Recent Trends and Variants in Diffusion Models

Jia-Wei Liao

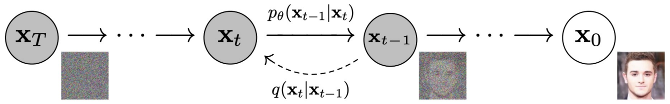
Ph.D. Candidate in Computer Science

National Taiwan University

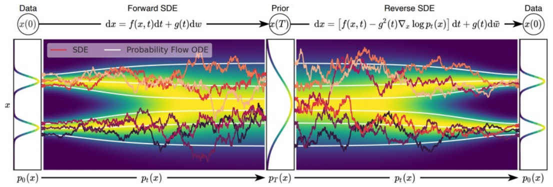
Outline

- Theoretical Foundations: From SDEs to Score-Based Modeling
- Training Objectives and Sampling Mechanisms
- Advanced Extensions: DDIM, Consistency Models, and CTMs

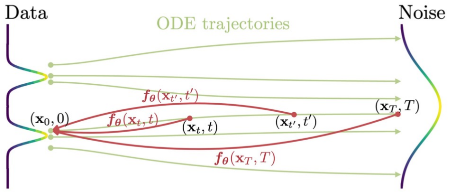
Evolution of Diffusion Models



DDPM [NeurIPS’20]
2020 / 6 (UCB)



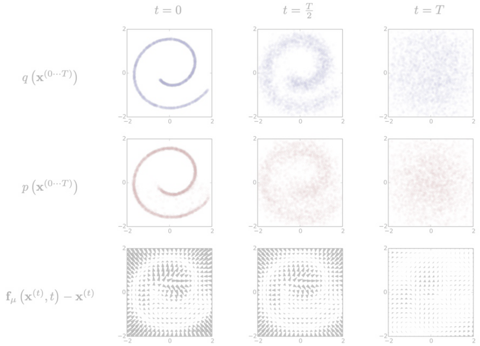
Score-based SDE [ICLR’21]
2020 / 11 (Ermon)



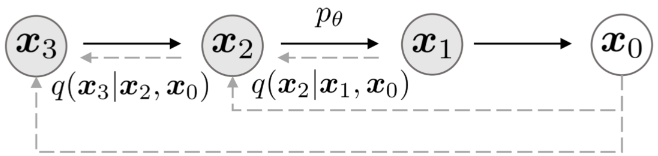
CM [ICML’23]
2023 / 3 (OpenAI)

Flow Matching!

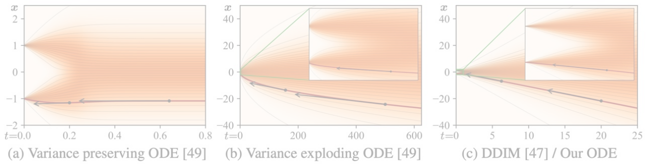
DPM [ICML’15]
2015 / 3 (Stanford)



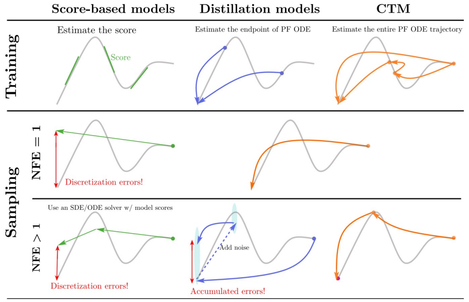
DDIM [ICLR’21]
2020 / 10 (Ermon)



EDM [NeurIPS’22]
2022 / 6 (NVIDIA)

