113-2 Embodied Vision

Diffusion Models and Flow Matching

Jia-Wei Liao

Ph.D. Candidate in Computer Science

National Taiwan University

Learning Resources for Diffusion Models

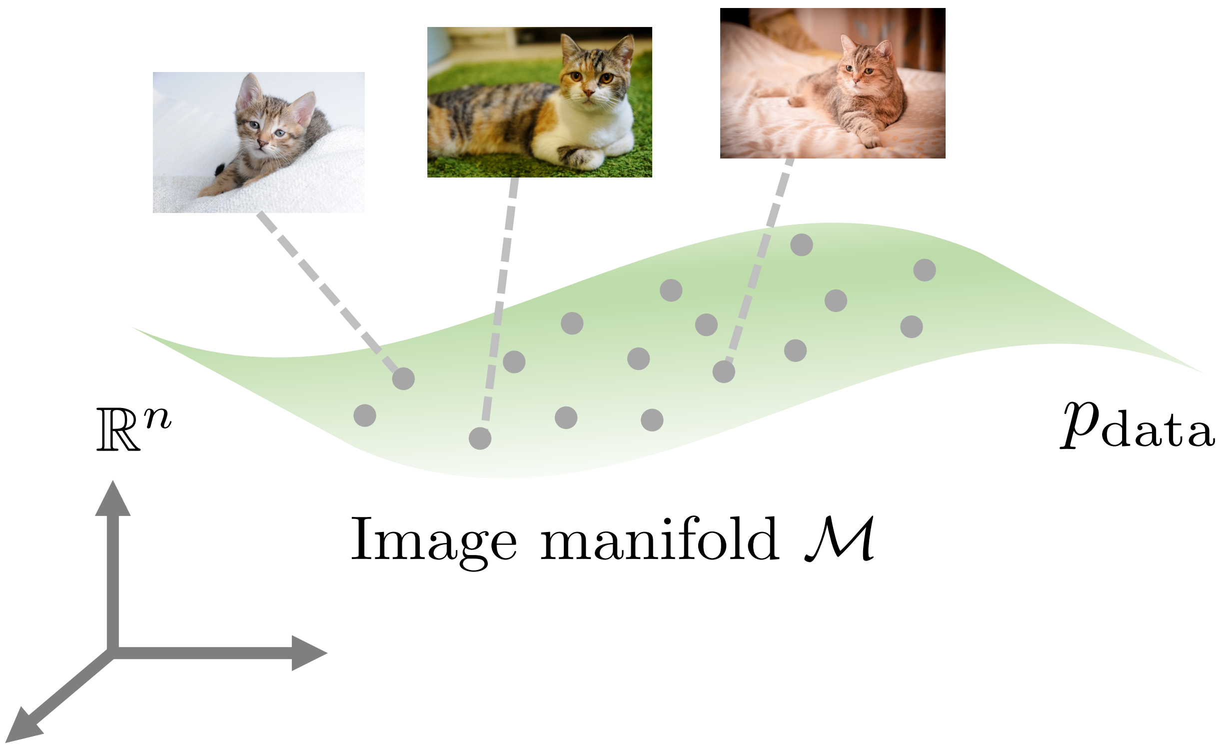
- 【漫士科普】人工智慧博士生告訴你 SORA 擴散模型究竟是怎麼產生影片的？
- Hung-Yi Lee YouTube
- Jia-Bin Huang YouTube
- Diffusion and Score-Based Generative Models (Yang Song)
- Evolution of Diffusion Models: From Birth to Enhanced Efficiency and Controllability (Jesse)
- Lil'Log What are Diffusion Models?
- 生成擴散模型漫談 (蘇劍林)

Outline

- Theoretical Foundations (ODE/SDE, Score Function)
- Training and Sampling
- Advanced Variants (DDIM, CM, CTM, Flow Matching, Rectified Flow)
- Controllability and Applications (CG, CFG, ControlNet)

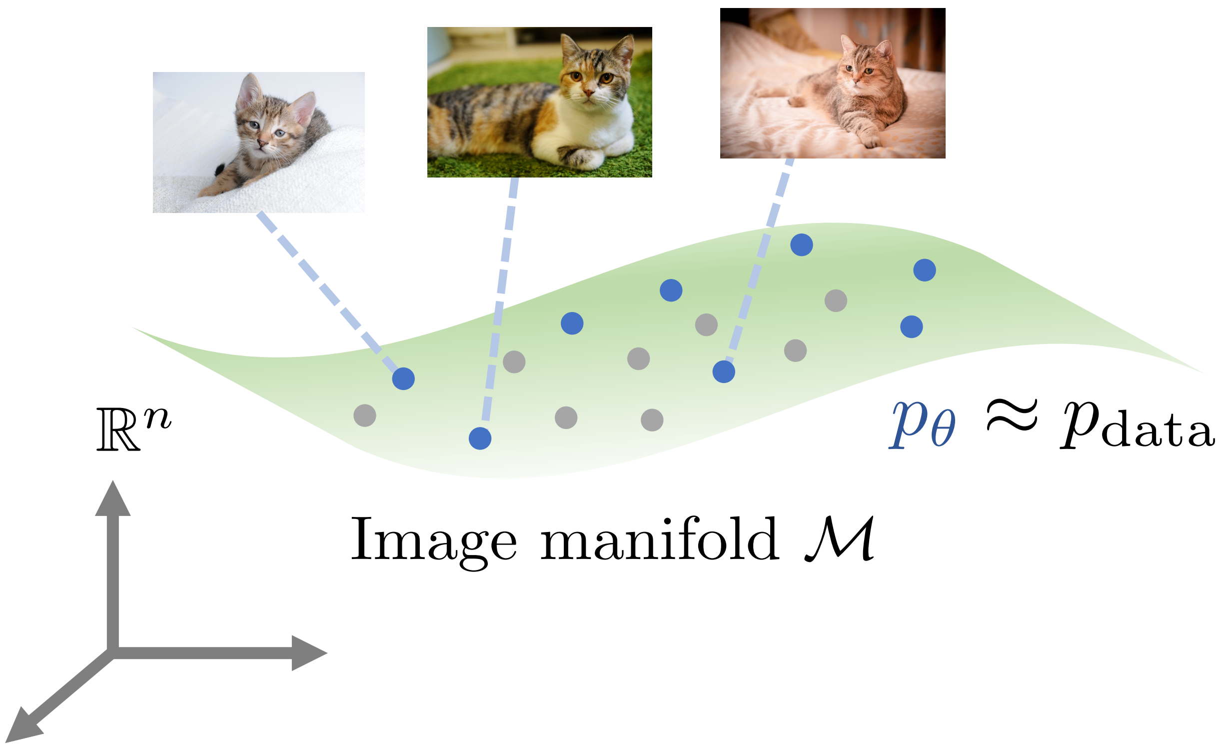
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional manifold



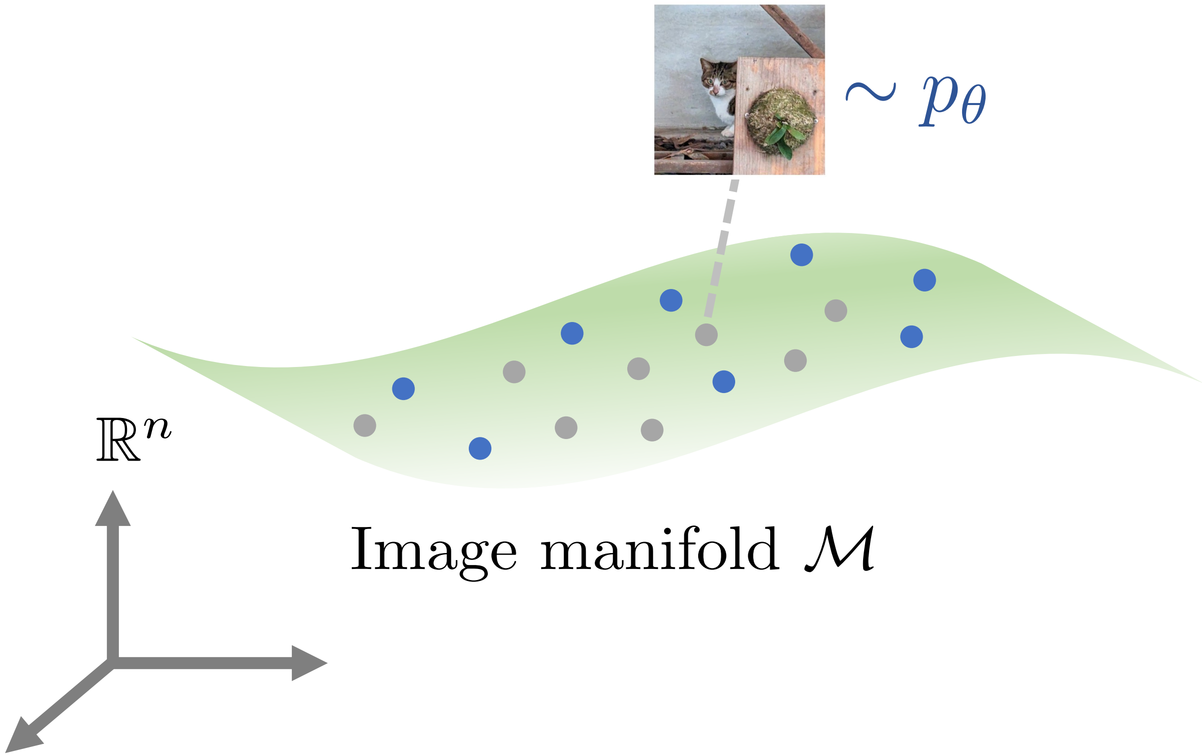
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional manifold

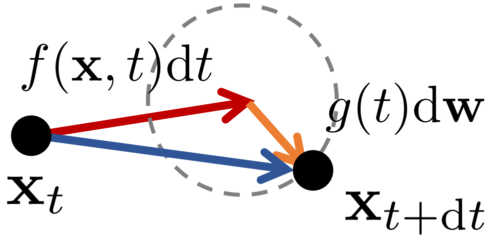


What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional manifold

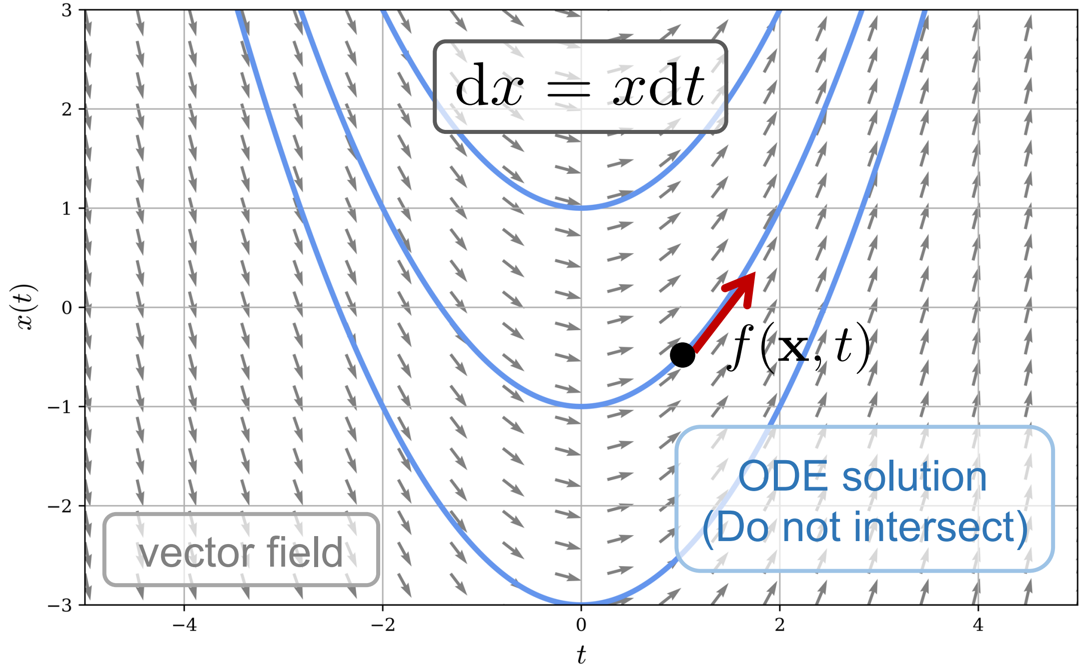


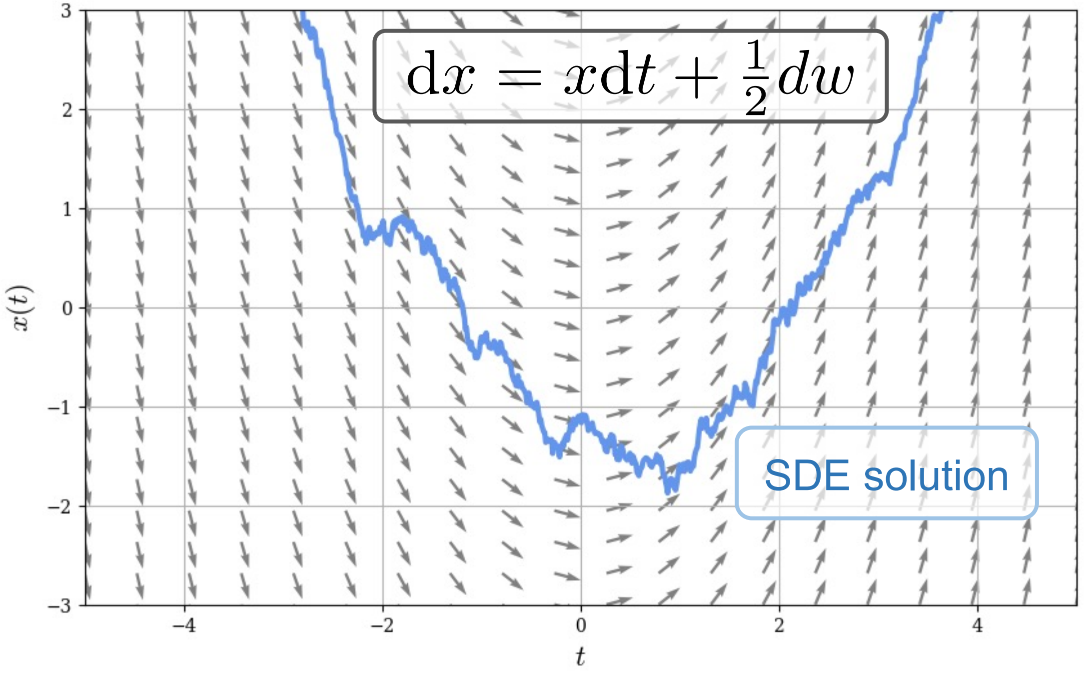
Recap ODE and SDE



$\mathrm{d}\mathbf{x} = f(\mathbf{x},t)\mathrm{d}t$ or $\dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = f(\mathbf{x},t)$

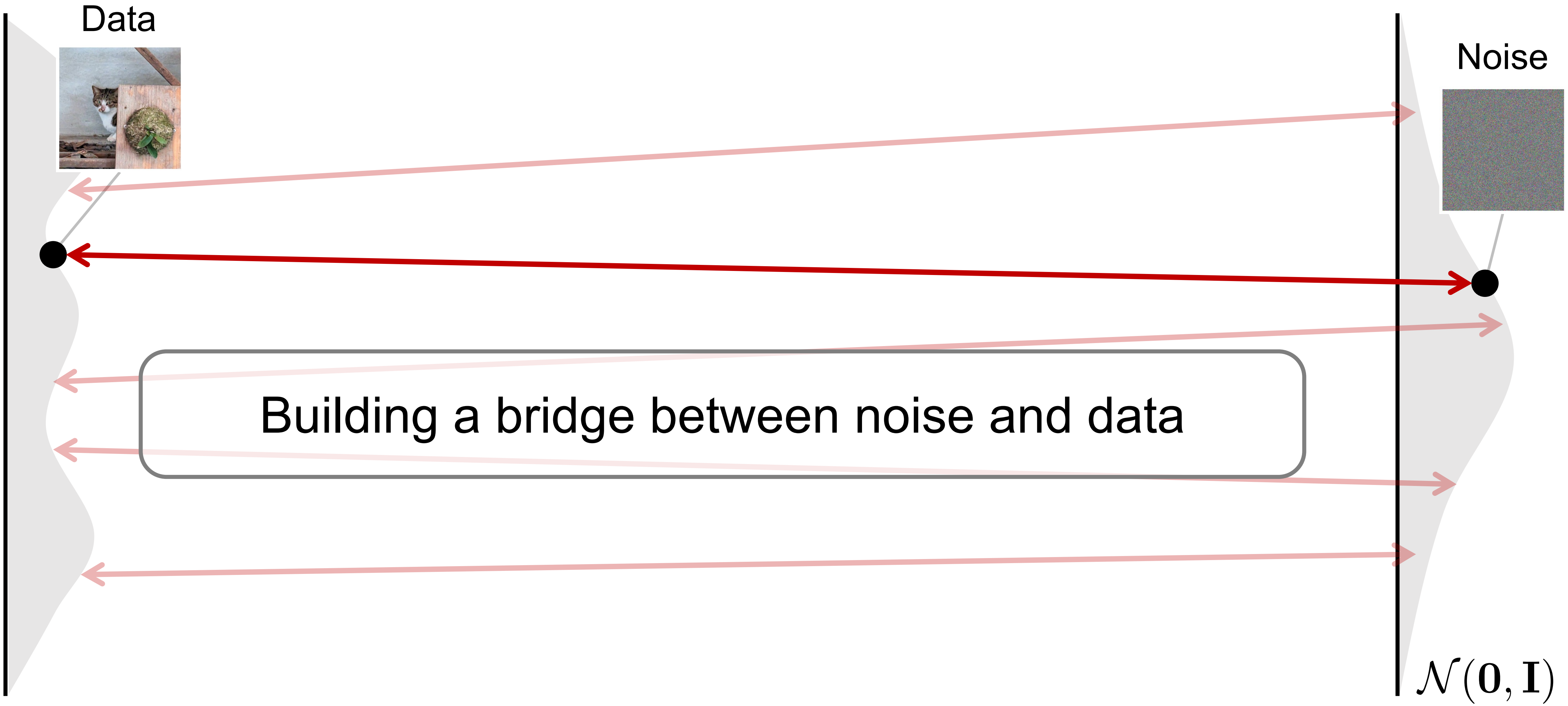
$$\mathrm{d}\mathbf{x} = \underbrace{f(\mathbf{x},t)\mathrm{d}t}_{\text{drift}} + \underbrace{g(t)\mathrm{d}\mathbf{w}}_{\text{diffusion}}$$





