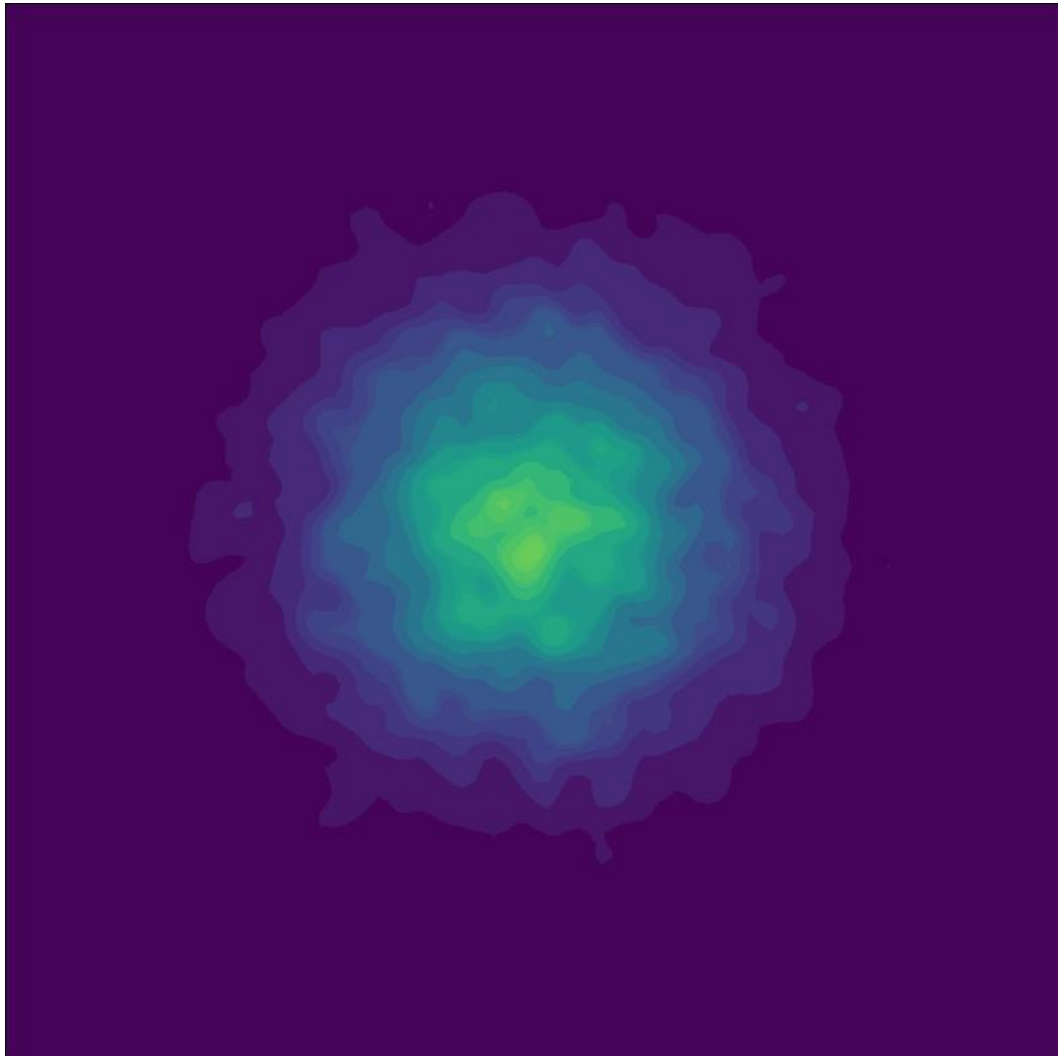
- $\boldsymbol{x}_1$: A sample from the data distribution.
- $\boldsymbol{x}_0$: A sample from the reference (base) distribution.

Lecture overview

- Notations
- **Continuous Normalizing Flow and Flow Matching**
  - Rezende and Mohamed, Variational Inference with Normalizing Flows, ICML'15.
- Conditional Flow Matching
- Reflow / k-Rectified Flow
- Applications

Continuous Normalizing Flow

- $p_0(\boldsymbol{x}_0)$: Reference distribution.
- $p_1(\boldsymbol{x}_1)$: Data distribution.

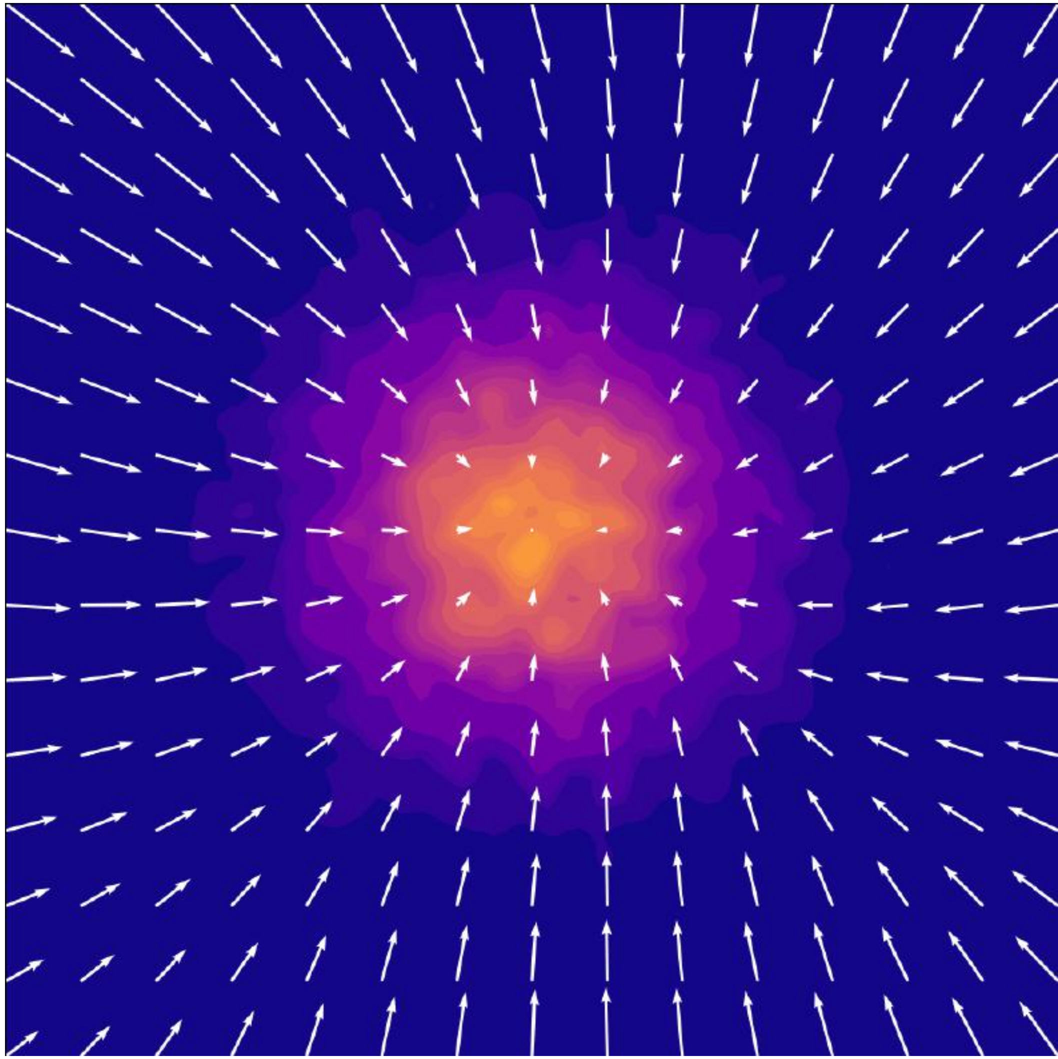


animation credit: Mark Ogata

Continuous Normalizing Flow

- $\phi_t(\boldsymbol{x}_0)$: A **flow map** that transports a sample $\boldsymbol{x}_0$ from $p_0(\boldsymbol{x}_0)$ to a sample $\boldsymbol{x}_t$ from $p_1(\boldsymbol{x}_1)$.