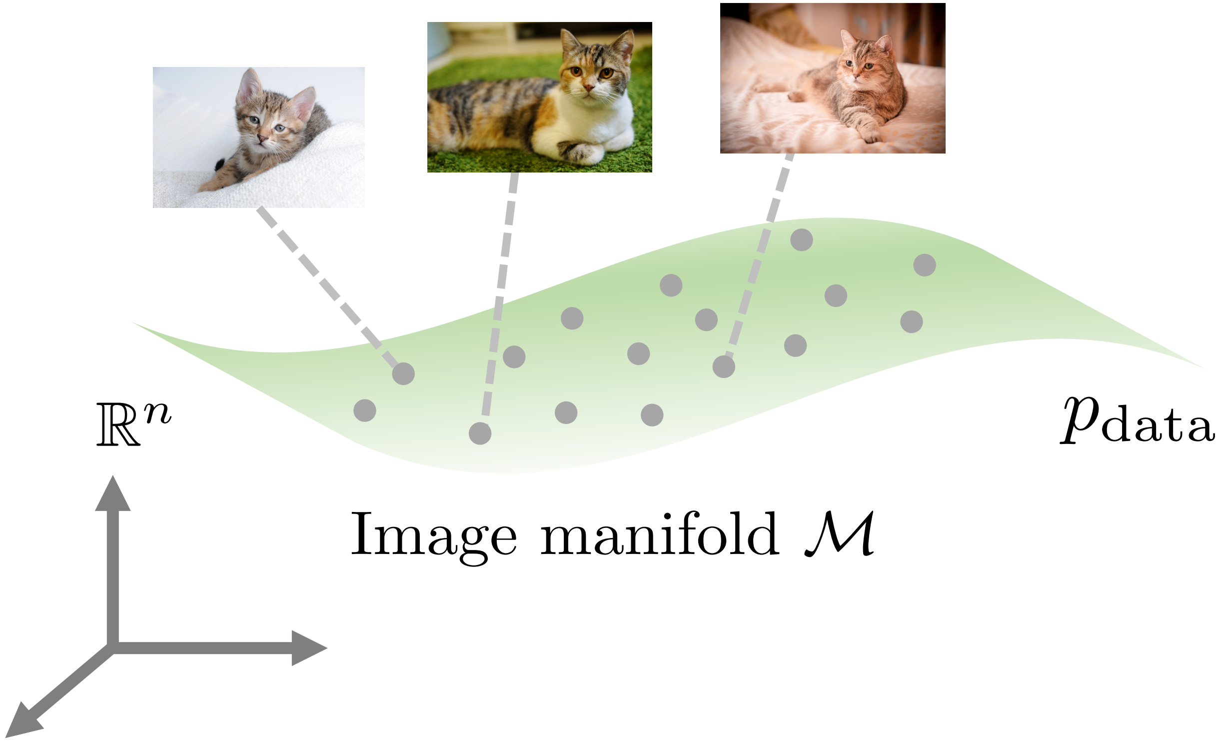


CTM [ICLR’24]
2023 / 10 (Sony AI)



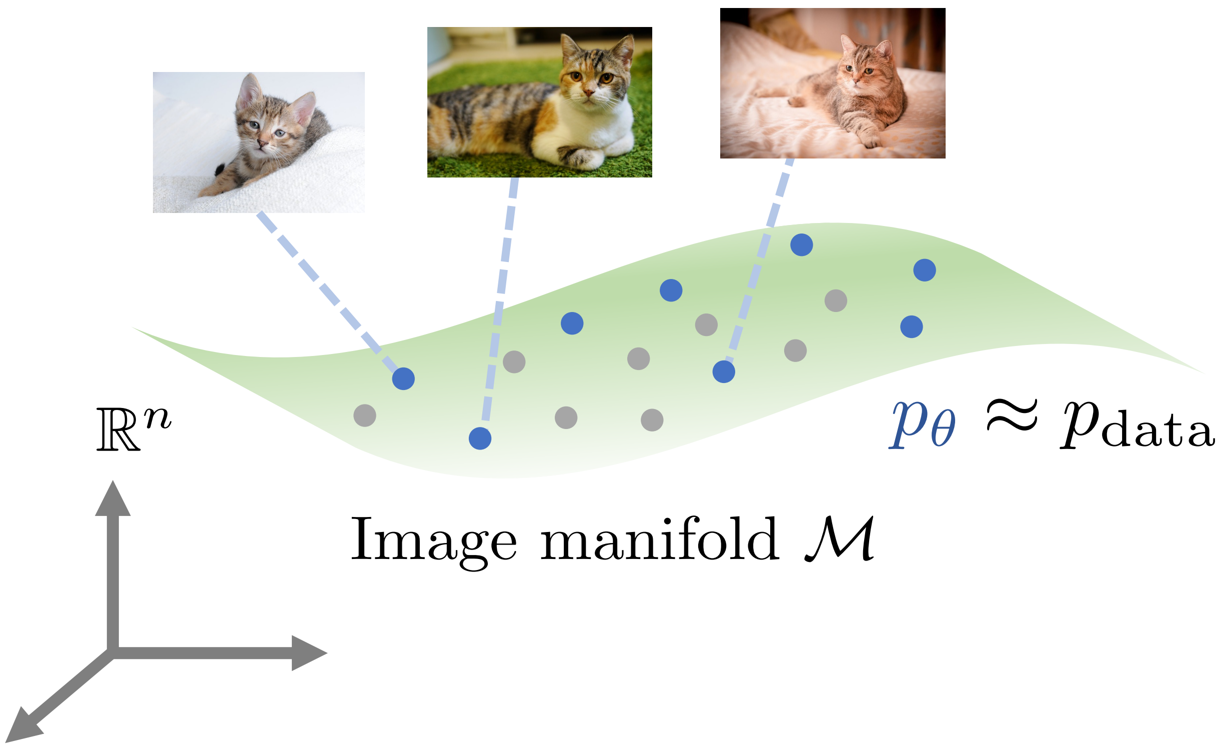
What is Generative Model Learning?