The arrow indicates the direction of movement at the moment of downward shift

The Goal of Diffusion Models



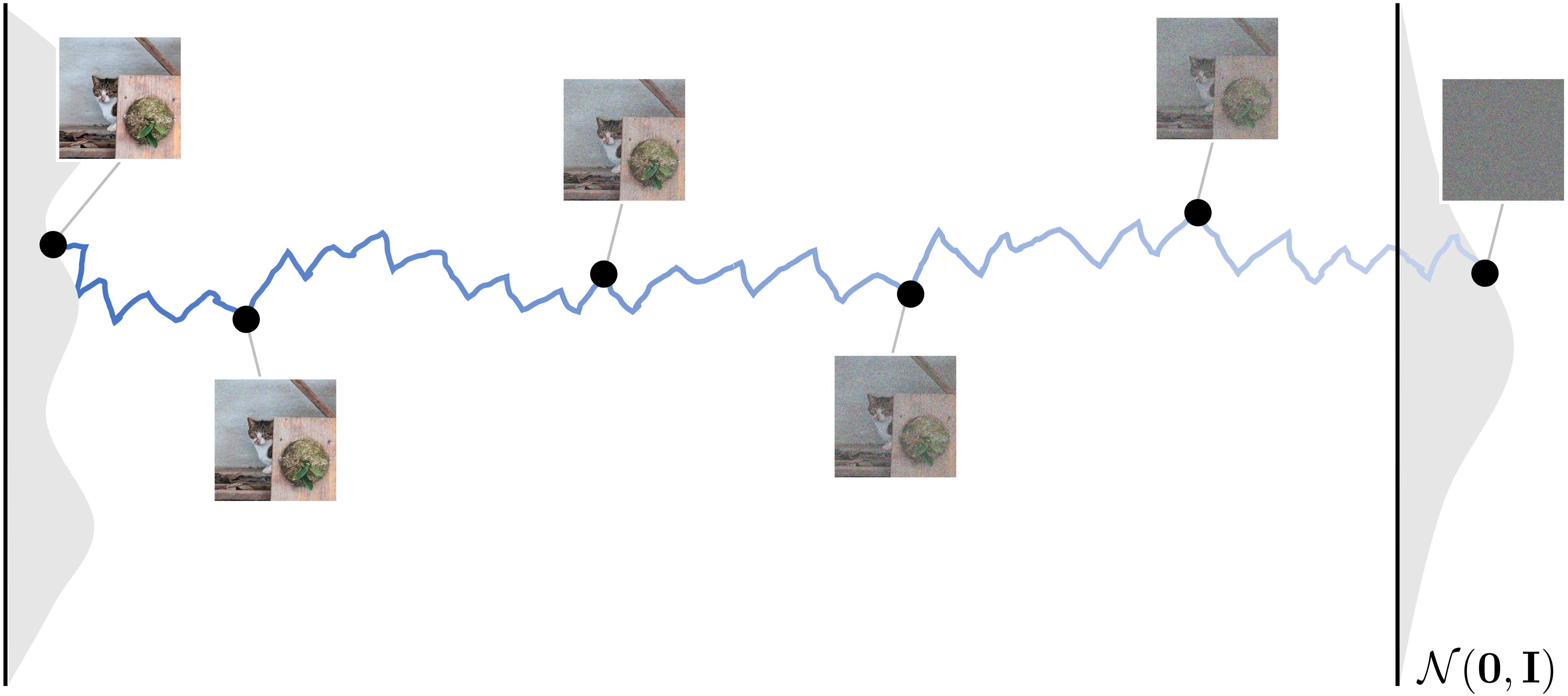
What is Diffusion Model?

Forward Process: add noise step by step, from data to pure noise

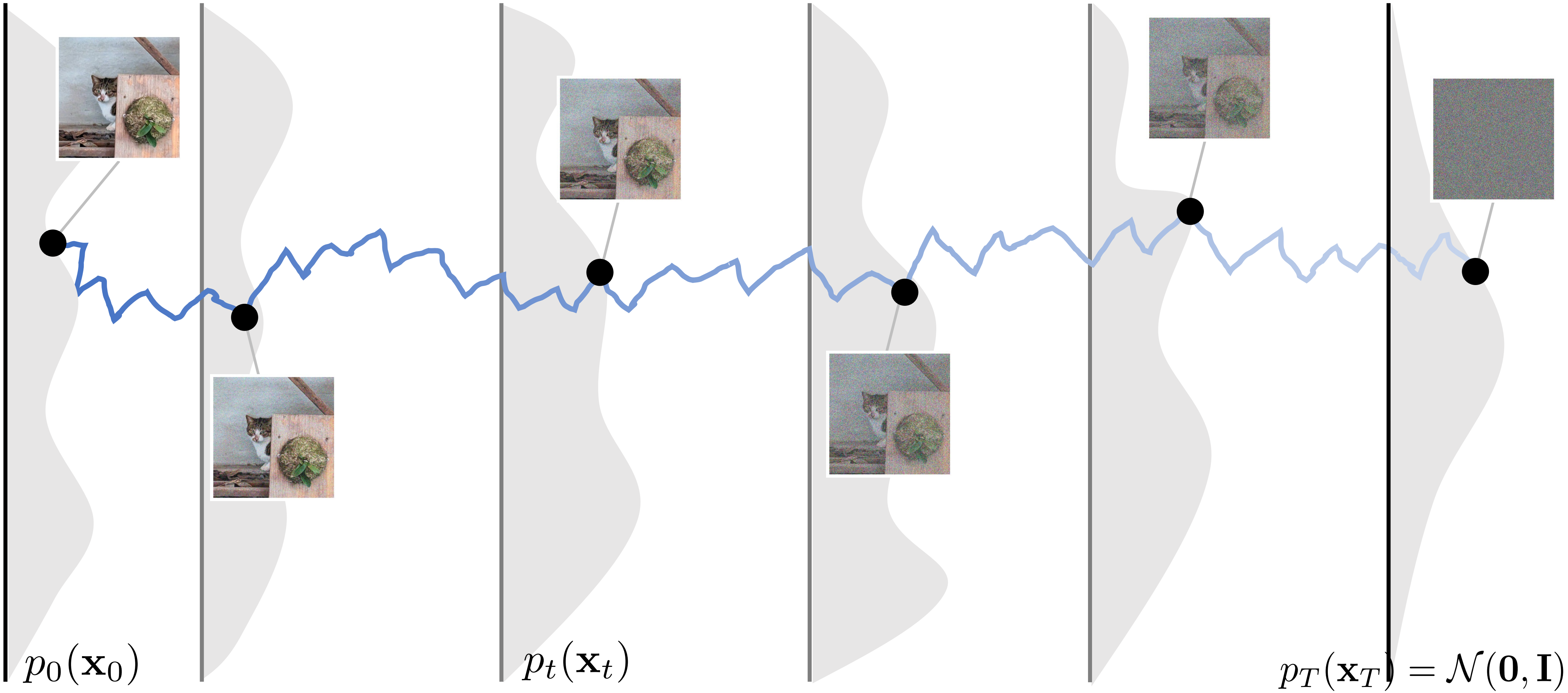
Reverse Process: generate data from pure noise by denoising

Creating noise from data is easy; creating data from noise is generative modeling – *Yang Song*

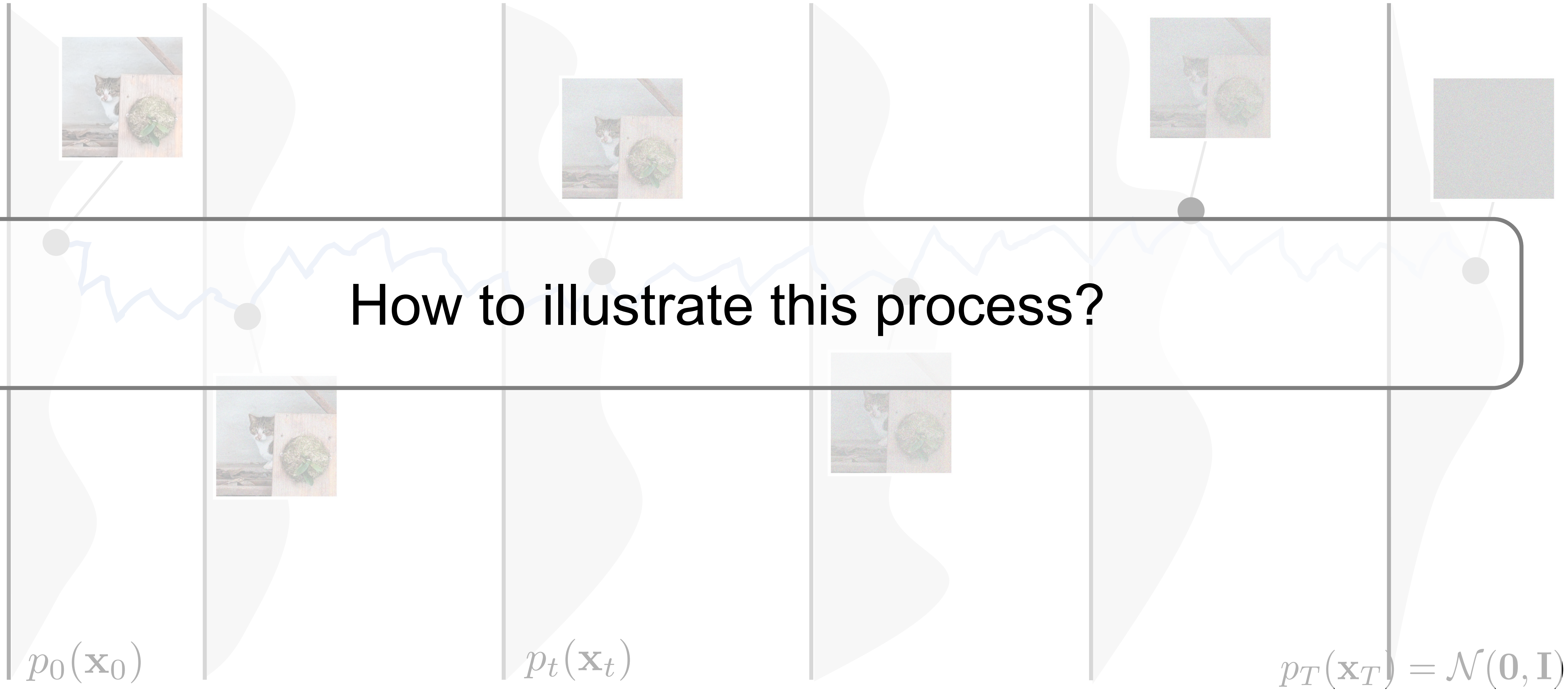
Diffusion Models



Diffusion Models

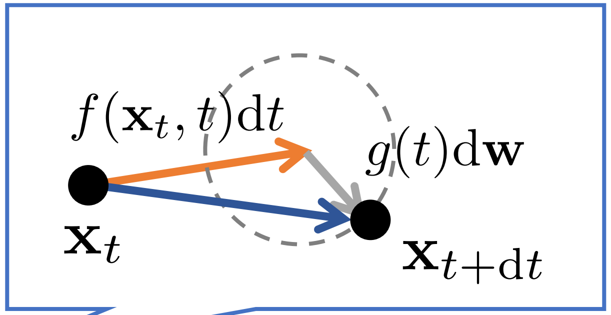


Diffusion Models



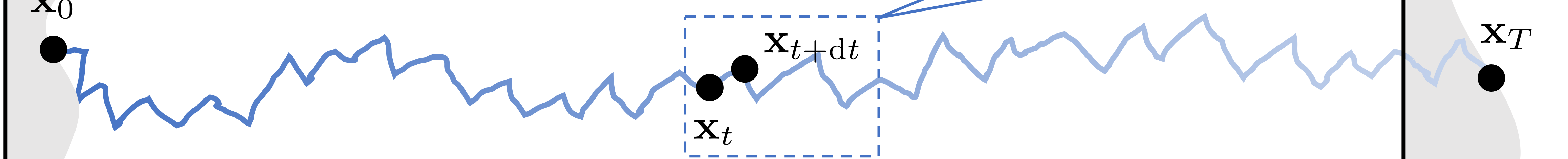
How to illustrate this process?

Score-based Diffusion Models [Song+ ICLR'21]



Forward SDE

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



Reverse SDE (Stochastic)

$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

Score function

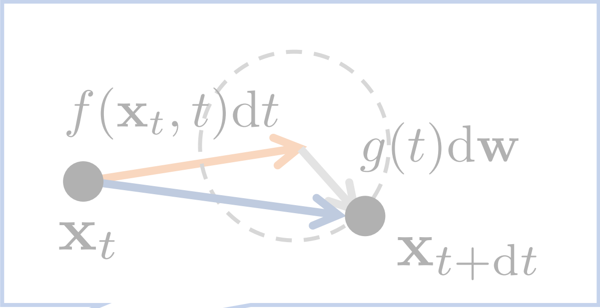
Reverse ODE (Deterministic)

$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - \frac{1}{2}g(t)^2 \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}]\mathrm{d}t$$

Have the same $\{p_t(\mathbf{x}_t)\}_{t=0}^{T}$

$\mathcal{N}(\mathbf{0}, \mathbf{I})$

Score-based Diffusion Models [Song+ ICLR'21]



Forward SDE

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

$\mathbf{x}_0$ $\mathbf{x}_{t-\mathrm{d}t}$ $\mathbf{x}_t$ $\mathbf{x}_{t+\mathrm{d}t}$ $\mathbf{x}_T$

What is intuition of score function?