- $u_t(\boldsymbol{x}_t)$: A time-dependent **vector field**.



animation credit: Mark Ogata

Continuous Normalizing Flow

The flow $\phi_t(\boldsymbol{x}_0)$ is defined via the ordinary differential equation (ODE):

$$\frac{\partial \phi_t(\boldsymbol{x}_0)}{\partial t} = v_t\big(\phi_t(\boldsymbol{x}_0)\big)$$

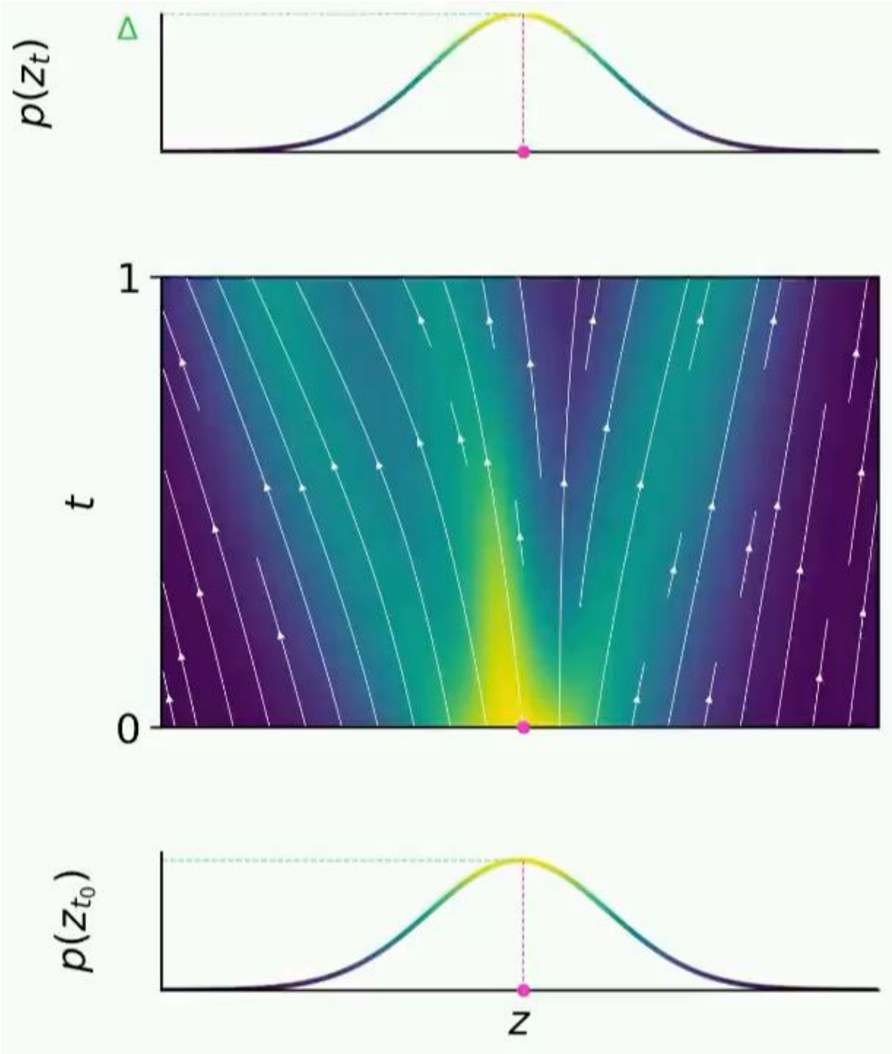
$$\phi_0(\boldsymbol{x}_0) = \boldsymbol{x}_0 \sim p_0$$

$$\phi_1(\boldsymbol{x}_0) = \boldsymbol{x}_1 \sim p_1$$

Continuous Normalizing Flow

Solving the ODE, the flow $\phi_t(\boldsymbol{x}_0)$ can be computed as follows:

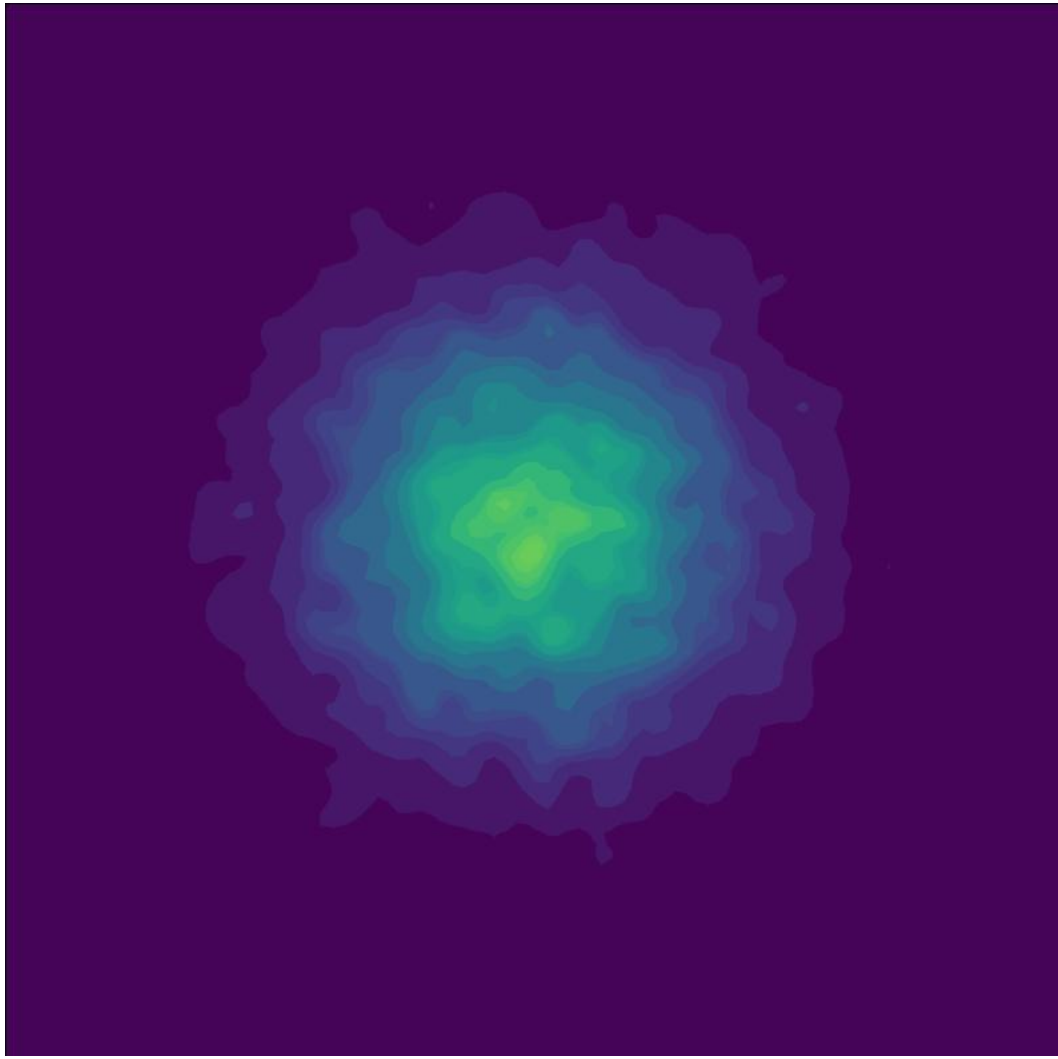
$$\boldsymbol{x}_t = \phi_t(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \int_0^t u_s(\boldsymbol{x}_s)\mathrm{d}s$$



animation credit: Jesse Bettencourt

Continuous Normalizing Flow

- $p_0(\boldsymbol{x}_0)$: Reference distribution.
- $p_1(\boldsymbol{x}_1)$: Data distribution.
- $p_t(\boldsymbol{x}_t)$: **Probability density path (or probability path, for short)**; a time-dependent probability density function.



animation credit: Mark Ogata

Continuous Normalizing Flow

- $\phi_t(\boldsymbol{x}_0)$: Flow map.
- $u_t(\boldsymbol{x}_t)$: Vector field.
- $p_t(\boldsymbol{x}_t)$: Probability path.

Continuous Normalizing Flow

How are the vector field $u_t(\boldsymbol{x}_t)$ and **flow map** $\phi_t(\boldsymbol{x}_0)$ related to the **probability path** $p_t(\boldsymbol{x}_t)$?