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



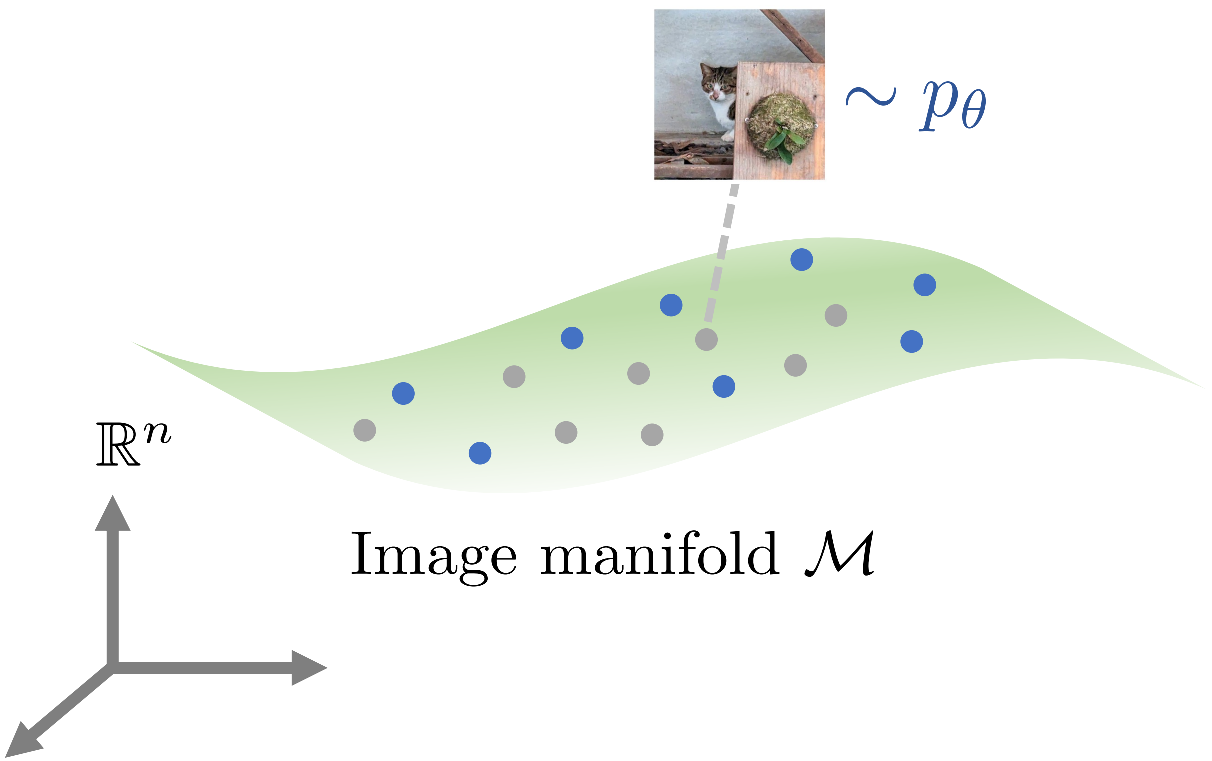
What is Generative Model Learning?