Reverse SDE (stochastic)

$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

Score function

Reverse ODE (deterministic)

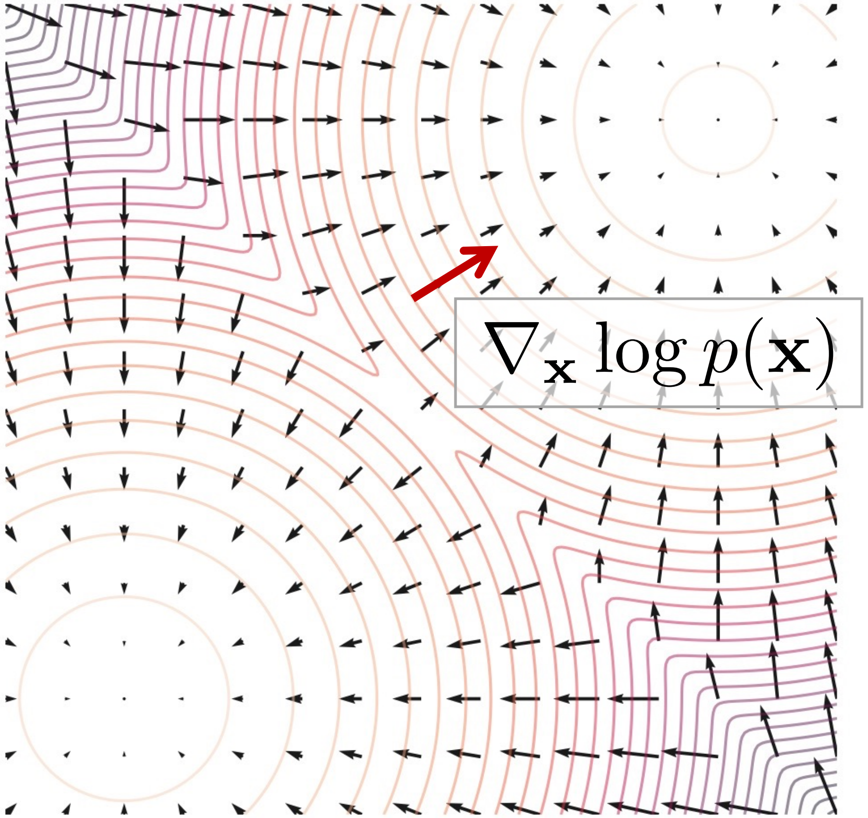
$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}]\mathrm{d}t$$

Have the same $\{p_t(\mathbf{x}_t)\}_{t=0}^T$

$\mathcal{N}(\mathbf{0}, \mathbf{I})$

Score Function

Probability Density Function $p(\mathbf{x})$



Score Function: The direction in which the probability density increases most rapidly

Score-based Diffusion Models [Song+ ICLR'21]

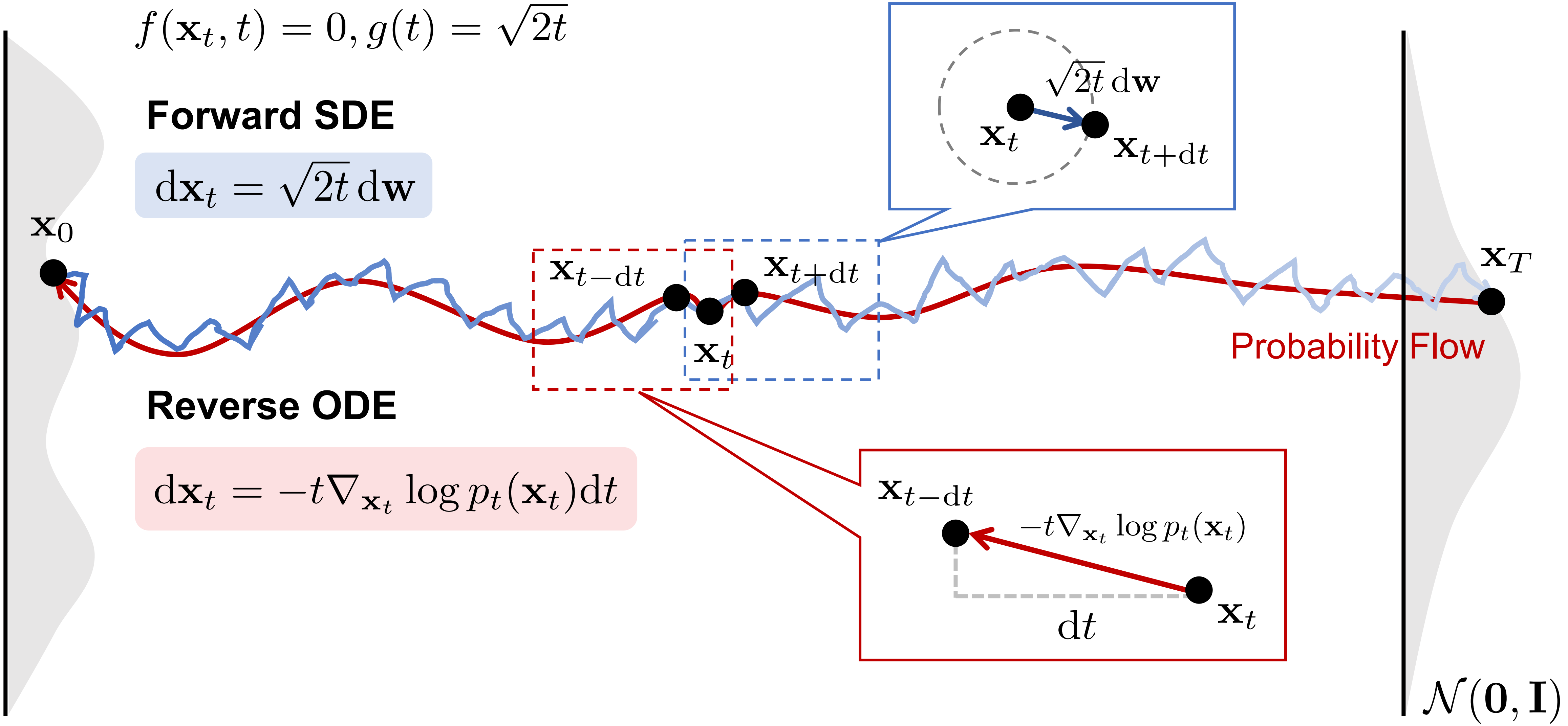
$f(\mathbf{x}_t, t) = 0, g(t) = \sqrt{2t}$

Forward SDE

$$\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}$$

Reverse ODE

$$\mathrm{d}\mathbf{x}_t = -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\mathrm{d}t$$

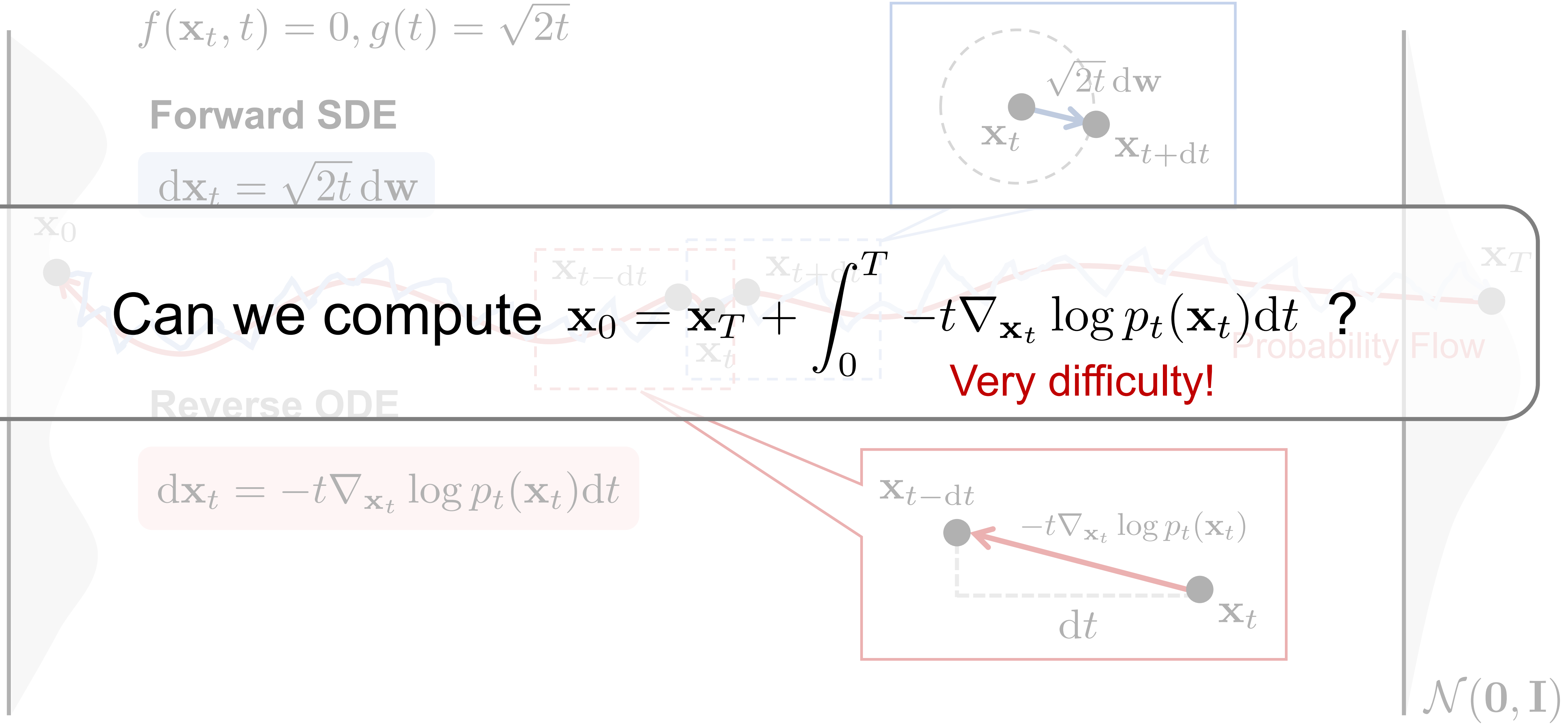


Score-based Diffusion Models [Song+ ICLR'21]

$f(\mathbf{x}_t, t) = 0, g(t) = \sqrt{2t}$

Forward SDE

$$d\mathbf{x}_t = \sqrt{2t}\, d\mathbf{w}$$



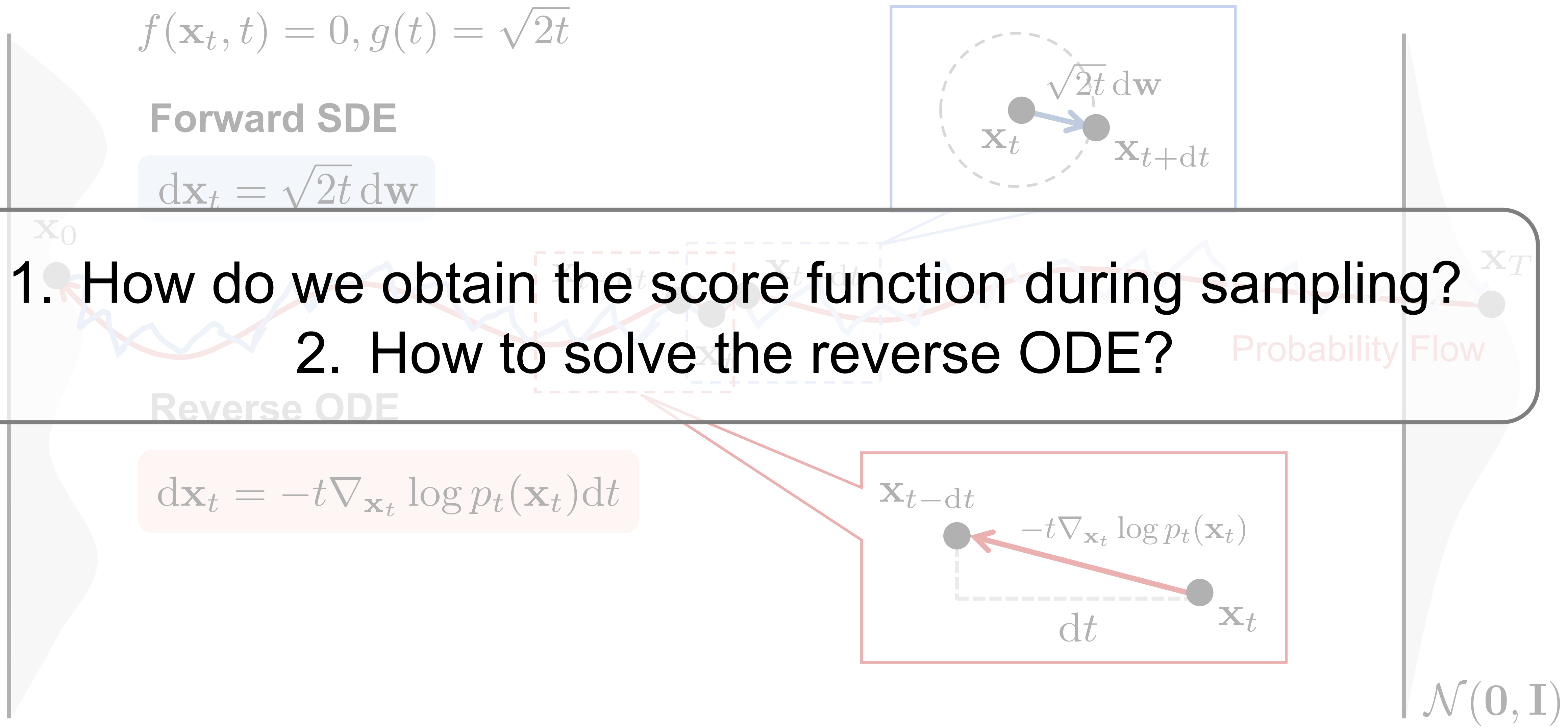
Can we compute $\mathbf{x}_0 = \mathbf{x}_T + \int_0^T -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) dt$?

Very difficulty!

Reverse ODE

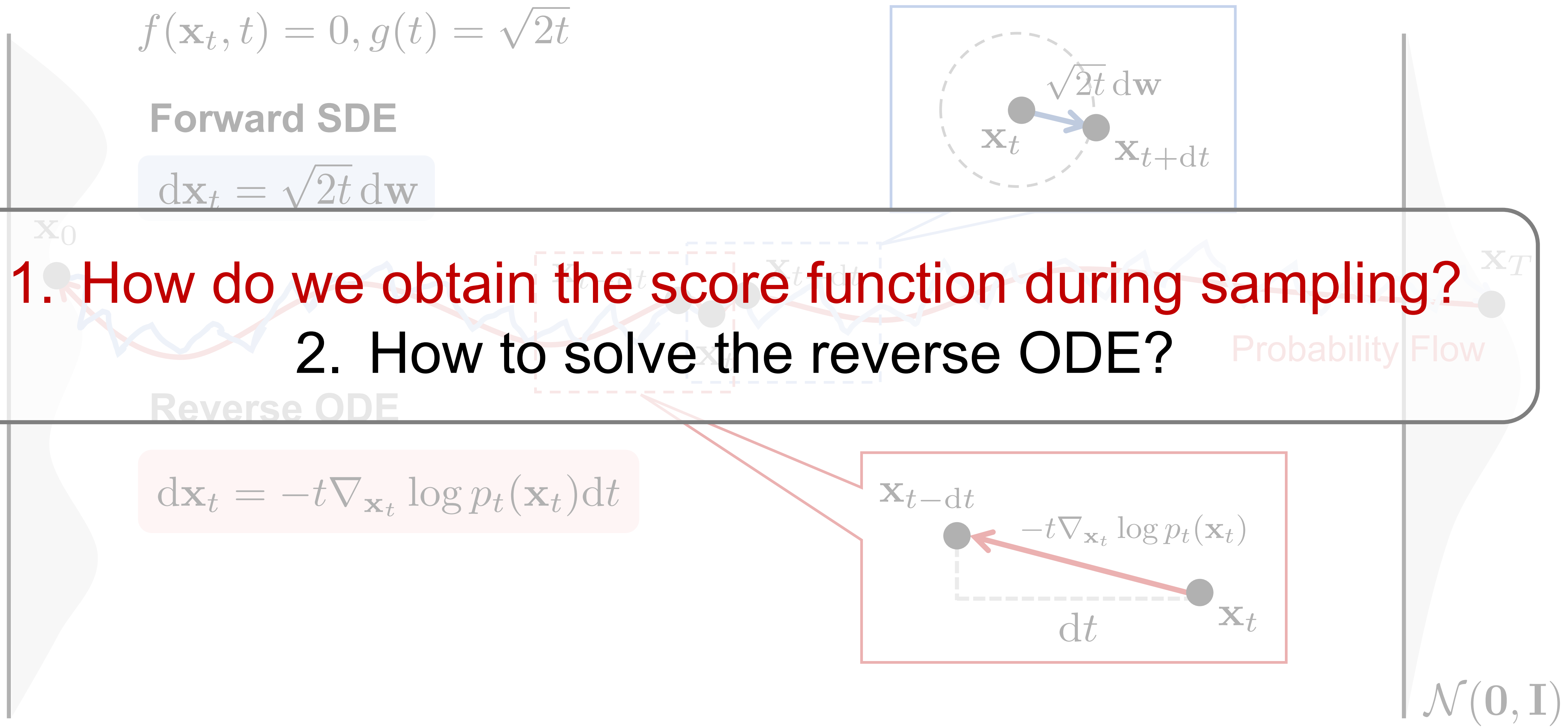
$$d\mathbf{x}_t = -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) dt$$

Score-based Diffusion Models [Song+ ICLR'21]



1. How do we obtain the score function during sampling?
2. How to solve the reverse ODE?

Score-based Diffusion Models [Song+ ICLR’21]



1. How do we obtain the score function during sampling?
2. How to solve the reverse ODE?

Training Score-based Diffusion Models

We train the model to approximate the score function

$$\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$$

During sampling, we use the model to predict the score and plug it into SDE/ODE

$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \boxed{\mathbf{s}_\theta(\mathbf{x}_t, t)}]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

The training objective can be expressed as

$$\mathcal{L}_{\mathrm{SM}}(\mathbf{x}_0; \theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2$$ (Score Matching)

Training Score-based Diffusion Models

We train the model to approximate the score function

$$\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$$

During sampling, we use the model to predict the score and plug it into SDE (ODE)

Can we easily compute the score function during training?