Fokker–Planck equation:

$$d\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)dt + g(\boldsymbol{x}_t, t)d\boldsymbol{w}$$

$$\Longrightarrow \frac{\partial p_t(\boldsymbol{x})}{\partial t} = -\nabla \cdot [f(\boldsymbol{x}_t, t)p_t(\boldsymbol{x}_t)] + \frac{1}{2}\nabla^2[g(\boldsymbol{x}_t, t)g(\boldsymbol{x}_t, t)^T p_t(\boldsymbol{x}_t)]$$

Continuous Normalizing Flow

In Fokker–Planck equation, consider

$$\frac{d\boldsymbol{x}_t}{dt} = u_t(\boldsymbol{x}_t) \quad \text{and} \quad g(\boldsymbol{x}_t, t) = 0$$

Then,

$$\frac{\partial p_t(\boldsymbol{x}_t)}{\partial t} = -\nabla \cdot [u_t(\boldsymbol{x}_t) p_t(\boldsymbol{x}_t)]$$

Continuous Normalizing Flow

Based on the instantaneous change of variable[*]:

$$\frac{\partial p_t(\boldsymbol{x}_t)}{\partial t} = -\nabla \cdot [u_t(\boldsymbol{x}_t)p_t(\boldsymbol{x}_t)] \Longrightarrow \frac{\partial \log p_t(\boldsymbol{x}_t)}{\partial t} = -\nabla \cdot u_t(\boldsymbol{x}_t)$$

By solving the ODE, the log probability path $p_t(\boldsymbol{x}_t)$ at arbitrary time $t \in [0,1]$ can be computed as follows:

$$\log p_t(\boldsymbol{x}_t) = \log p_0(\boldsymbol{x}_0) - \int_0^t \nabla \cdot u_s(\boldsymbol{x}_s)\, \mathrm{d}s$$

Flow Matching

- Let's assume we know a **vector field** $u_t(\boldsymbol{x}_t)$ that maximizes the following for every data point $\boldsymbol{x}_1$:

$$\log p_1(\boldsymbol{x}_1) = \log p_0(\boldsymbol{x}_0) - \int_0^1 \nabla \cdot u_t(\boldsymbol{x}_t)\, dt$$

- Then, we can learn such a vector field using a **neural network** $v_\theta(\boldsymbol{x}_t, t)$ by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{t \sim \mathcal{U}[0,1]} \mathbb{E}_{\boldsymbol{x}_t \sim p_t} [\| v_\theta(\boldsymbol{x}_t, t) - u_t(\boldsymbol{x}_t) \|^2]$$

Flow Matching

- Isn't this similar to what we've seen before?

$$\mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\boldsymbol{x}_t\sim p_t}[\|v_\theta(\boldsymbol{x}_t,t)-u_t(\boldsymbol{x}_t)\|^2]$$

- Noise prediction network in diffusion models!

$$\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0),t>1,q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\|\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)-\boldsymbol{\varepsilon}_t\|^2]$$

Continuous Normalizing Flow

How can we know the vector field $u_t(\boldsymbol{x}_t)$?

Given the samples $\boldsymbol{x}$ from the unknown data distribution p_1, now we can consider learning the vector field $u_\theta(\boldsymbol{x}_t, t)$ by minimizing the negative log likelihood:

$$\mathcal{L} = -\mathbb{E}_{\boldsymbol{x}_1 \sim p_1} \log p_1(\boldsymbol{x}_1)$$

$$= -\mathbb{E}_{\boldsymbol{x}_1 \sim p_1} \left[\log p_0(\boldsymbol{x}_0) - \int_0^t \nabla \cdot u_s(\boldsymbol{x}_s) \, \mathrm{d}s \right]$$

Continuous Normalizing Flow

Challenges

The computation of the objective involves **solving an ODE**, which is very computationally expensive and numerically unstable.

Lecture overview

- Notations
- Continuous Normalizing Flow and Flow Matching
- **Conditional Flow Matching**
  - Lipman et al., Flow Matching for Generative Modeling, ICLR 2023.
- Reflow / k-Rectified Flow
- Applications

Conditional Probability Paths

Let's think about a conditional probability path $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ such that:

$$p_t(\boldsymbol{x}_t) = \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)\, p_1(\boldsymbol{x}_1) d\boldsymbol{x}_1$$

$$p_0(\boldsymbol{x}_0|\boldsymbol{x}_1) = \mathcal{N}(0, I)$$

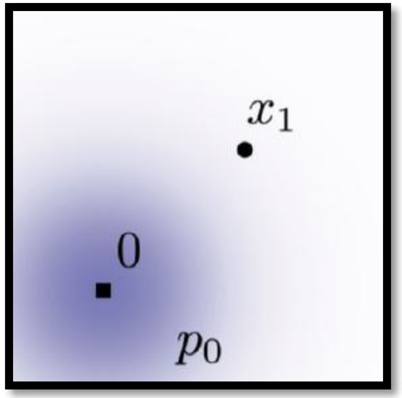
$$p_1(\boldsymbol{x}_1|\boldsymbol{x}_1) = \mathcal{N}\left(\boldsymbol{x}_1, \sigma_{\min}^2 I\right)$$

where $\sigma_{\min}$ is a very small value.

Conditional Probability Paths

$$p_0(\boldsymbol{x}_0|\boldsymbol{x}_1) = \mathcal{N}(0, I)$$

$$p_1(\boldsymbol{x}_1|\boldsymbol{x}_1) = \mathcal{N}\left(\boldsymbol{x}_1, \sigma_{\min}^2 I\right)$$



It can be seen as

- the reference distribution remaining the same, while
- the data distribution is a concentrated Gaussian centered at the given sample.

Lipman et al., Flow Matching for Generative Modeling, ICLR 2023.

Flow Matching

- Isn't the conditional probability path $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ similar to what we've seen before?

- The forward process of diffusion models!

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

  for the Variance Preserving (VP) case.
  (Note that the forward now goes $1 \rightarrow 0$ instead of $0 \rightarrow T$).

Flow Matching

In Flow Matching, we defined the followings:

- $\phi_t(\boldsymbol{x}_0)$: Flow map.
- $u_t(\boldsymbol{x}_t)$: Vector field.
- $p_t(\boldsymbol{x}_t)$: Probability path.

Conditional Flow Matching

Similarly in Conditional Flow Matching, we define the followings:

- $\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1)$: Conditional flow map.
- $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$: Conditional vector field.
- $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$: Conditional probability path.

Conditional Flow Maps

While there are infinite ways to define the **conditinal flow map** $\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1)$, let's consider a simple case:

$$\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_0 + \mu_t(\boldsymbol{x}_1)$$

Conditional Flow Maps

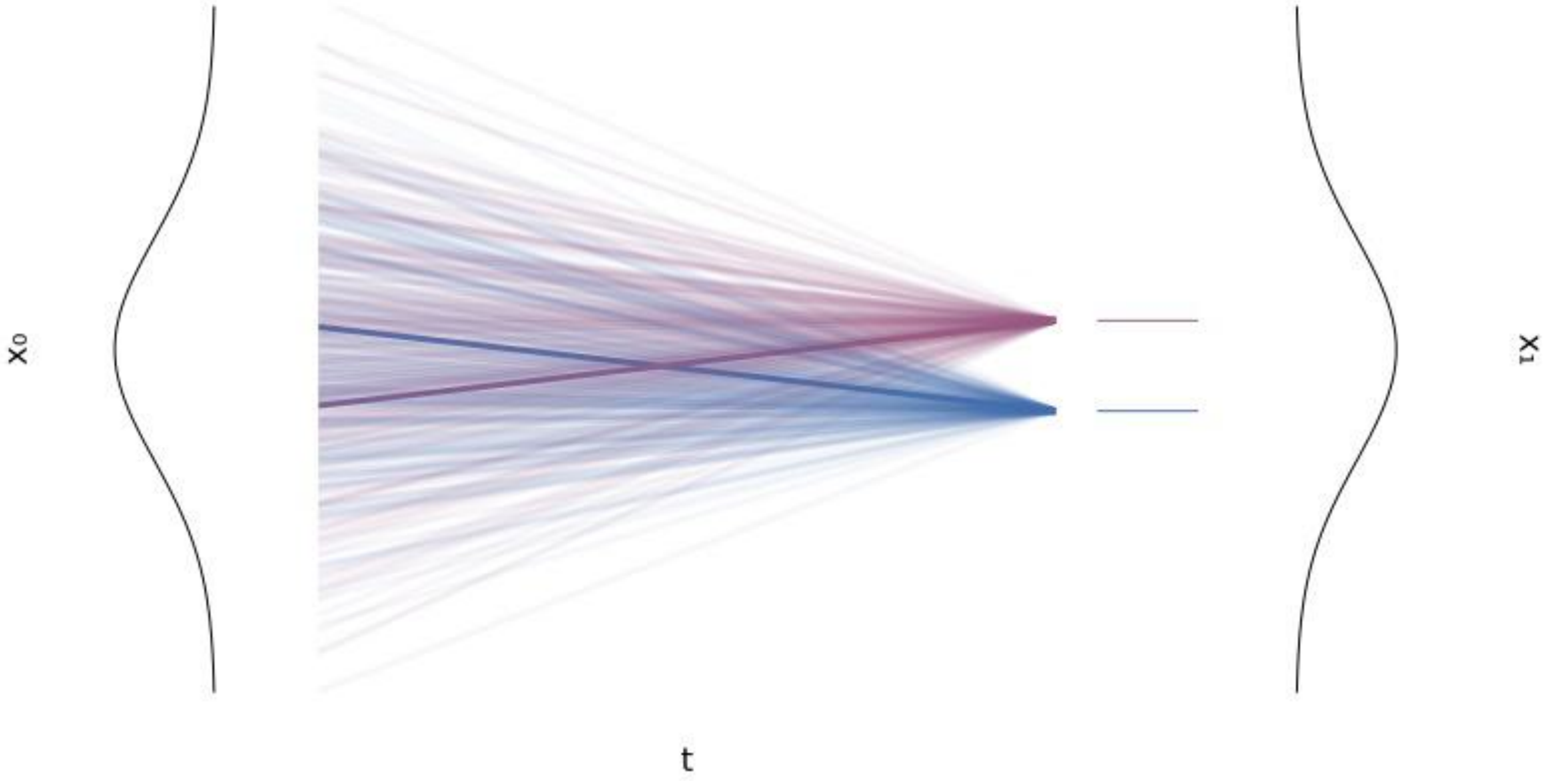


Image credit: Tor Fjelde, Emile Mathieu, Vincent Dutordoir

Conditional Probability Paths

Given the linear conditional flow map $\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1)$, the corresponding **conditional probability path** $p_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ is defined as a Gaussian distribution:

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mathcal{N}(\mu_t(\boldsymbol{x}_1), \sigma_t^2(\boldsymbol{x}_1)I)$$

such that

$$\mu_0(\boldsymbol{x}_1) = 0 \text{ and } \sigma_0(\boldsymbol{x}_1) = 1$$
$$\mu_1(\boldsymbol{x}_1) = \boldsymbol{x}_1 \text{ and } \sigma_1(\boldsymbol{x}_1) = \sigma_{\min}$$

Conditional Probability Paths

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mathcal{N}(\mu_t(\boldsymbol{x}_1), \sigma_t^2(\boldsymbol{x}_1)I)$$

$\mu_0(\boldsymbol{x}_1) = 0$ and $\sigma_0(\boldsymbol{x}_1) = 1$
$\mu_1(\boldsymbol{x}_1) = \boldsymbol{x}_1$ and $\sigma_1(\boldsymbol{x}_1) = \sigma_{\min}$

Then,

$$p_0(\boldsymbol{x}_0|\boldsymbol{x}_1) = p_0(0, I)$$

$$p_1(\boldsymbol{x}_1|\boldsymbol{x}_1) = \mathcal{N}\left(\boldsymbol{x}_1, \sigma_{\min}^2 I\right)$$

Conditional Flow Maps

Check that the following conditinal flow map

$$\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_0 + \mu_t(\boldsymbol{x}_1)$$

with

$$\mu_0(\boldsymbol{x}_1) = 0 \text{ and } \sigma_0(\boldsymbol{x}_1) = 1$$
$$\mu_1(\boldsymbol{x}_1) = \boldsymbol{x}_1 \text{ and } \sigma_1(\boldsymbol{x}_1) = \sigma_{\min}$$

satisfies the boundary conditions:

$$\phi_0(\boldsymbol{x}_0|\boldsymbol{x}_1) = \boldsymbol{x}_0$$

$$\phi_1(\boldsymbol{x}_0|\boldsymbol{x}_1) \approx \boldsymbol{x}_1$$

Conditional Vector Fields

How about the conditional vector field $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$?

Theorem[*]. Given the conditinal flow map

$$\phi_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_0 + \mu_t(\boldsymbol{x}_1)$$

the unique conditional vector field $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ is defined as:

$$u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \frac{\sigma_t'(\boldsymbol{x}_1)}{\sigma_t(\boldsymbol{x}_1)}\left(\boldsymbol{x}_t - \mu_t(\boldsymbol{x}_1)\right) + \mu_t'(\boldsymbol{x}_1)$$

*Chen et al., Neural Ordinary Differential Equations, NeurIPS 2018

Conditional Flow Matching

Conditional Flow Map:

$$\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1) = \sigma_t(\boldsymbol{x}_1)\boldsymbol{x}_0 + \mu_t(\boldsymbol{x}_1)$$

Conditional Probability Path:

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \mathcal{N}(\mu_t(\boldsymbol{x}_1), \sigma_t^2(\boldsymbol{x}_1)I)$$

Conditional Vector Field:

$$u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \frac{\sigma_t'(\boldsymbol{x}_1)}{\sigma_t(\boldsymbol{x}_1)}\left(\boldsymbol{x}_t - \mu_t(\boldsymbol{x}_1)\right) + \mu_t'(\boldsymbol{x}_1)$$

Option 1: Linear Trajectory (Optimal Transport)

Let's think about the simplest option:

$$\mu_t(\boldsymbol{x}_1) = t\boldsymbol{x}_1$$

$$\sigma_t(\boldsymbol{x}_1) = 1 - (1 - \sigma_{\min})t \approx 1 - t$$

How are the followings defined?

- **Conditional Probability Path** $p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$
- **Conditional Flow Map** $\phi_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$
- **Conditional Vector Field** $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$

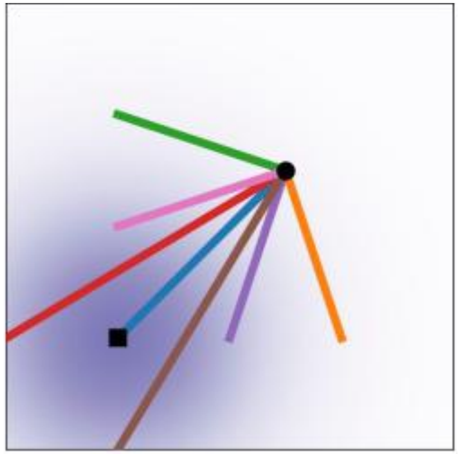
Option 1: Linear Trajectory (Optimal Transport)

Conditional Probability Path:

$$p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1) \approx \mathcal{N}(t\boldsymbol{x}_1, (1-t)I)$$

Conditional Flow Map:

$$\phi_t(\boldsymbol{x}_0|\boldsymbol{x}_1) \approx (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$$



Conditional Vector Field:

$$u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) \approx \frac{\boldsymbol{x}_1 - \boldsymbol{x}_t}{1-t}$$

The conditional flow map is defined as a linear interpolation!

Option 2: Diffusion Trajectory

Recall that $p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$ was also defined as the forward process in diffusion!

If the conditional probability path $p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$ is defines in the same way as VP-diffusion:

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

what should the conditional vector field $u_t(x_t|x_1)$ be?

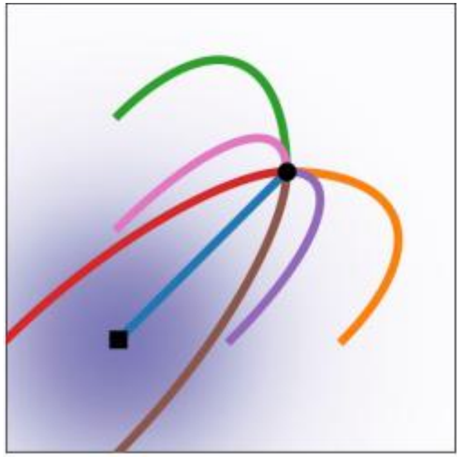
Option 2: Diffusion Trajectory

Conditional Probability Path:

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

Conditional Vector Field:

$$u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) = \frac{\alpha'_{1-t}(\alpha_{1-t}\boldsymbol{x}_t - \boldsymbol{x}_1)}{1-\alpha_{1-t}^2}$$



The conditional flow map follows the same trajectory as that of diffusion models.

Conditional Flow Matching

- In the Conditional Normalizing Flow (CNF), we generally don't have access to a closed-form vector field $u_t(\cdot)$. Also, learning $u_t(\cdot)$ is very computationally expensive and numerically unstable.

- In constrast, in Conditional Flow Matching (CMF), since we specify $p_1(\boldsymbol{x}_1|\boldsymbol{x}_1) = \mathcal{N}\left(\boldsymbol{x}_1, \sigma_{\min}^2 I\right)$, an appropriate conditional vector field $u_t(\cdot|\boldsymbol{x}_1)$ can be defined in closed form.

Conditional Flow Matching

What can we do with the conditional vector field $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$?

Important Fact 1.

The marginal vector field $u_t(\boldsymbol{x}_t)$ can be defined based on the conditional vector field $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$ as follows:

$$u_t(\boldsymbol{x}_t) = \int u_t(\boldsymbol{x}_t|\boldsymbol{x}_1) \frac{p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)p_1(\boldsymbol{x}_1)}{p_t(\boldsymbol{x}_t)} d\boldsymbol{x}_1$$

Conditional Flow Matching

Important Fact 2.

The loss of the flow matching

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\boldsymbol{x}_t\sim p_t}[\|v_\theta(\boldsymbol{x}_t,t) - u_t(\boldsymbol{x}_t)\|^2]$$

can be replaced with an equivalent loss regressing the conditional vector field:

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\boldsymbol{x}_1\sim p_1,\boldsymbol{x}_t\sim p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)}[\|v_\theta(\boldsymbol{x}_t,t) - u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)\|^2]$$

Conditional Flow Matching

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\boldsymbol{x}_1\sim p_1,\boldsymbol{x}_t\sim p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)}[||v_\theta(\boldsymbol{x}_t,t) - u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)||^2]$$

While we don't have access to a closed-form $u_t(x_t)$, we do have one for the conditional case $u_t(x_t|x_1)$!