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



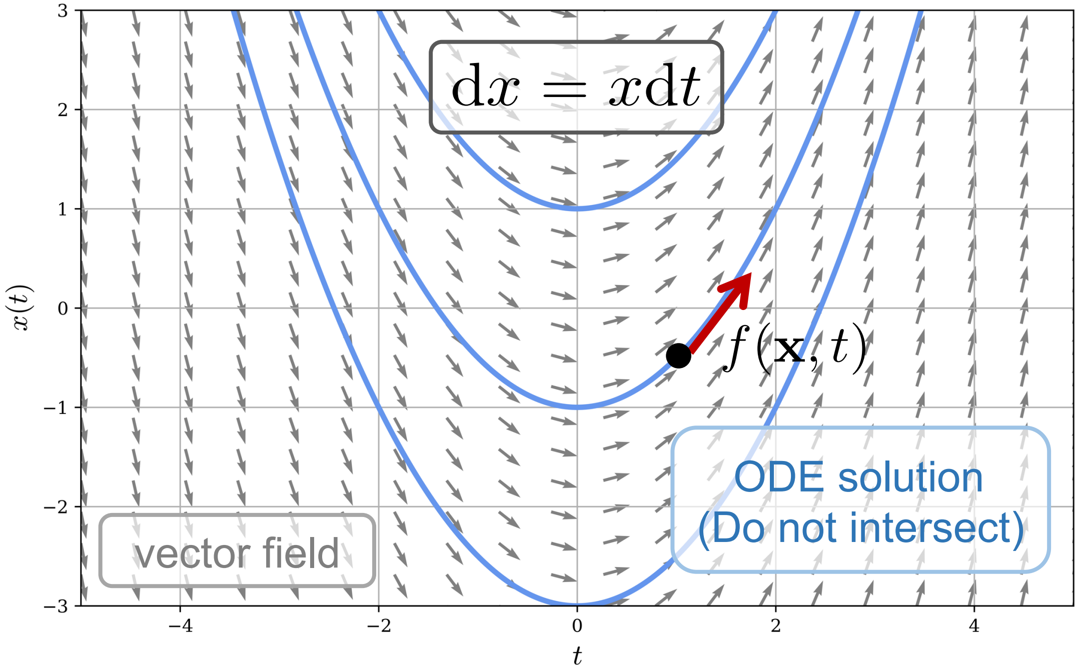
What is Generative Model Learning?