$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \boxed{\mathbf{s}_\theta(\mathbf{x}_t, t)}]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

The training objective can be expressed as

$$\mathcal{L}_{\mathrm{SM}}(\mathbf{x}_0; \theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2 \quad \text{(Score Matching)}$$

Training Score-based Diffusion Models

Score Matching (SM) [Hyvarinen JMLR’05]**:**

$$\mathcal{L}_{\mathrm{SM}}(\mathbf{x}_0;\theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0}\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\|_2^2$$

Denoising Score Matching (DSM) [Vincent Neural Comput.’11]**:**

$$\mathcal{L}_{\mathrm{DSM}}(\mathbf{x}_0;\theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0}\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \boxed{\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)}\|_2^2$$

$$\|$$

$$\frac{\mathbf{x}_0 - \mathbf{x}_t}{\sigma_t^2}$$

noise level

$$+\epsilon_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2\mathbf{I})$$

$$p_t(\mathbf{x}_t|\mathbf{x}_0)$$

$$\mathbf{x}_0 \sim p_{\mathrm{data}} \qquad \mathbf{x}_t$$

Score-based Diffusion Models [Song+ ICLR'21]

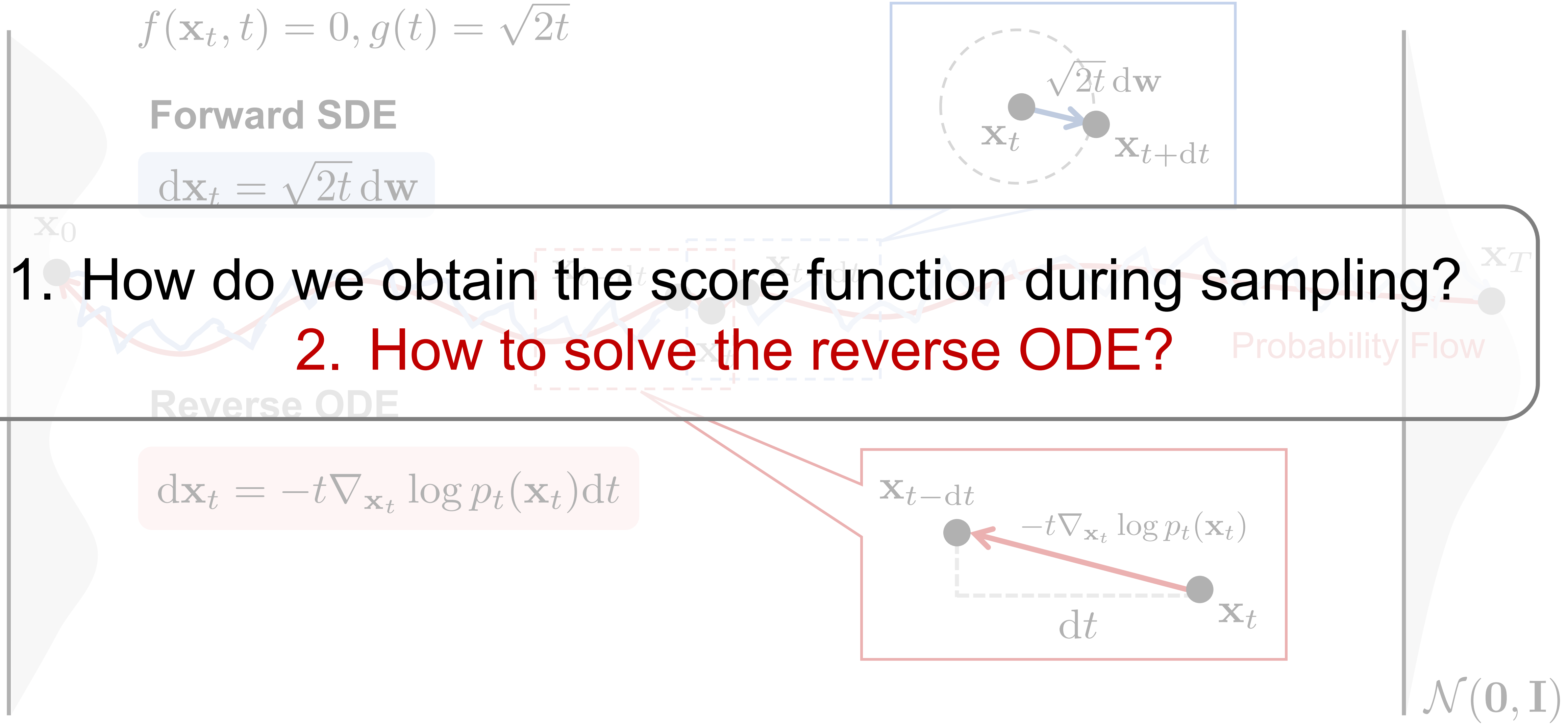
$f(\mathbf{x}_t, t) = 0, g(t) = \sqrt{2t}$

Forward SDE

$\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}$

Reverse ODE

$\mathrm{d}\mathbf{x}_t = -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\mathrm{d}t$



1. How do we obtain the score function during sampling?
2. How to solve the reverse ODE?

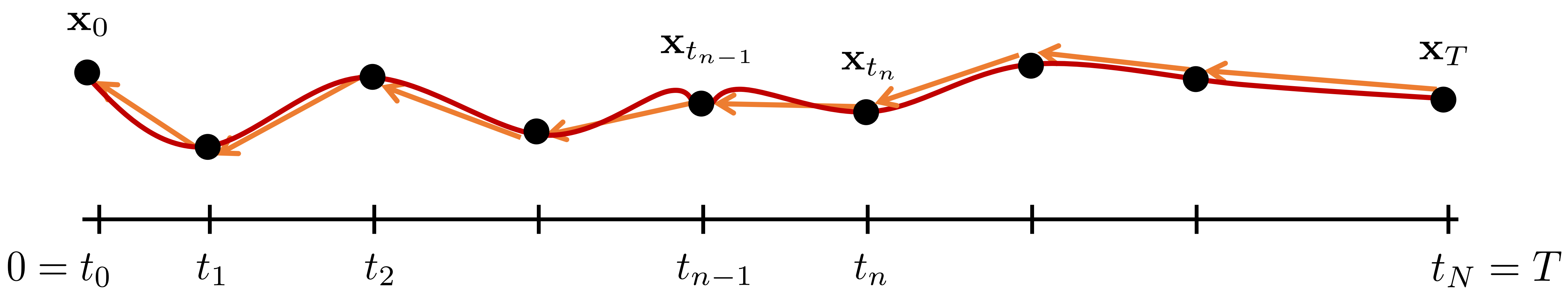
How to Solve the Reverse ODE?

Euler's Method (1st order):

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = -t\mathbf{s}_\theta(\mathbf{x}_t, t)$$

$$\approx$$

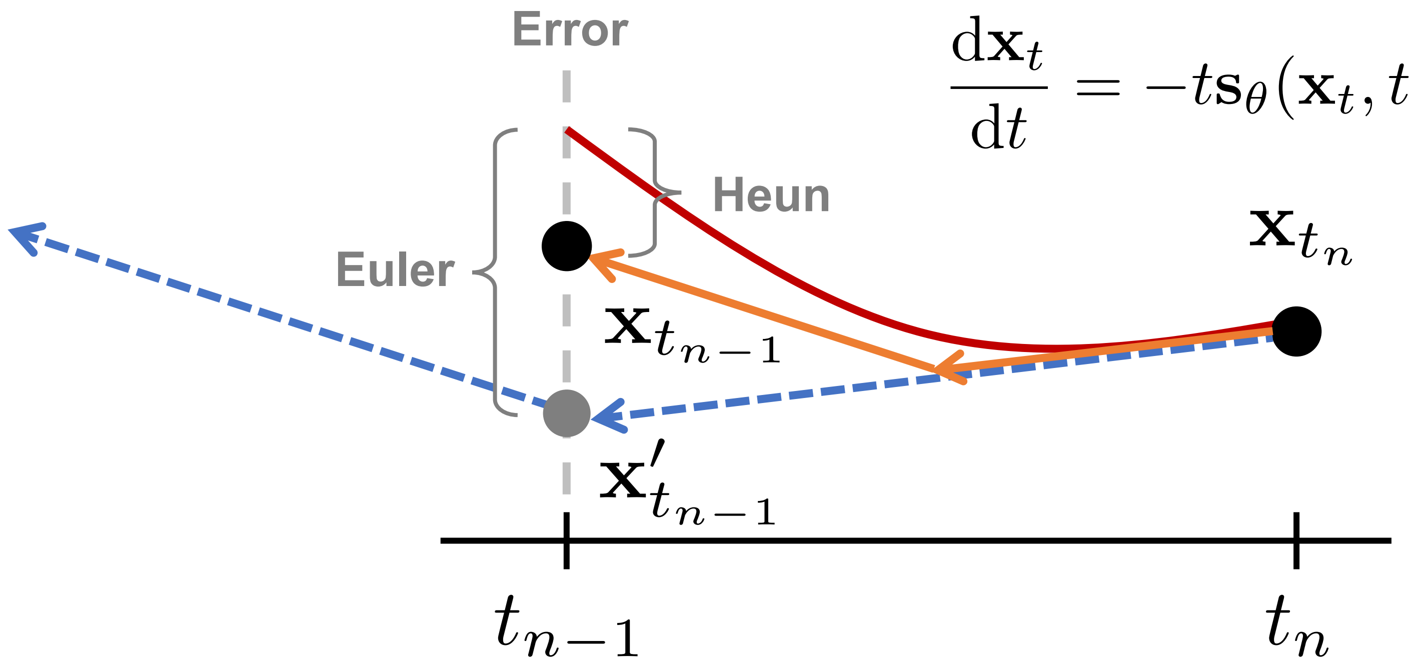
$$\frac{\mathbf{x}_{t_{n-1}} - \mathbf{x}_{t_n}}{t_{n-1} - t_n} + O(n)$$



$$\mathbf{x}_{t_{n-1}} = \mathbf{x}_{t_n} - (t_{n-1} - t_n)t_n\mathbf{s}_\theta(\mathbf{x}_{t_n}, t_n)$$

How to Solve the Reverse ODE?

Heun's Method (2nd order):

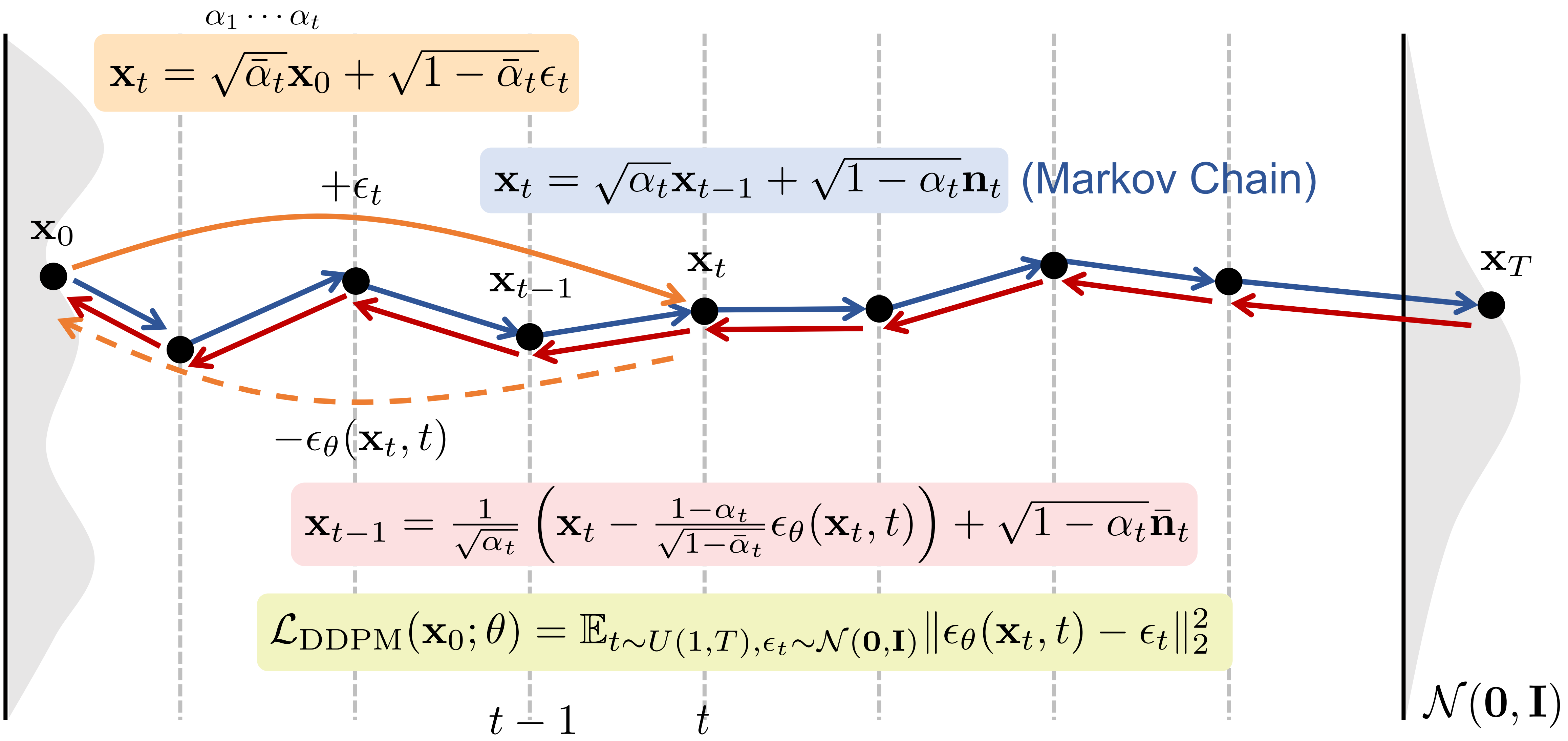


$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = -t\mathbf{s}_\theta(\mathbf{x}_t, t)$$

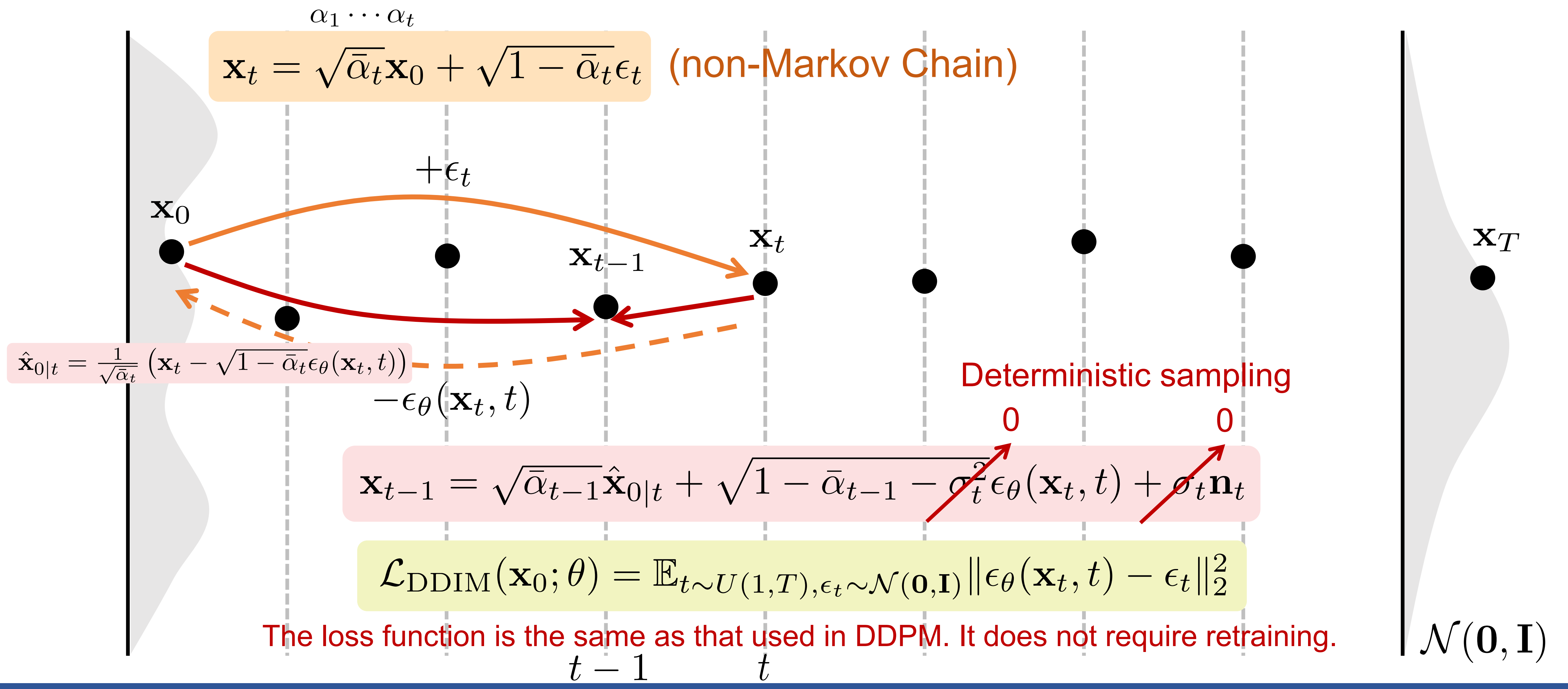
$$\mathbf{x}'_{t_{n-1}} = \mathbf{x}_{t_n} - (t_{n-1} - t_n)t_n\mathbf{s}_\theta(\mathbf{x}_{t_n}, t_n)$$

$$\mathbf{x}_{t_{n-1}} = \mathbf{x}_{t_n} - \frac{1}{2}(t_{n-1} - t_n)\left[t_n\mathbf{s}_\theta(\mathbf{x}_{t_n}, t_n) + t_{n-1}\mathbf{s}_\theta(\mathbf{x}'_{t_{n-1}}, t_{n-1})\right]$$