Diffusion Model Training

Here, we matched the generative time domain to the interval $0 \rightarrow 1$.

1. Sample a data point $\boldsymbol{x}_1 \sim p_1$.
2. Sample a timestep $t \sim \mathcal{U}[0, 1]$.
3. Sample noise $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
4. Compute $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\varepsilon}_t$ (same as sampling $\boldsymbol{x}_t$).
5. Take gradient descent step on the **score matching loss**.

Conditional Flow Matching Model Training

1. Sample a data point $\boldsymbol{x}_1 \sim p_1$.
2. Sample a timestep $t \sim \mathcal{U}[0, 1]$.
3. Sample $\boldsymbol{x}_t \sim p_{t|1}(\boldsymbol{x}_t|\boldsymbol{x}_1)$.
4. Compute the conditional velocity $u_t(\boldsymbol{x}_t|\boldsymbol{x}_1)$.
5. Take gradient descent step on the velocity matching loss.

Generation

1. Sample a standard Gaussian sample $\boldsymbol{x}_0 \sim p_0$.
2. With the initial condition $\boldsymbol{x}_0$,
   solve the ODE using the learned vector field $v_\theta(\boldsymbol{x}_t, t)$
   to compute $\boldsymbol{x}_1$:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \int_0^1 v_\theta(\boldsymbol{x}_t, t)\mathrm{d}t$$

Comparison

	CIFAR-10			ImageNet 32×32			ImageNet 64×64		
Model	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**

Lipman et al., Flow Matching for Generative Modeling, ICLR 2023.

Conditioned/Guided Generation

- CFG
- ControlNet
- LoRA
- Inversion-based editing
- Attention-map-based editing

Lecture overview

- Notations
- Continuous Normalizing Flow and Flow Matching
- Conditional Flow Matching
- **Reflow / k-Rectified Flow**
  - Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.
  - Liu et al., Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, ICLR 2023.
- Applications

Terminology

- A conditional flow matching model will be simply called a **flow model**.

- The linear trajectory flow model is called a **rectified flow** model.

Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

Flow Model Training

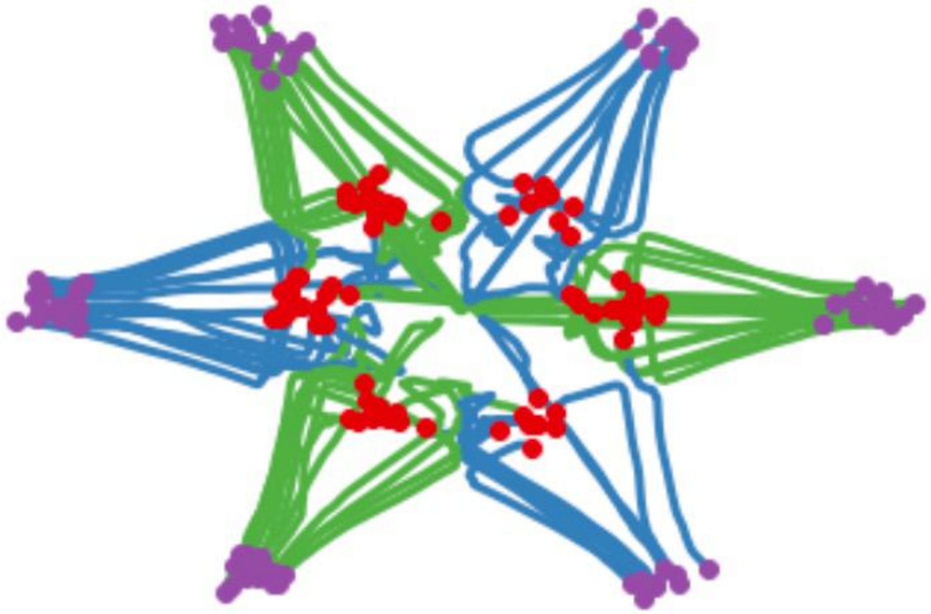
1. Sample a data point $\boldsymbol{x}_1 \sim p_1$
   and a standard Gaussian sample $\boldsymbol{x}_0 \sim p_0$.
2. Sample a timestep $t \sim \mathcal{U}[0, 1]$.
3. Compute $\boldsymbol{x}_t = \phi_t(\boldsymbol{x}_0 | \boldsymbol{x}_1)$.
4. Compute the conditional velocity $u_t(\boldsymbol{x}_t | \boldsymbol{x}_1)$.
5. Take gradient descent step on the conditional flow matching loss.

Conditional Flow Matching Model Training

When training a flow model,
random $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ pairs are sampled.

Reflow / k-Rectified Flow

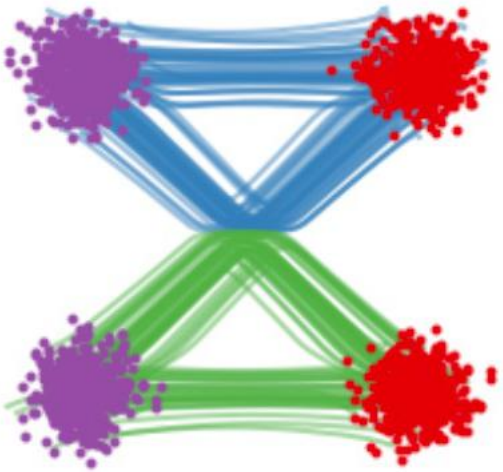
- Assume that there is a pre-trained a diffusion/flow model.
- The trajectories may not be straight.



Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

Reflow / k-Rectified Flow

- Assume that there is a **pre-trained** a diffusion/flow model.
- The trajectories may not be straight, **even with** a **rectified flow** model.

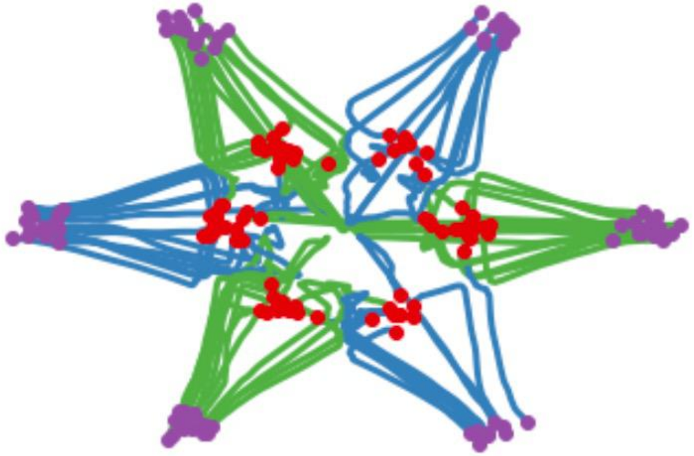


Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

Reflow / k-Rectified Flow

What if we sample $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ pairs from
a **pretrained** diffusion/flow model
to train another **rectified flow model**?

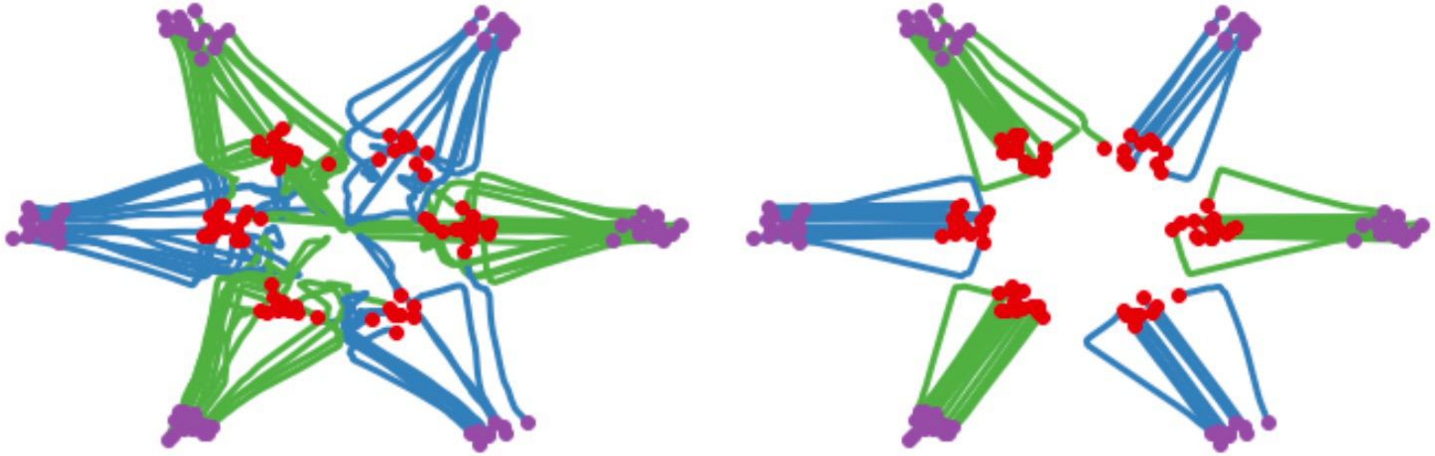
Reflow / k-Rectified Flow



Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

2-Rectified Flow

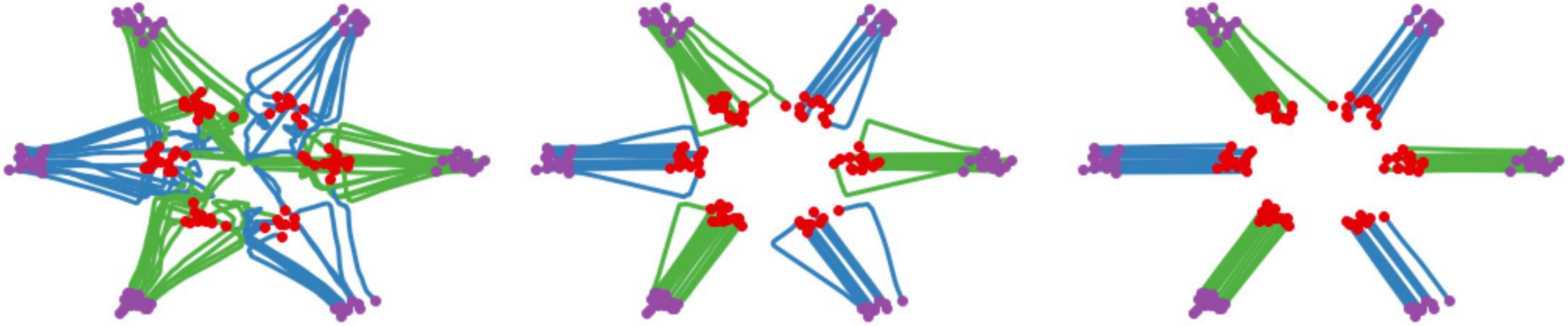
Now we get straighter trajectories!



Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

3-Rectified Flow

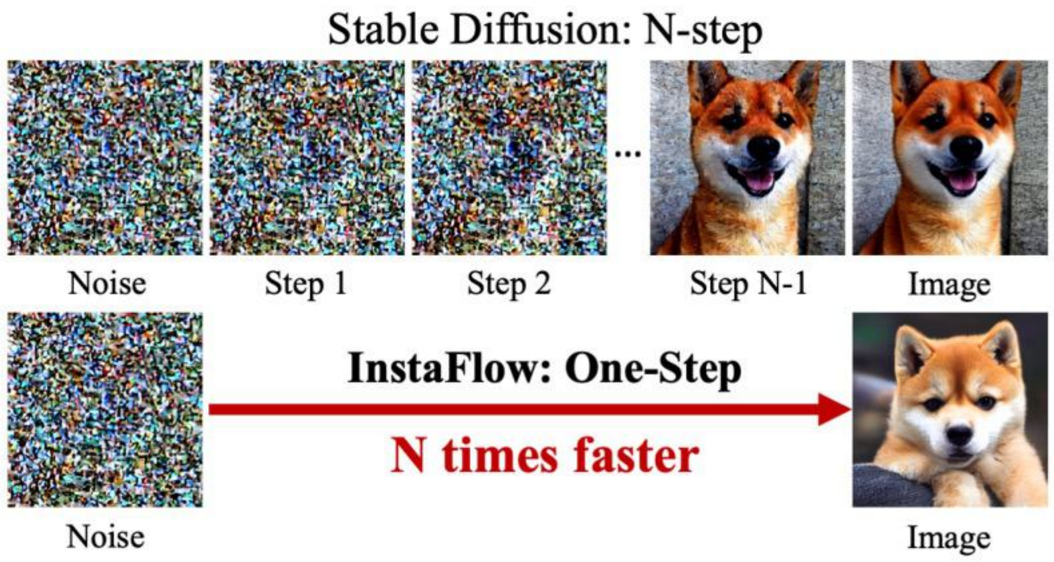
Repeat one more time. Even straighter trajectories.

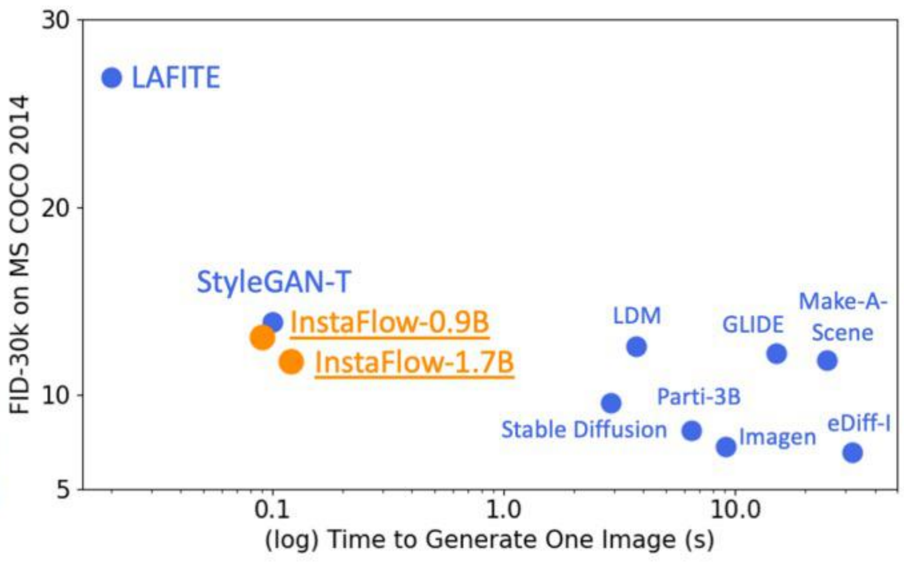


Liu et al., Rectified Flow: A Marginal Preserving Approach to Optimal Transport, 2022.

Reflow / k-Rectified Flow

Straighter trajectories enable **faster generation**, even in a **single step**!





Liu et al., InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation, ICLR 2024.

Rival

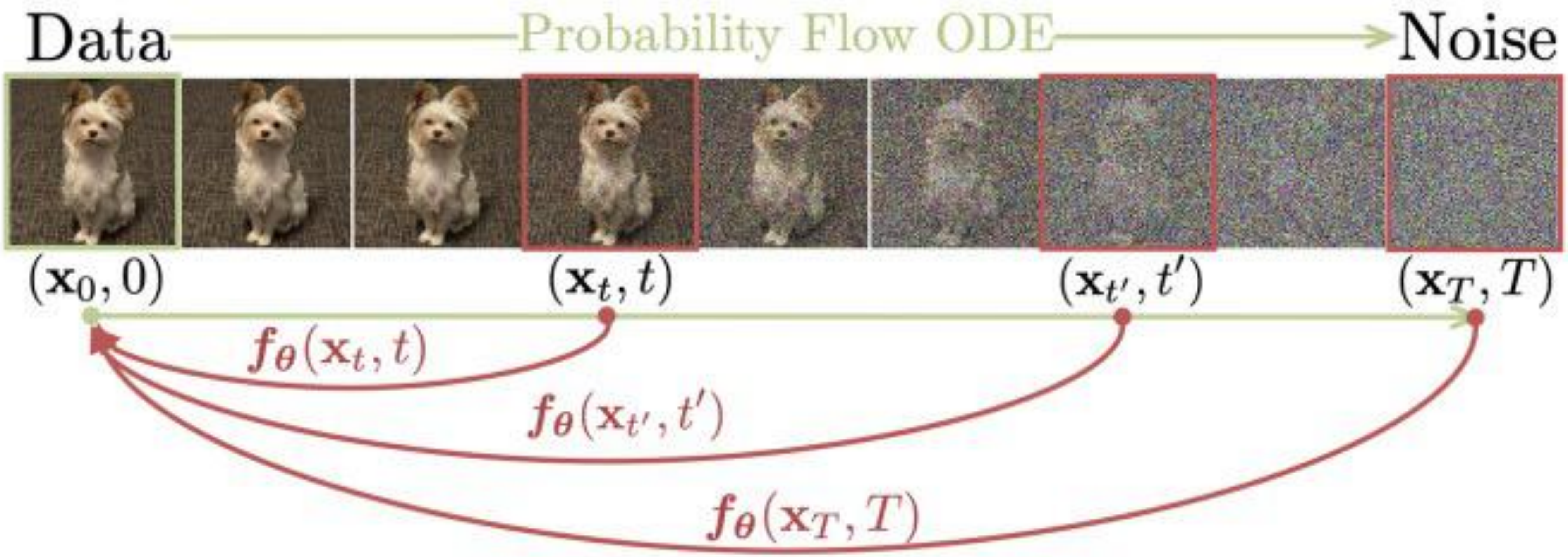
Consistency model

vs.