Denoising Diffusion Probabilistic Models (DDPM) [Ho+ NeurIPS’20]

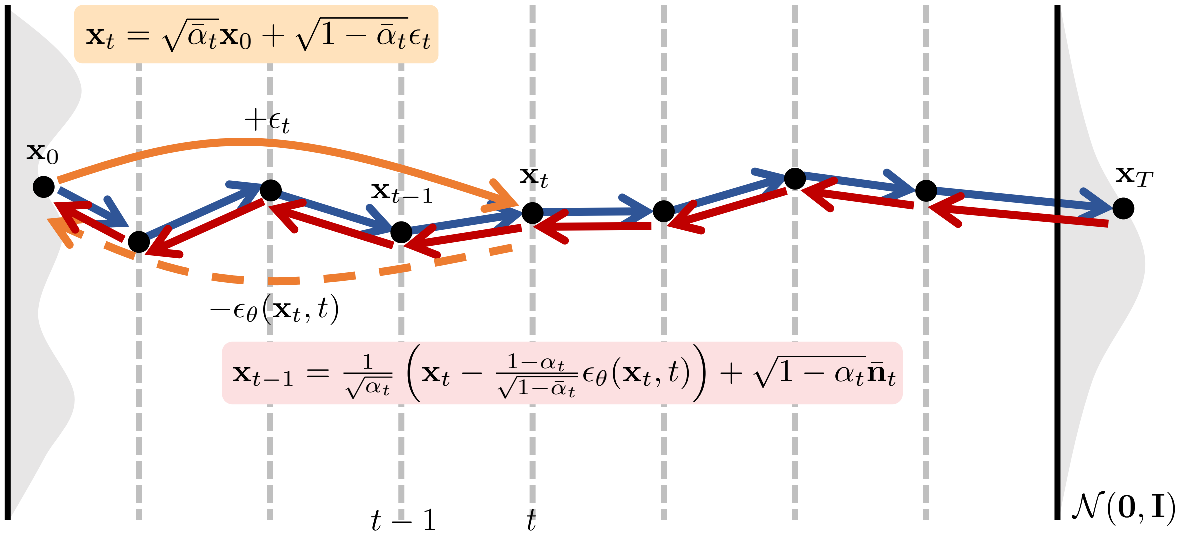


Denoising Diffusion Implicit Models (DDIM) [Song+ ICLR'21]



Summary: Discreate vs Continuous

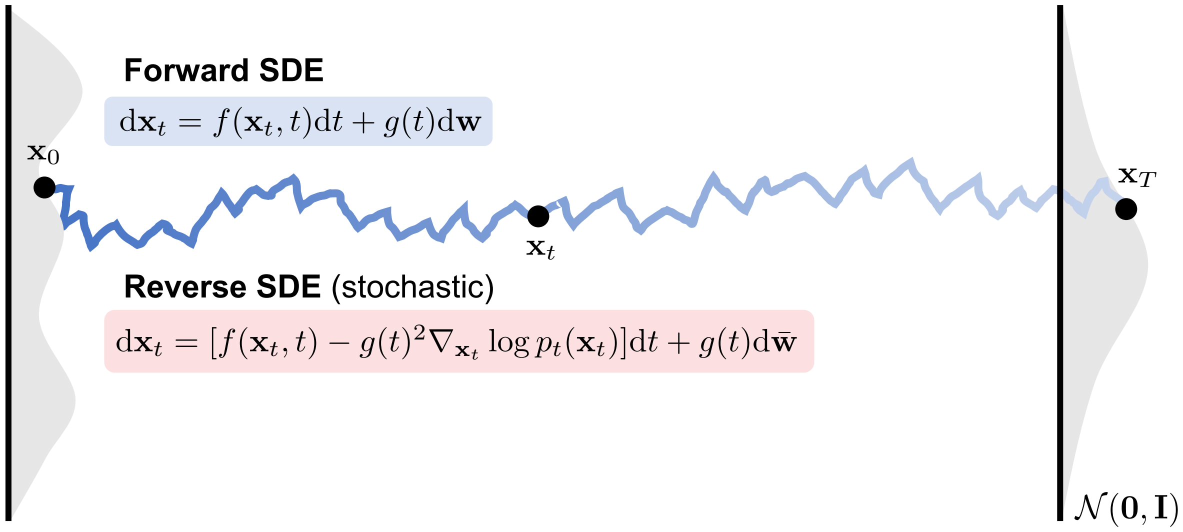
DDPM



$$\mathcal{L}_{\mathrm{DDPM}}(\mathbf{x}_0;\theta) = \mathbb{E}_{t\sim U(1,T),\epsilon_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon_t\|_2^2$$

Discretized in the training phase

Score-based SDE



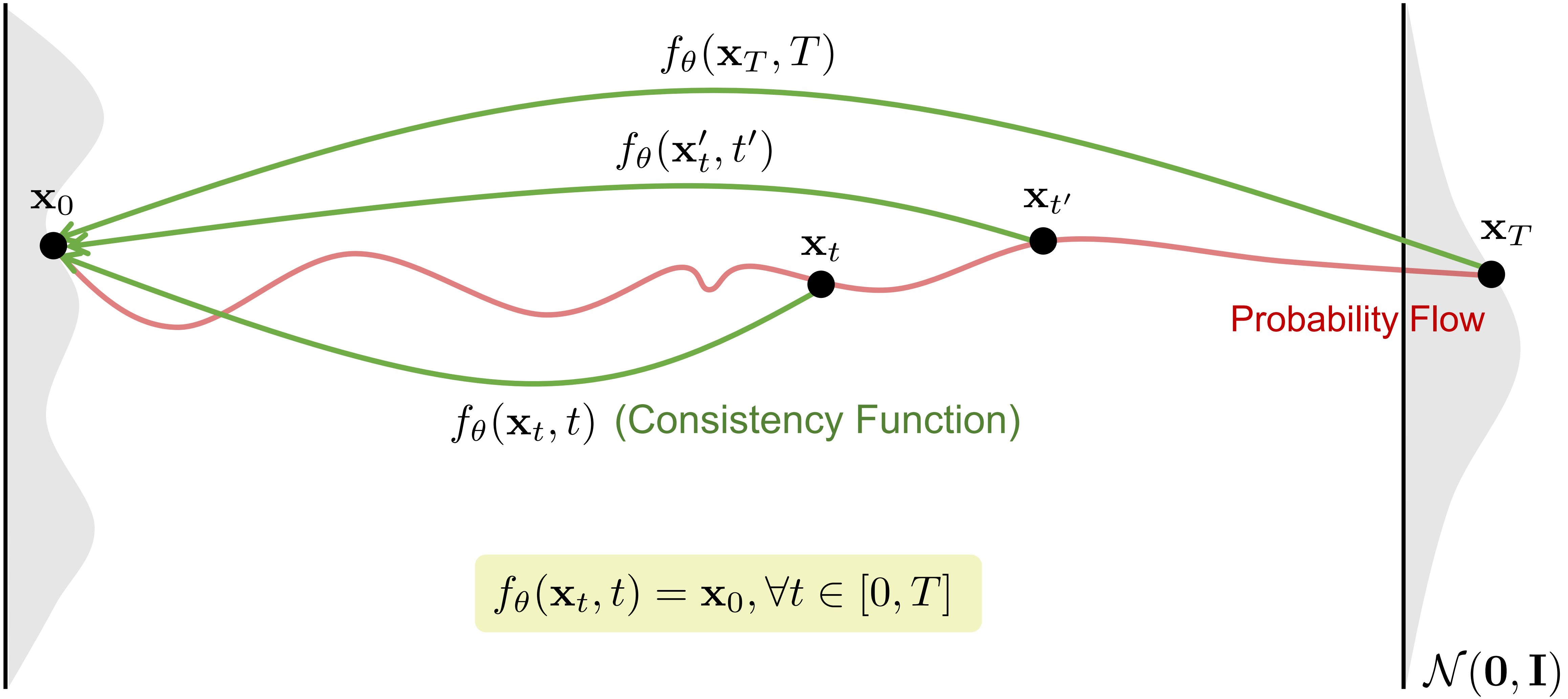
$$\mathcal{L}_{\mathrm{DSM}}(\mathbf{x}_0;\theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0}\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2$$

Discretized during sampling

The error is only affected by the order of the solver

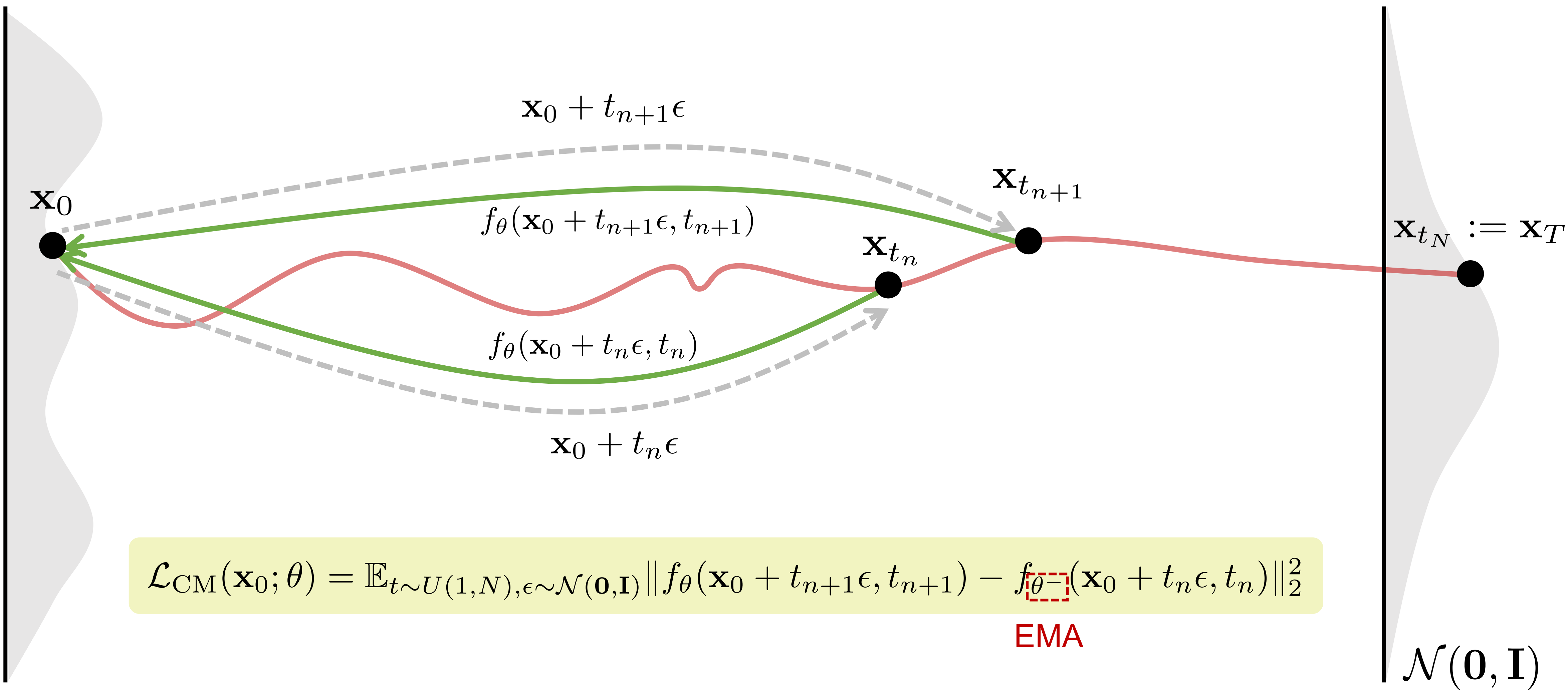
Consistency Models (CM) [Song+ ICML'23]

$$\mathrm{d}\mathbf{x}_t = -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\mathrm{d}t$$



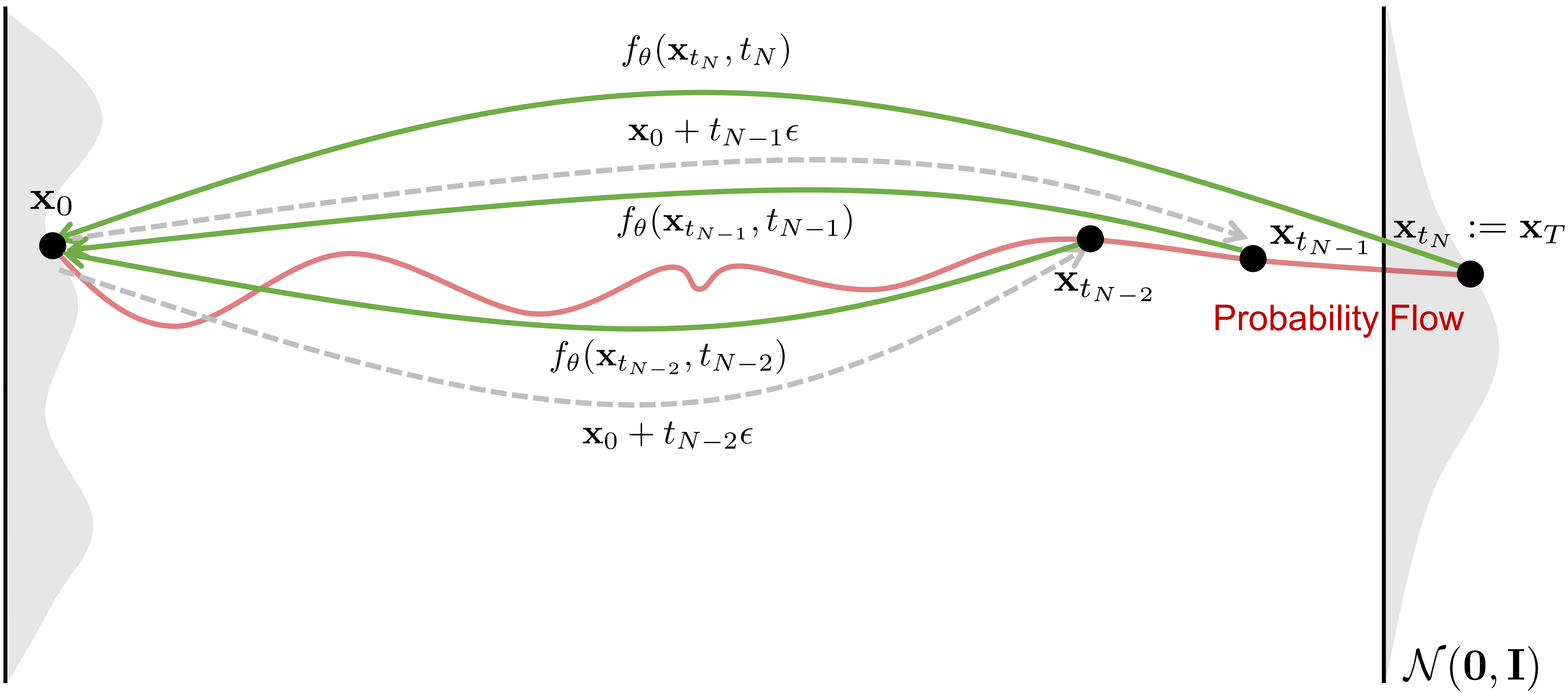
$$f_\theta(\mathbf{x}_t, t) = \mathbf{x}_0, \forall t \in [0, T]$$