Reflow

Consistency Models

Given a solution trajectory $\{\mathbf{x}_t\}_{t\in[0,T]}$, consistency models f_θ learn to **map any $\mathbf{x}_t$ directly to its final clean sample $\mathbf{x}_0$.**

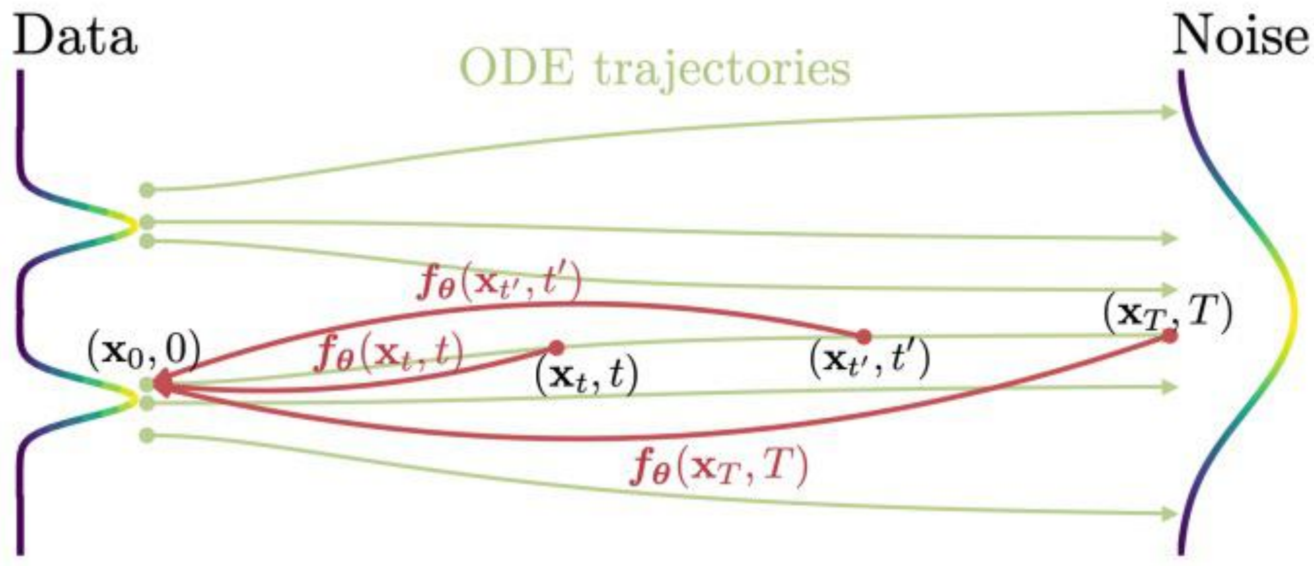


Song et al., Consistency Models, ICML 2023

Consistency Loss

In order to map any $\mathbf{x}_t$ along the trajectory to the same $\mathbf{x}_0$, it's trained with the following consistency loss:

$$\mathcal{L}_{Consistency} = \|f_\theta(\mathbf{x}_t, t) - f_\theta(\mathbf{x}_{t+1}, t+1)\|$$



Song et al., Consistency Models, ICML 2023

Slide credit: Juil Koo

Comparison with Diffusion Models

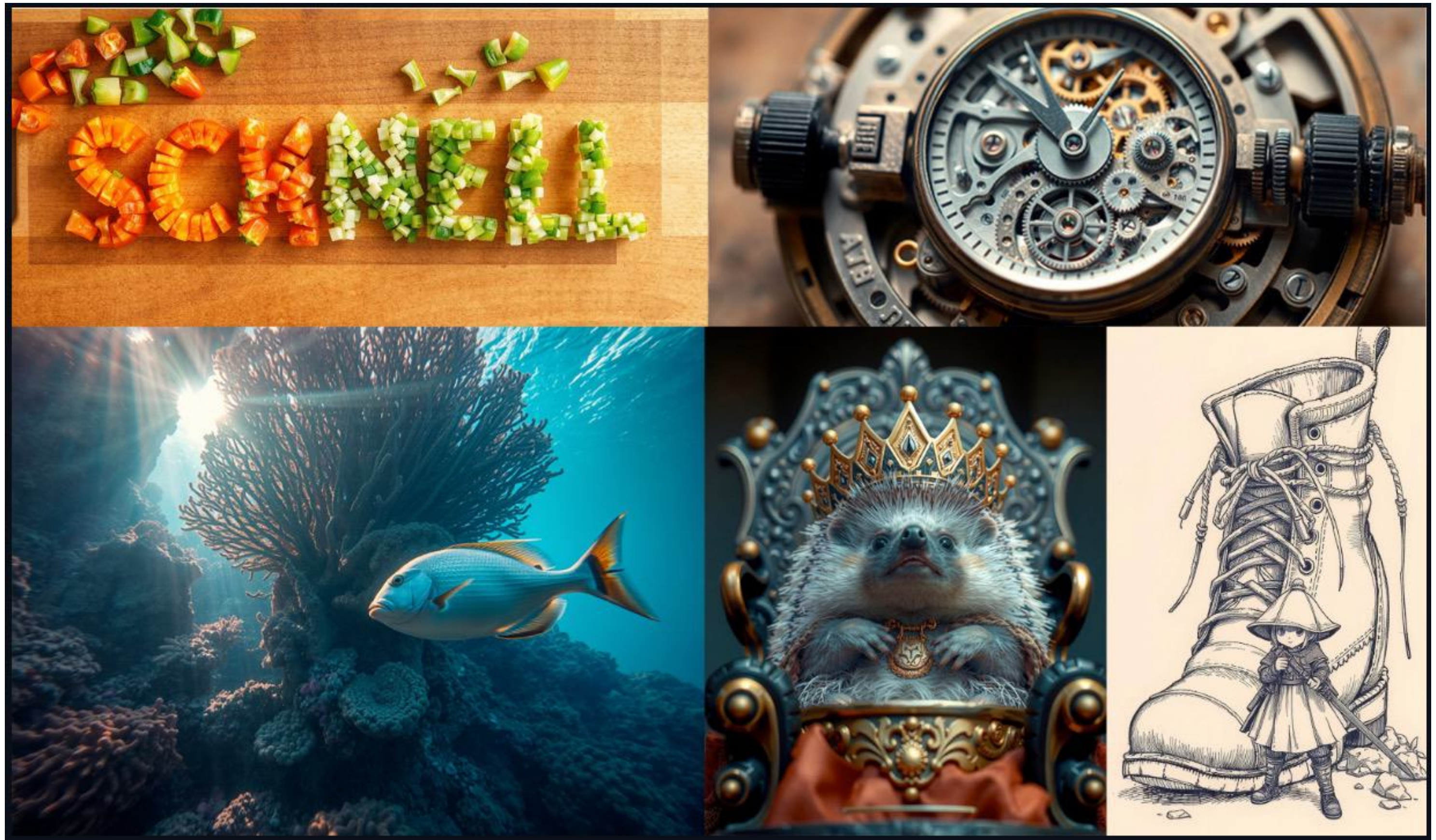
Consistency Models
($S = 1$)

Comparison with Consistency Models

Consistency models				
CD (LPIPS) [Song et al., 2023]	1	6.20	0.68	0.63
	2	4.70	0.69	0.64
	3	4.32	0.70	0.64
CT (LPIPS) [Song et al., 2023]	1	13.0	0.71	0.47
	2	11.1	0.69	0.56
iCT [Song and Dhariwal, 2023]	1	4.02	0.70	0.63
	2	3.20	0.73	0.63
iCT-deep [Song and Dhariwal, 2023]	1	3.25	0.72	0.63
	2	2.77	0.74	0.62
CTM + GAN [Kim et al., 2023]	1	**1.92**		0.57
	2	**1.73**		0.57
Rectified flows				
2-rectified flow++ (ours)	1	4.31		
	2	3.64		

Lee et al., Improving the Training of Rectified Flows, NeurIPS 2024.

Example: Flux.1 Schnell



`FLUX.1 [schnell]` is a 12 billion parameter rectified flow transformer capable of generating images from text descriptions. For more information, please read our blog post.

https://huggingface.co/black-forest-labs/FLUX.1-schnell

GPT 4o Failure Cases

GPT 4o

"Four drums, seven tomatoes, and five candles."