Training CM

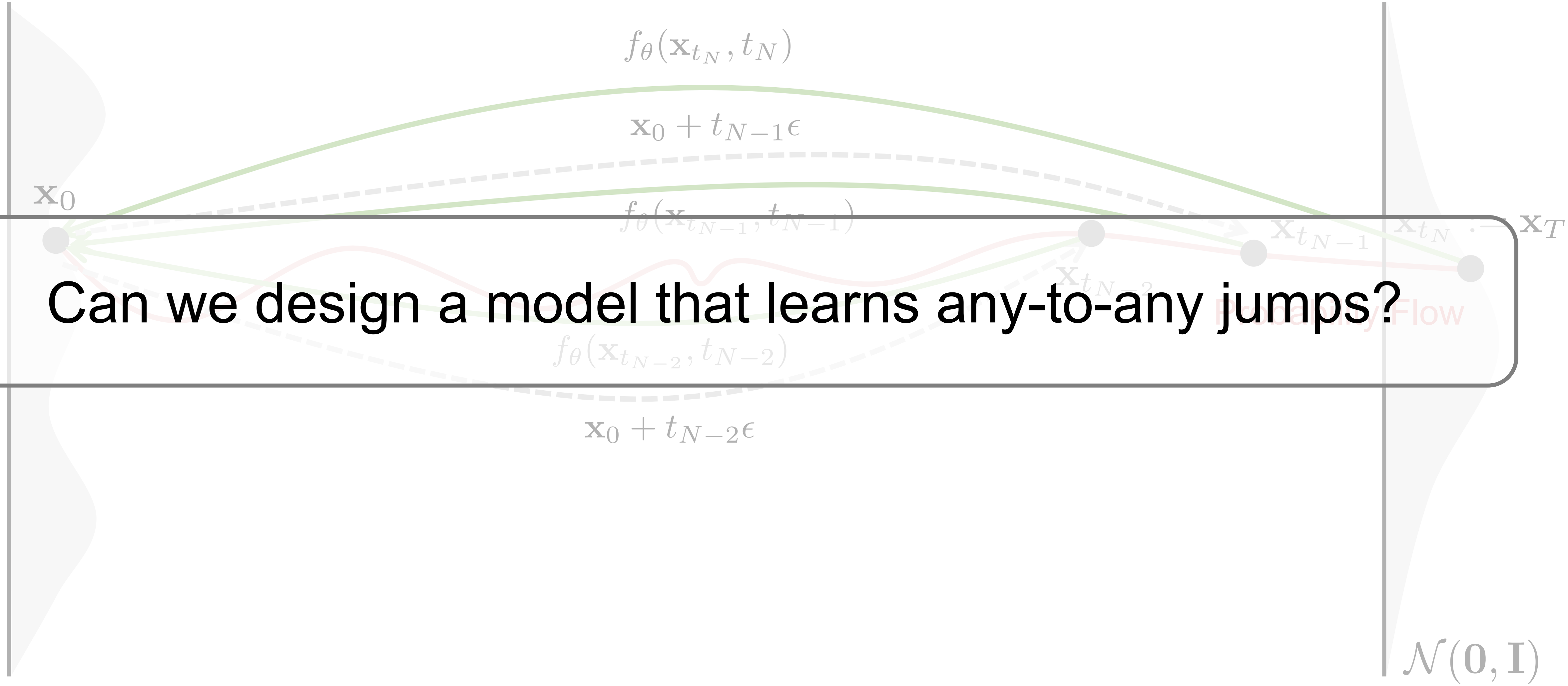


$$\mathcal{L}_{\text{CM}}(\mathbf{x}_0; \theta) = \mathbb{E}_{t \sim U(1,N), \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})} \| f_\theta(\mathbf{x}_0 + t_{n+1}\epsilon, t_{n+1}) - f_{\boxed{\theta^-}}(\mathbf{x}_0 + t_n\epsilon, t_n) \|_2^2$$

Sampling with CM



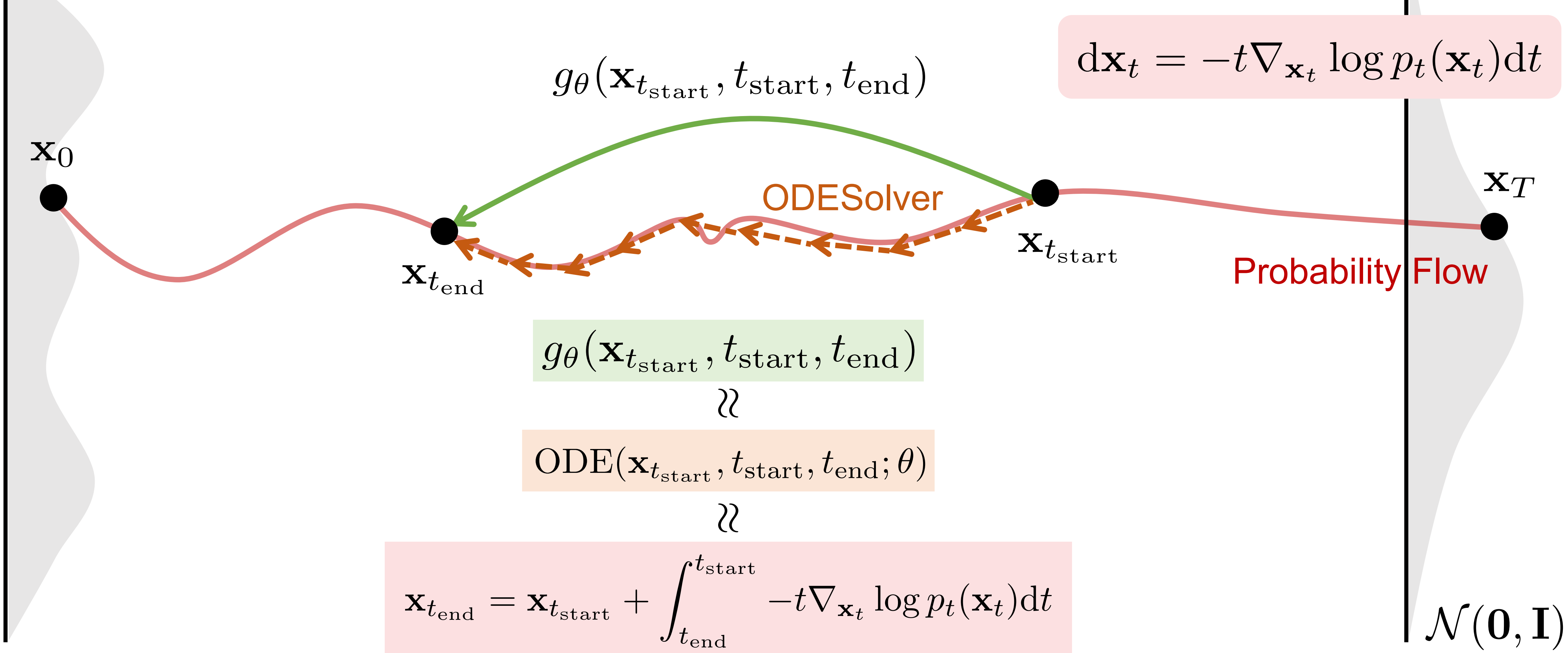
Sampling with CM



Can we design a model that learns any-to-any jumps?

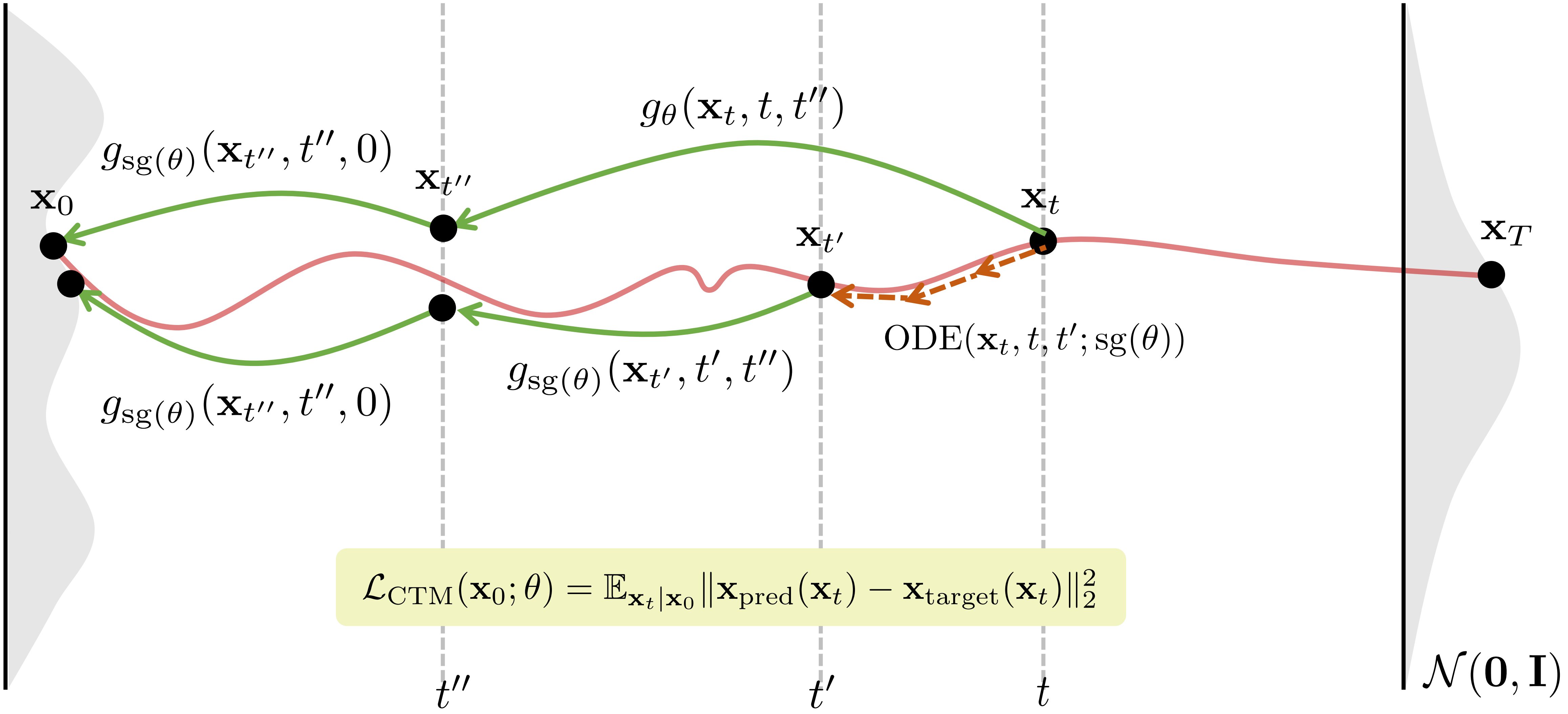
Consistency Trajectory Models (CTM) [Kim+ ICLR’24]

CTM learns any-to-any jump on the trajectory – *Jesse*



$$g_\theta(\mathbf{x}_{t_\text{start}}, t_\text{start}, t_\text{end})$$
$$\wr\wr$$
$$\text{ODE}(\mathbf{x}_{t_\text{start}}, t_\text{start}, t_\text{end}; \theta)$$
$$\wr\wr$$
$$\mathbf{x}_{t_\text{end}} = \mathbf{x}_{t_\text{start}} + \int_{t_\text{end}}^{t_\text{start}} -t\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)dt$$

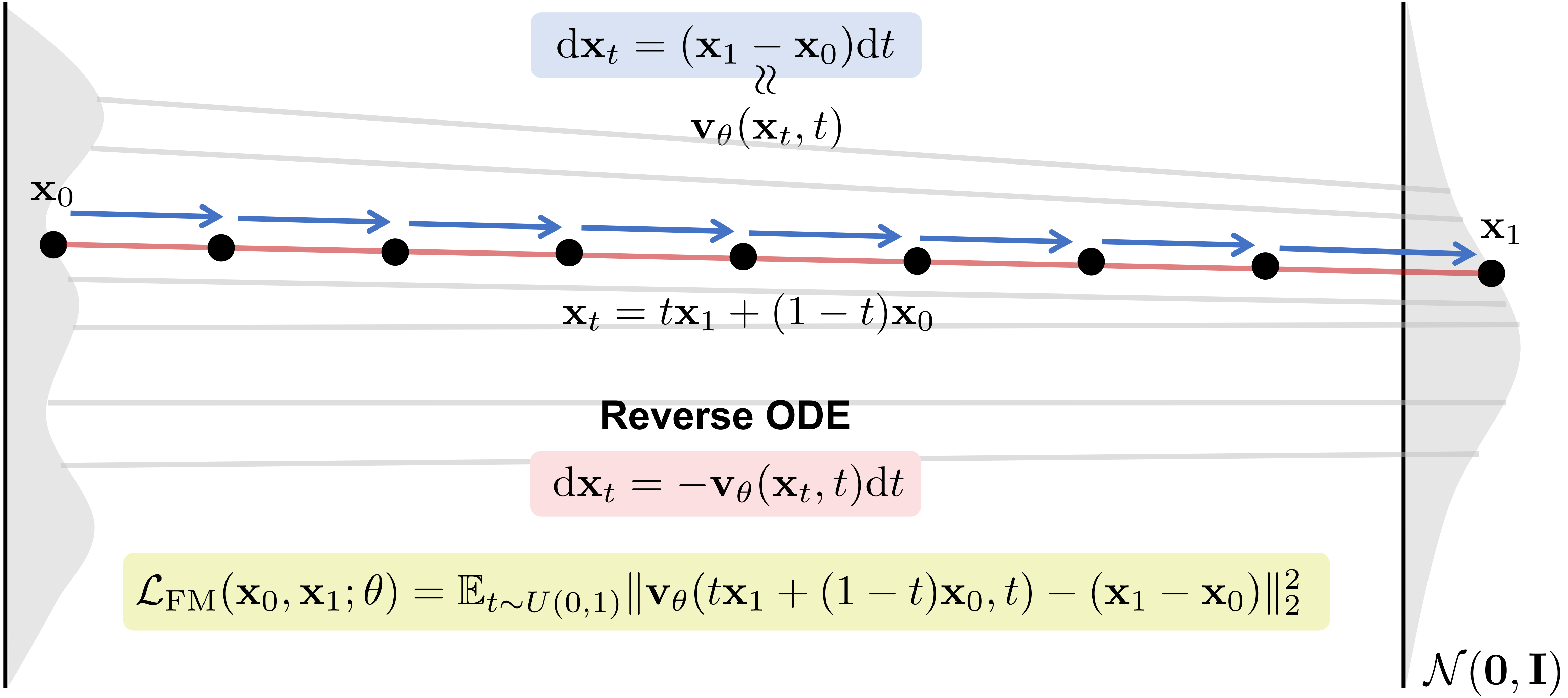
Training CTM



Flow Matching [Lipman+ ICLR’23][Liu+ ICLR’23]

Forward ODE

$$\mathrm{d}\mathbf{x}_t = (\mathbf{x}_1 - \mathbf{x}_0)\mathrm{d}t \approx \mathbf{v}_\theta(\mathbf{x}_t, t)$$