GPT 4o Failure Cases

GPT 4o

"In a room, all the chairs are occupied except one."

Guided Generation Using Flow Models

GPT 4o

FLUX + Inference-Time Scaling

"Four drums, seven tomatoes, and five candles."

Kim et al. Inference-Time Scaling for Flow Models via Stochastic Generation and Rollover Budget Forcing, NeurIPS 2025.

Guided Generation Using Flow Models

GPT 4o

FLUX + Inference-Time Scaling

“In a room, all the chairs are occupied except one.”

Kim et al. Inference-Time Scaling for Flow Models via Stochastic Generation and Rollover Budget Forcing, NeurIPS 2025.

Diffusion vs. Flow

Diffusion	Flow
Score matching	Velocity matching
SDE	ODE
Diffusion trajectories	Linear trajectories
Consistency model	Reflow

Lecture overview

- Notations
- Continuous Normalizing Flow and Flow Matching
- Conditional Flow Matching
- Reflow / k-Rectified Flow
- **Applications**

Guided Generation Using Flow Models

Min et al., ORIGEN: Zero-Shot 3D Orientation Grounding in Text-to-Image Generation, NeurIPS 2025.

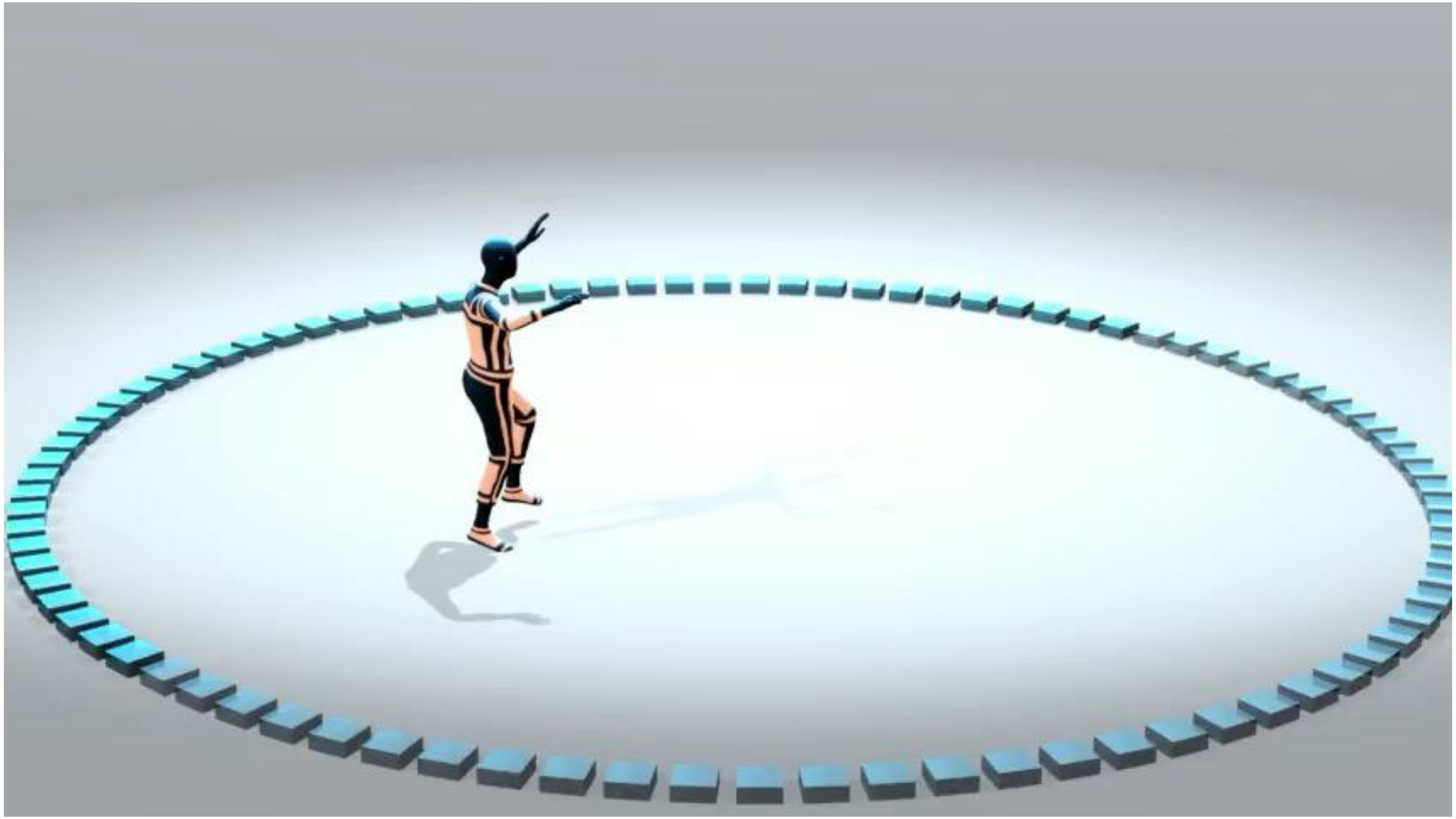
Guided Generation Using Flow Models

Yoon et al., Psi-Sampler: Initial Particle Sampling for SMC-Based Inference-Time Reward Alignment in Score Models, NeurIPS 2025.

Audio Generation

https://audioldm.github.io/audioldm2/

Motion Generation

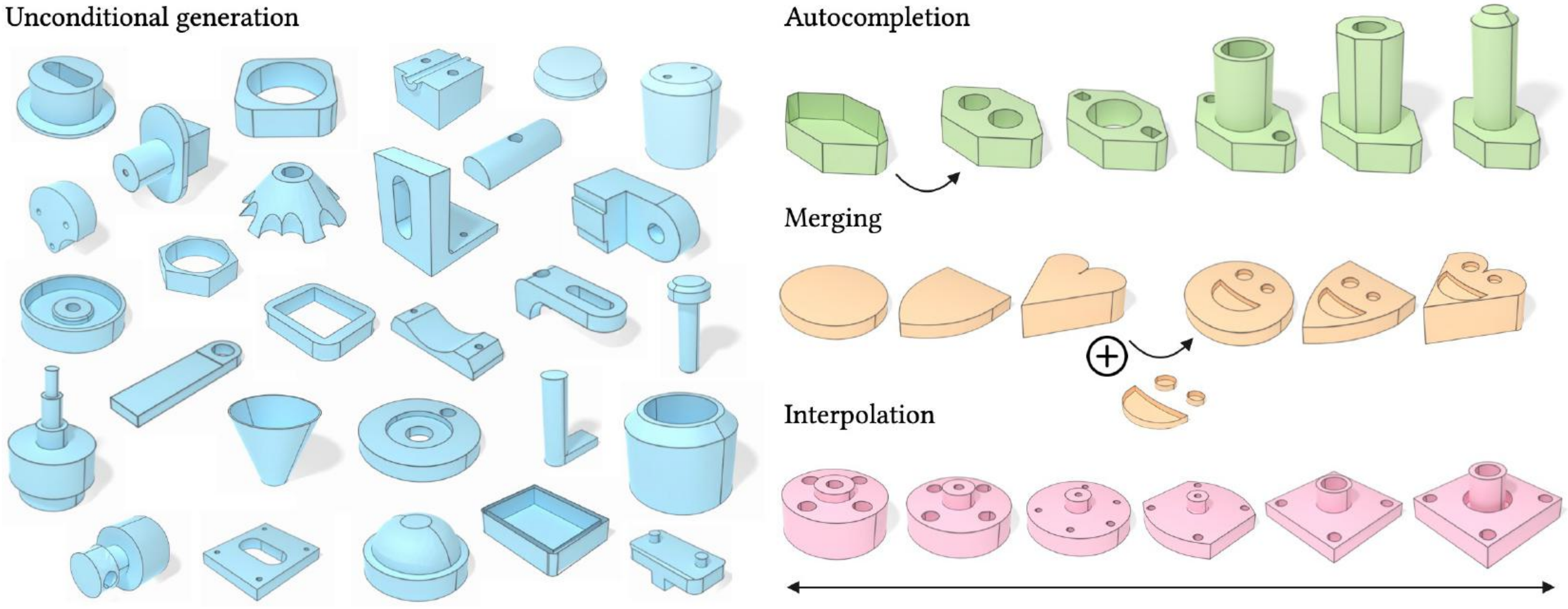


Alexanderson et al., Listen, Denoise, Action! Audio-Driven Motion Synthesis with Diffusion Models, SIGGRAPH 2023.

3D Generation

Zhang et al., BANG: Dividing 3D Assets via Generative Exploded Dynamics, SIGGRAPH 2025.

CAD Generation



Lee et al., BrepDiff: Single-stage B-rep Diffusion Model, SIGGRAPH 2025.

Next lecture:
Video Generation