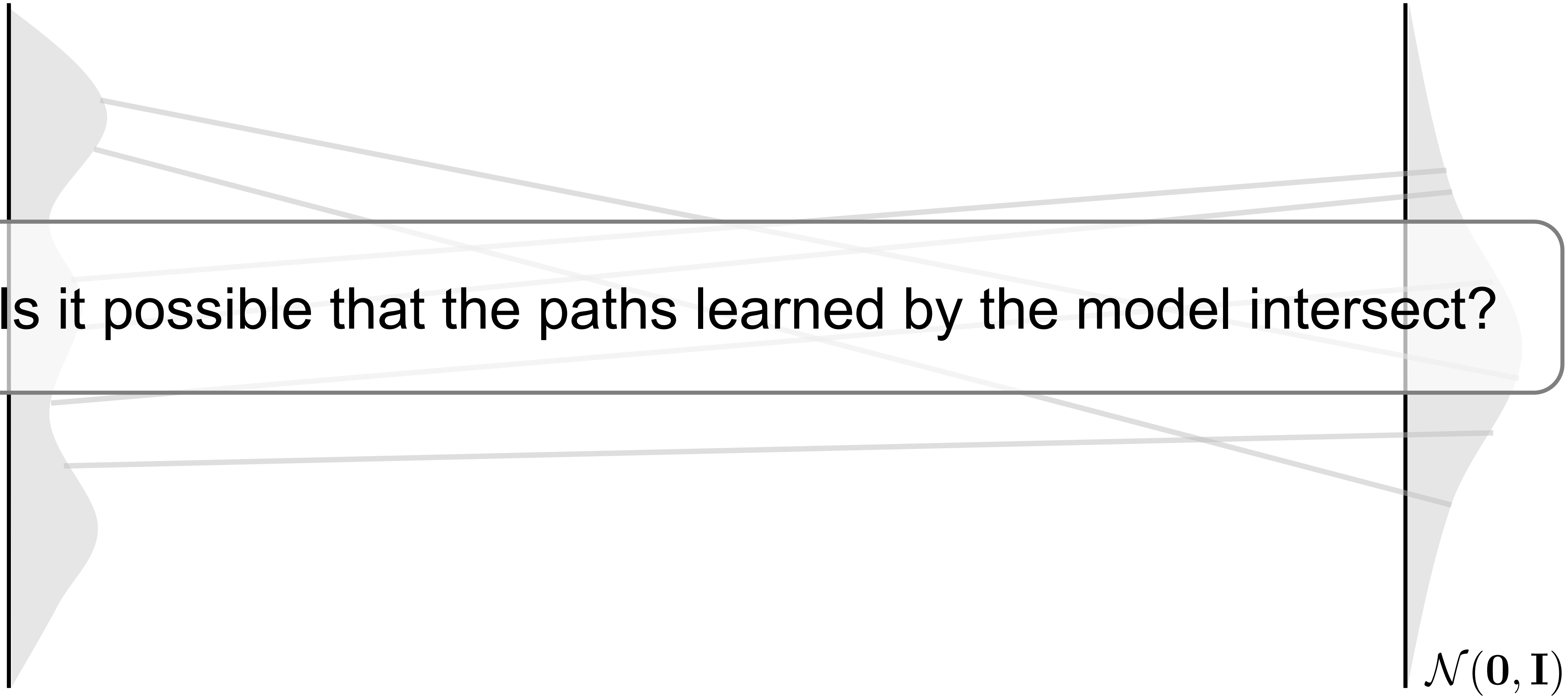


Reverse ODE

$$\mathrm{d}\mathbf{x}_t = -\mathbf{v}_\theta(\mathbf{x}_t, t)\mathrm{d}t$$

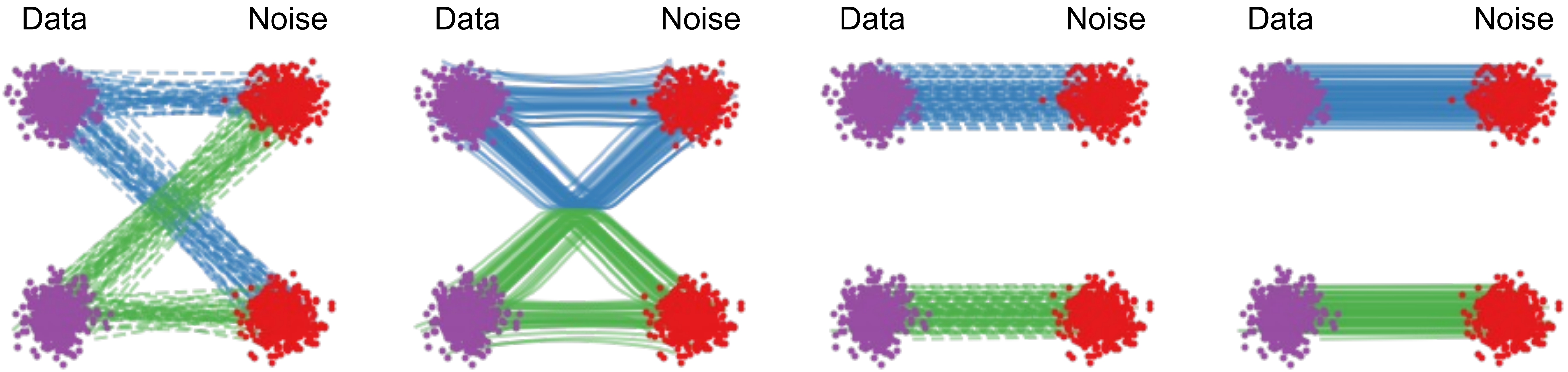
$$\mathcal{L}_{\mathrm{FM}}(\mathbf{x}_0, \mathbf{x}_1; \theta) = \mathbb{E}_{t\sim U(0,1)} \|\mathbf{v}_\theta(t\mathbf{x}_1 + (1-t)\mathbf{x}_0, t) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2$$

Flow Matching [Lipman+ ICLR'23][Liu+ ICLR'23]



Is it possible that the paths learned by the model intersect?

k-Rectified Flow [Liu+ ICLR’23]



$$\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$$

Flow Matching
(1-Rectified Flow)
(not straight but non-crossing)

$$\mathbf{x}_t^{(1)} = t\mathbf{x}_1^{(1)} + (1-t)\mathbf{x}_0^{(1)}$$

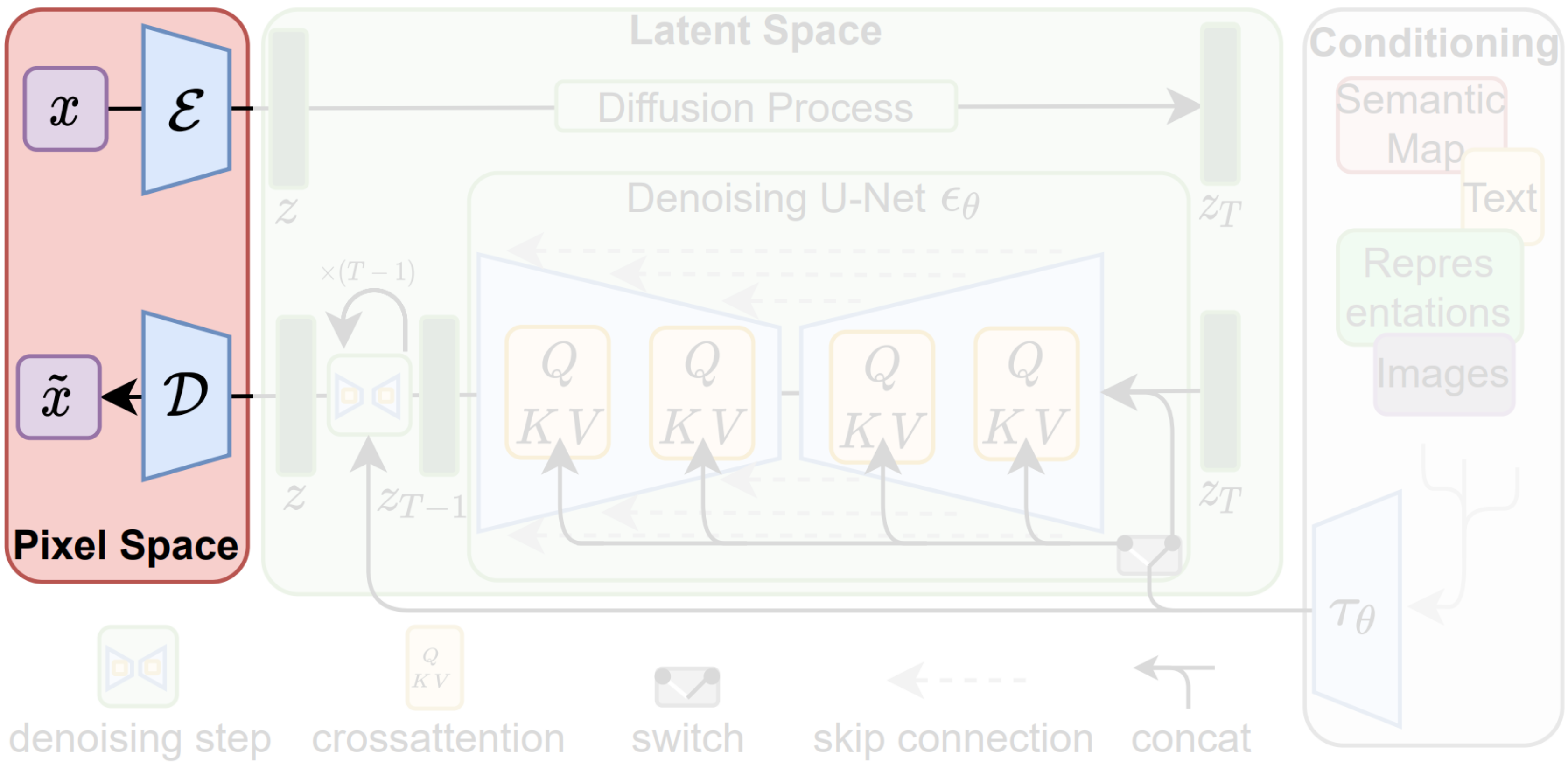
(Sample the data-noise pair
from 1-Rectified Flow)

2-Rectified Flow
(straight)

How to reduce the computational cost of diffusion models?

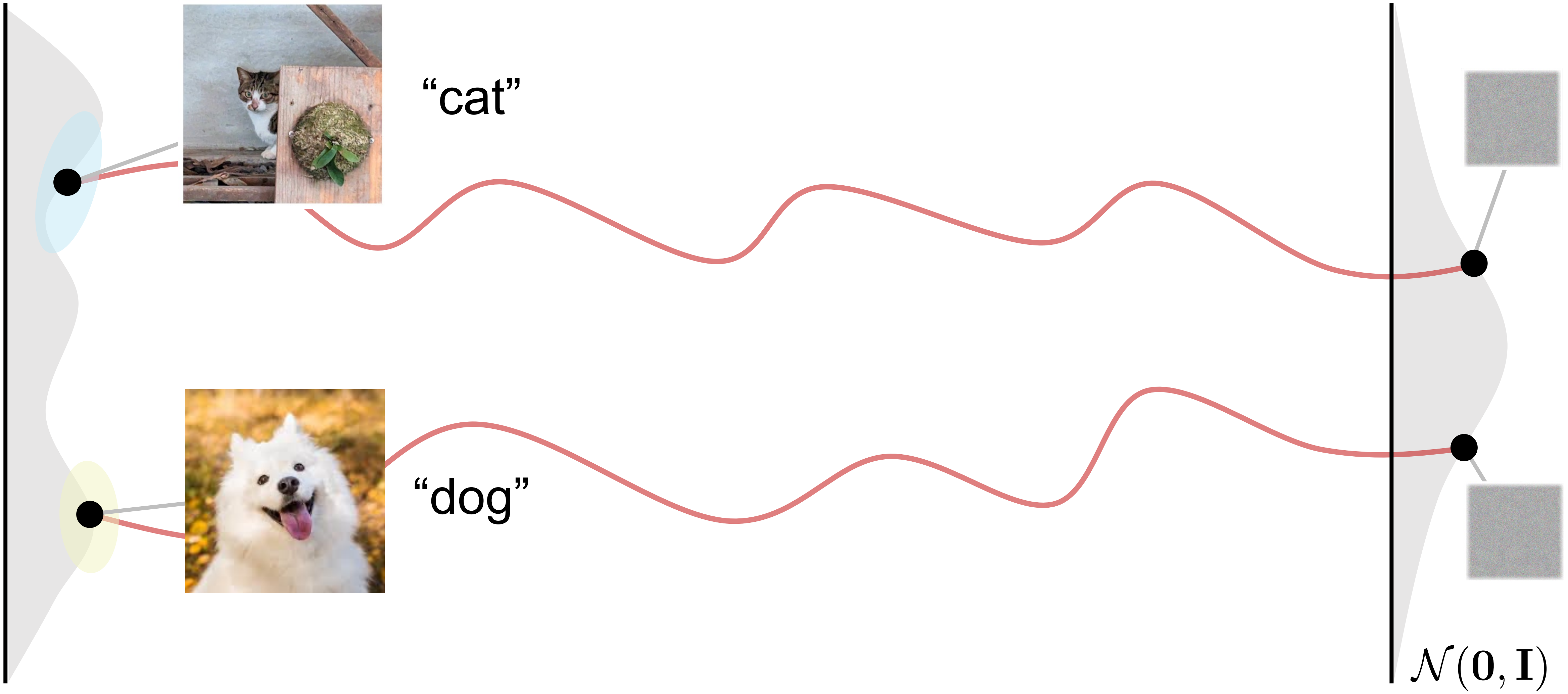
Latent Diffusion Model (LDM) [Rombach+ CVPR'22]

Use the pretrained-VAE to compress the image to latent reducing computation time

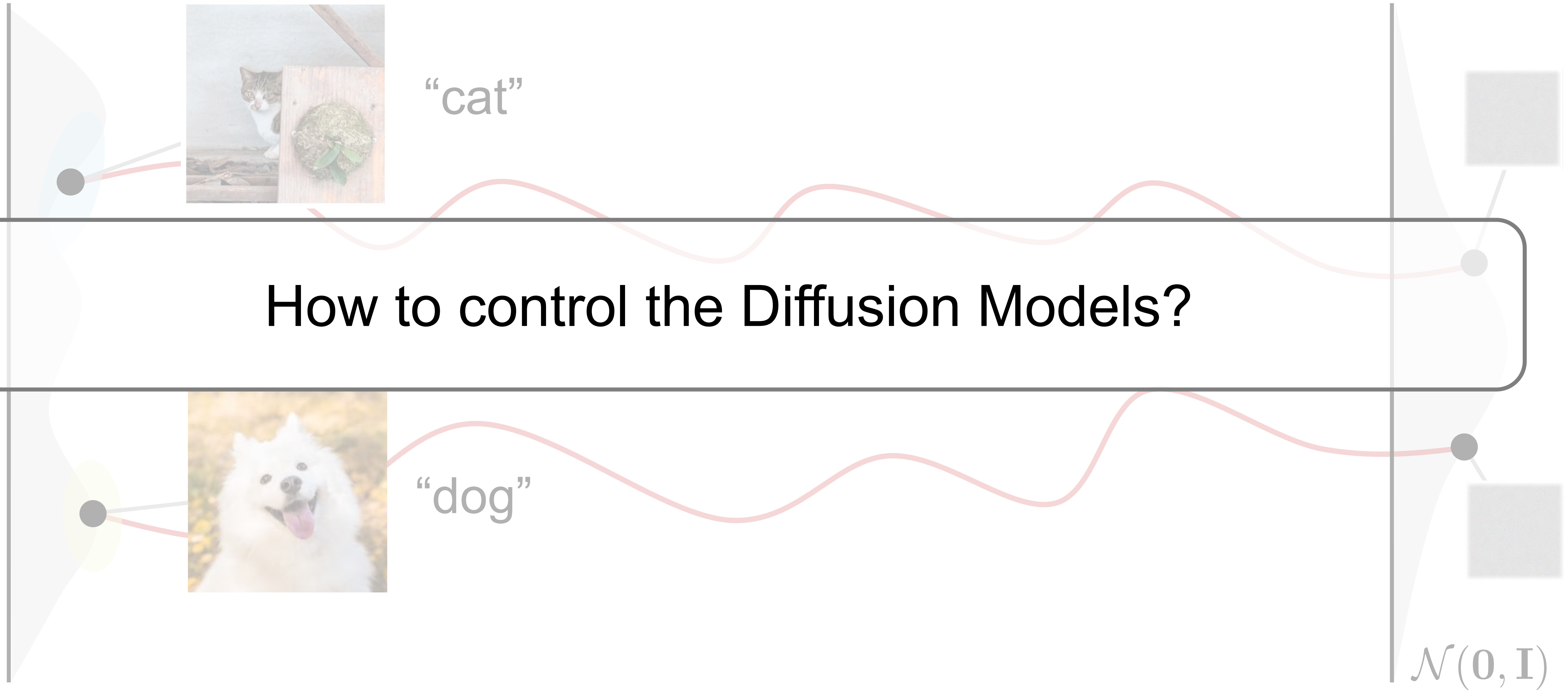


Can we control the Diffusion Models?

Conditional Diffusion Models

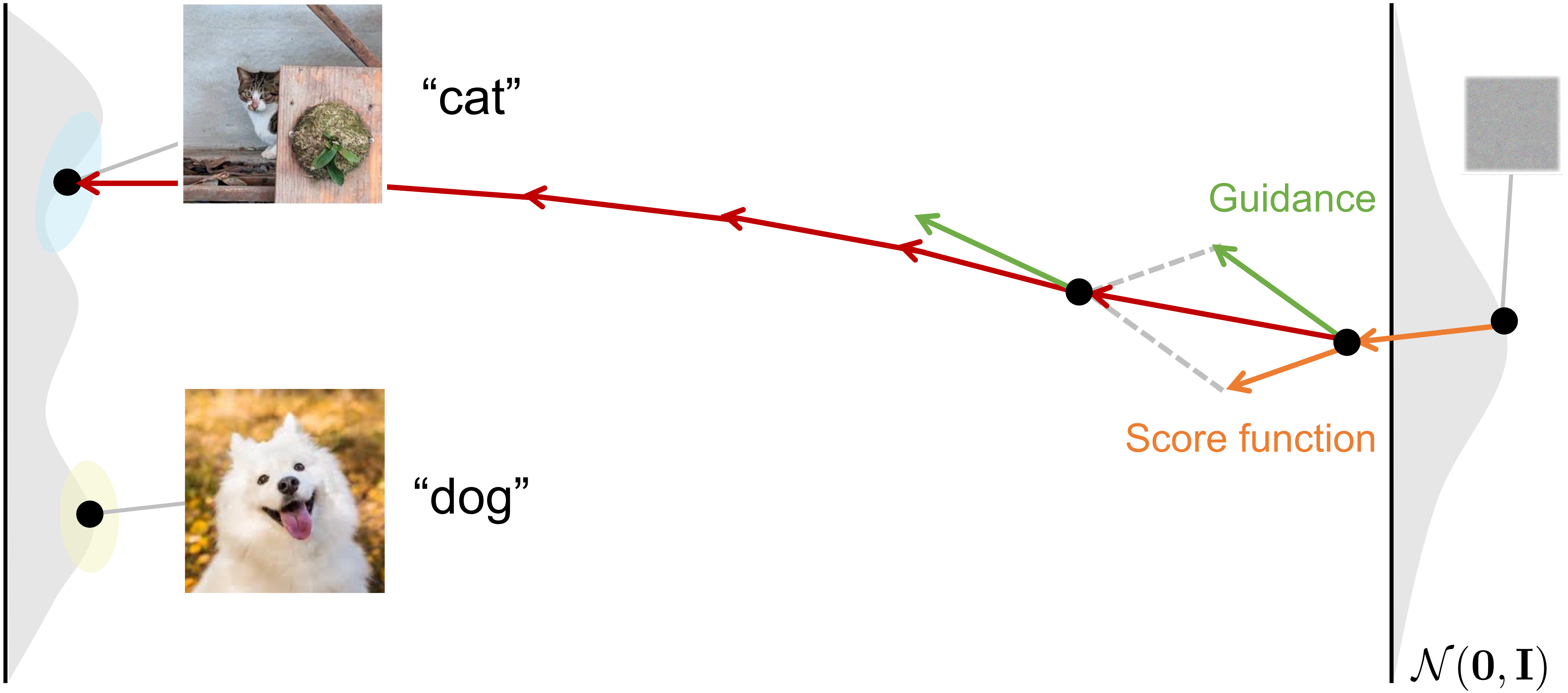


Conditional Diffusion Models



How to control the Diffusion Models?

Conditional Diffusion Models



Classifier Guidance (CG) [Dhariwal+ NeurIPS'21]

- **Bayes' Rule:**

$$P\Bigg(\mathbf{x}_t =$$

$$\Bigg|\, \mathbf{y} = \text{“cat”}\Bigg) = \frac{p(\mathbf{x}_t)p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})}$$

- **Bayes' Rule for Score Function:**

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) - \cancelto{0}{\nabla_{\mathbf{x}_t} \log p(\mathbf{y})}$$

$$= \underbrace{\boxed{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}}_{\text{Unconditional score}} + \nabla_{\mathbf{x}_t} \log \underbrace{\boxed{p(\mathbf{y}|\mathbf{x}_t)}}_{\text{Classifier (need to additional training)}}$$

Classifier-Free Guidance (CFG) [Ho+ NeurIPSW'21]

From CG, we have

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Reweight the coefficient between unconditional score and classifier score

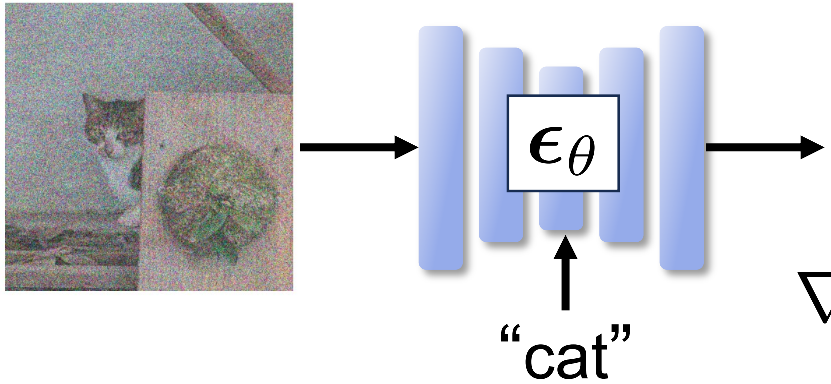
$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) := \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \boxed{\gamma} \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

$$= \gamma \underbrace{\boxed{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})}}_{\text{Conditional score}} + (1-\gamma) \underbrace{\boxed{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}}_{\text{Unconditional score}}$$

Training CFG

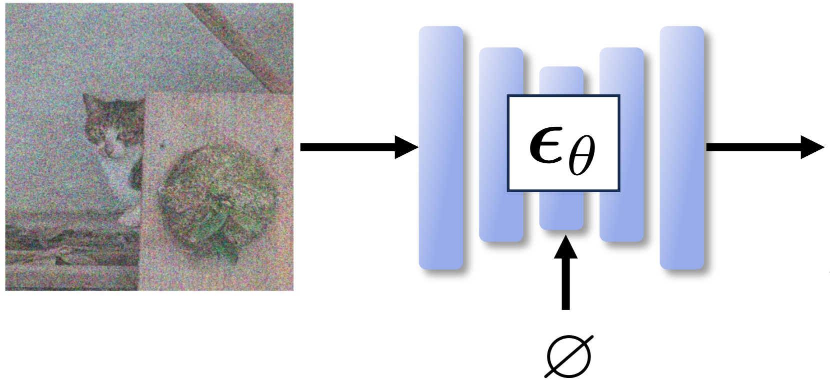
$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})}_{\text{Conditional score}} + (1-\gamma) \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}_{\text{Unconditional score}}$$

Predict Conditional Score



$$\mathbf{s}_\theta(\mathbf{x}_t, t, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$$

Predict Unconditional Score

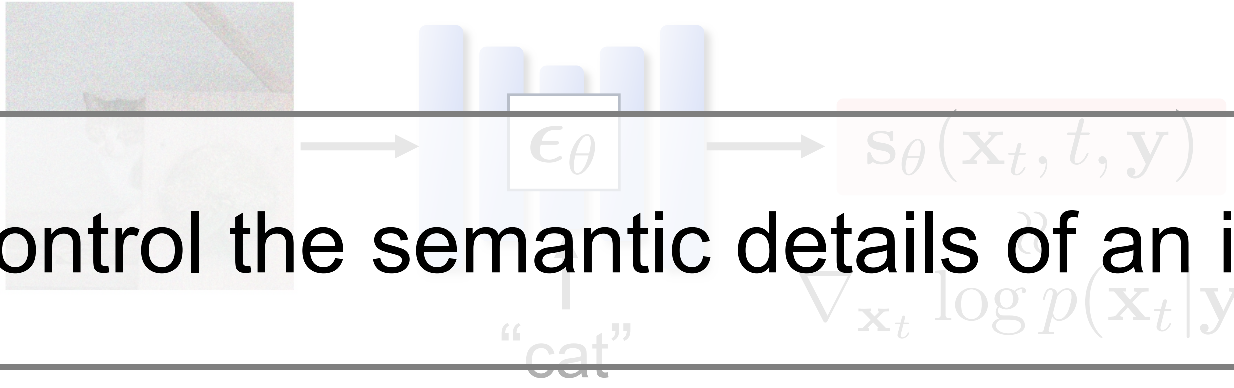


$$\mathbf{s}_\theta(\mathbf{x}_t, t, \varnothing) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Training CFG

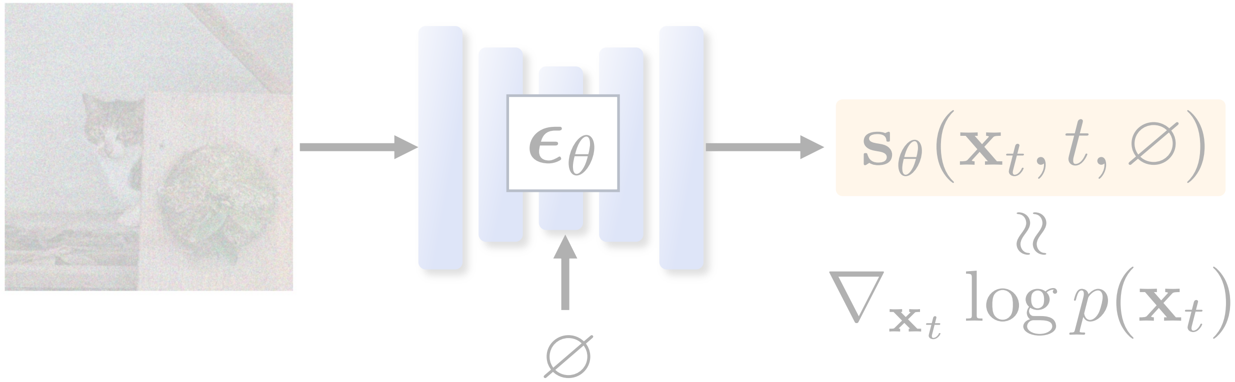
$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})}_{\text{Conditional score}} + (1-\gamma) \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}_{\text{Unconditional score}}$$

Predict Conditional Score

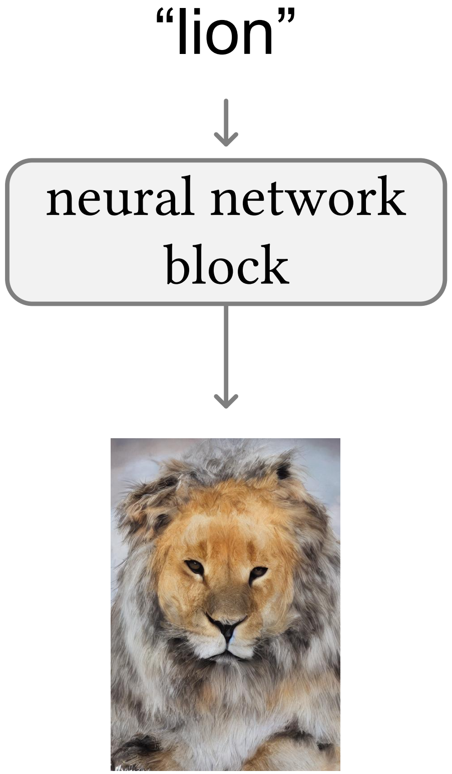


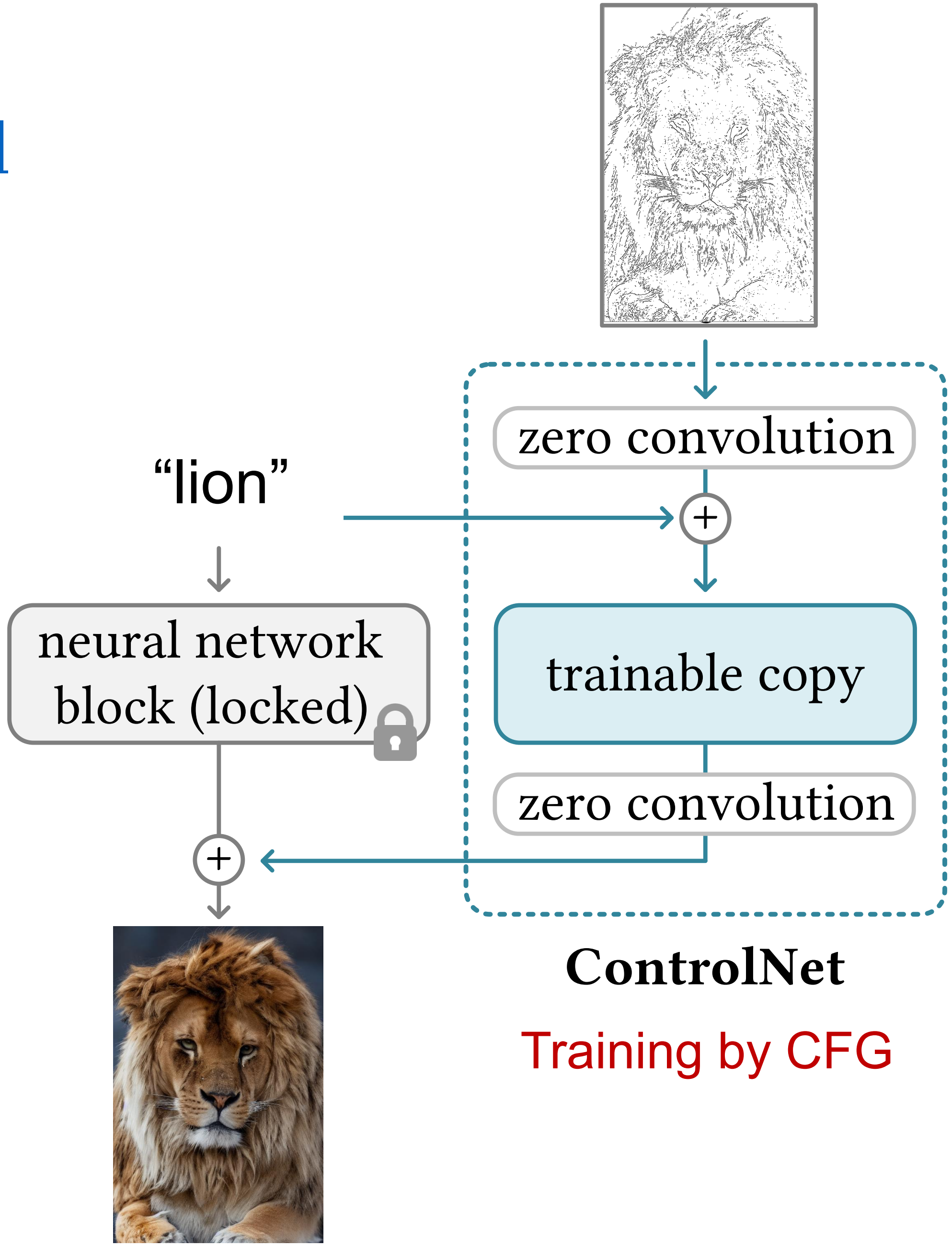
Can we control the semantic details of an image?

Predict Unconditional Score



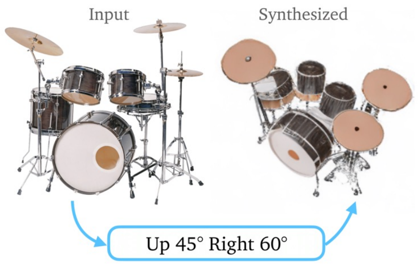
ControlNet [Zhang+ ICCV’23]





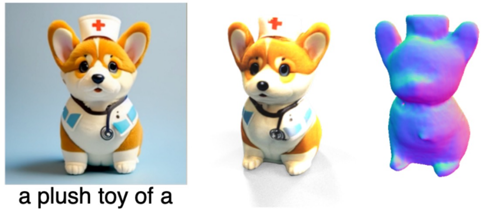
Training by CFG

Applications of Diffusion Models in 3D Vision



Zero-1-to-3
2023 / 5

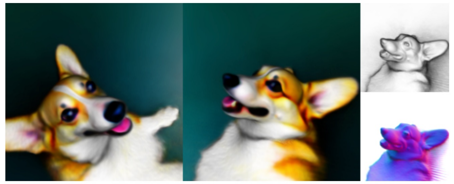
Magic123
2023 / 6



a plush toy of a corgi nurse

One-2-3-45++
2023 / 11

DreamFusion
2022 / 9

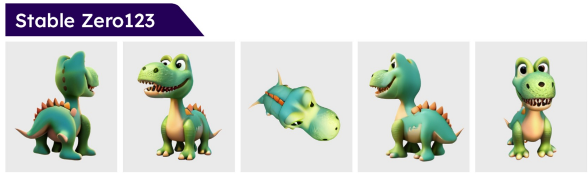


a corgi taking a selfie*

One-2-3-45
2023 / 6

Zero123++
2023 / 10

Stable Zero123
2023 / 12



Takeaway

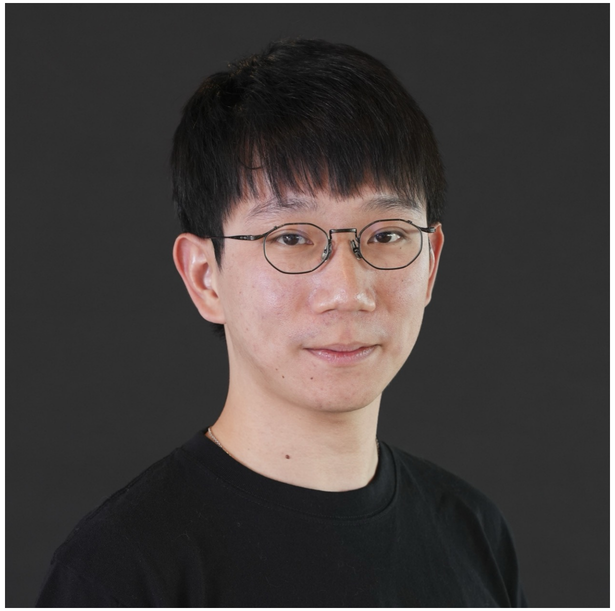
- Diffusion models build a bridge between noise and data, forming a powerful generative modeling framework.
- Score-based models leverage SDE/ODE formulations and score functions to guide the reverse process.
- Variants like DDIM, CM, CTM, and Flow Matching offer trade-offs in speed, quality, and control.
- Conditional methods like CFG and ControlNet significantly improve controllability and application scope.
- Diffusion models are already widely applied in 2D/3D image synthesis, speech, and multi-modal generation.

Acknowledgement

Some concepts and insights in these slides are inspired by Yang Song and Jesse

[Yang Song]

Research Scientist at OpenAI



[Chieh-Hsin (Jesse) Lai]

Research Scientist at Sony AI

Thank you