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

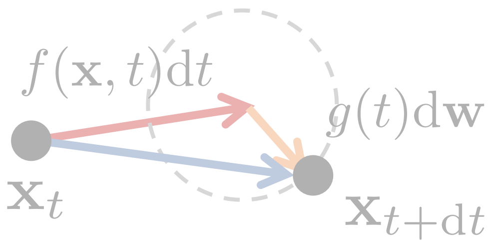


Recap ODE and SDE

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

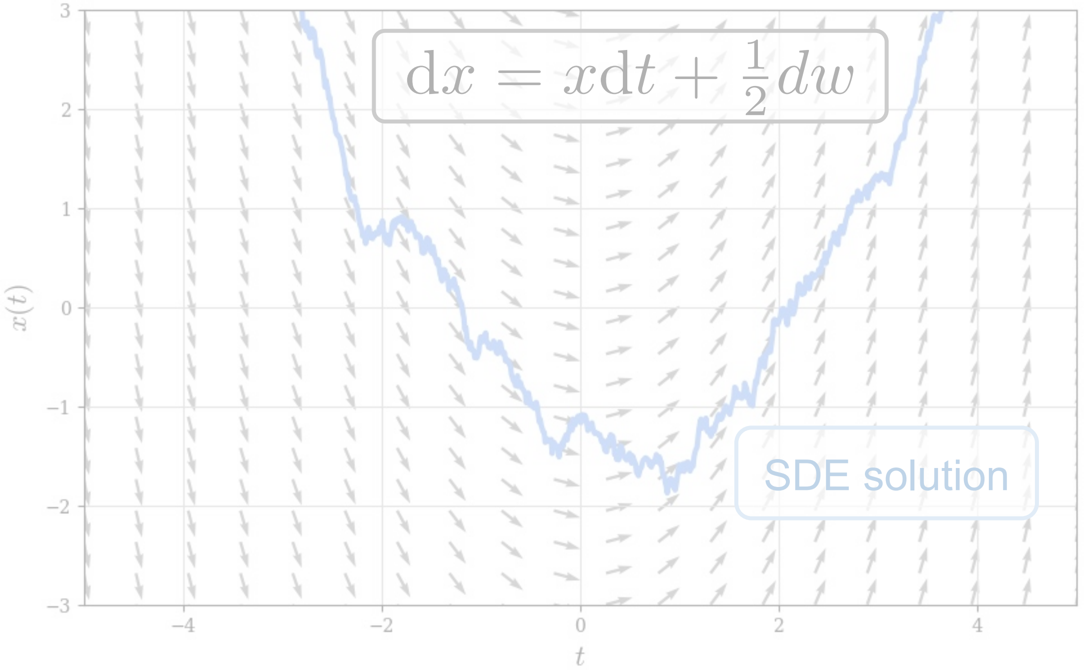


The arrow indicates the direction of movement at the moment of downward shift

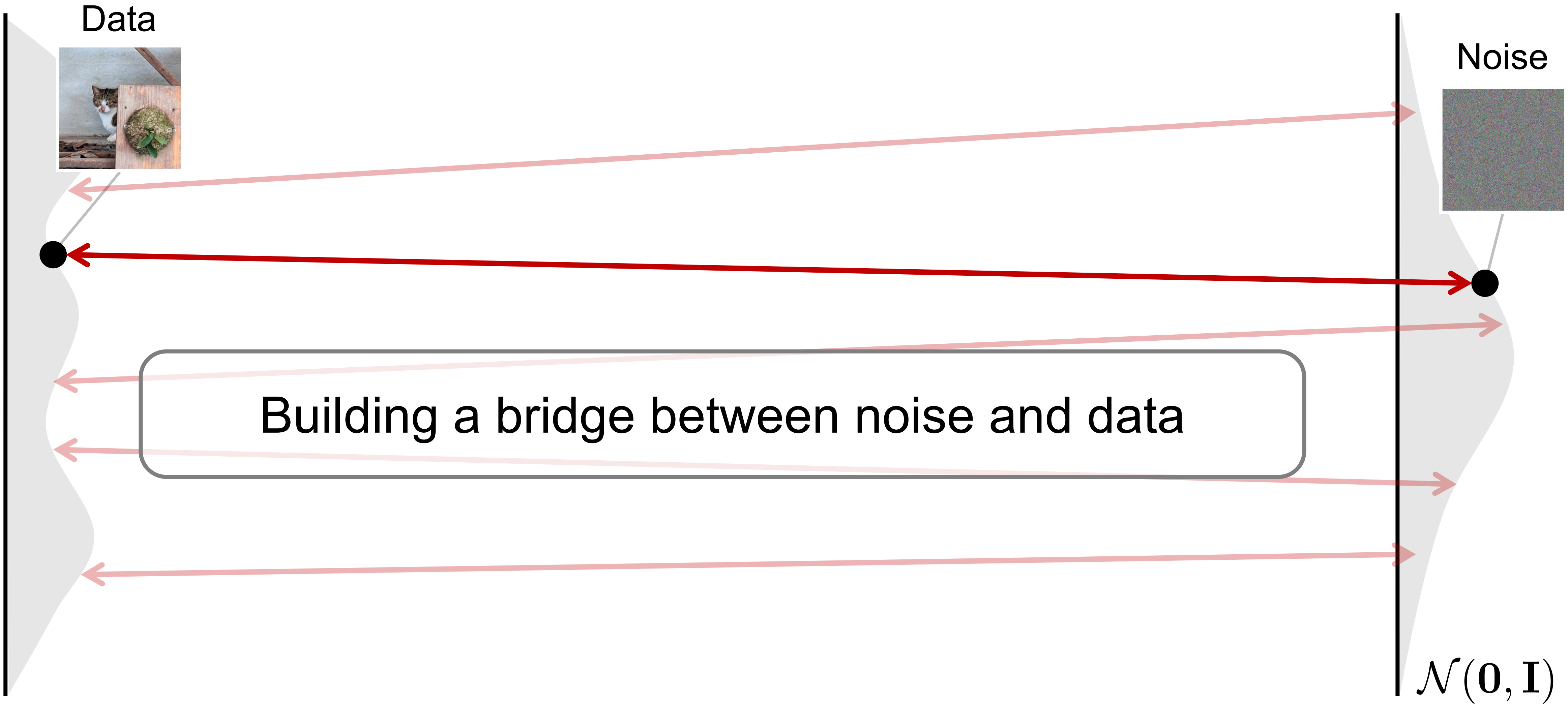


$\mathrm{d}\mathbf{x} = f(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$

drift diffusion



The Goal of Diffusion Models



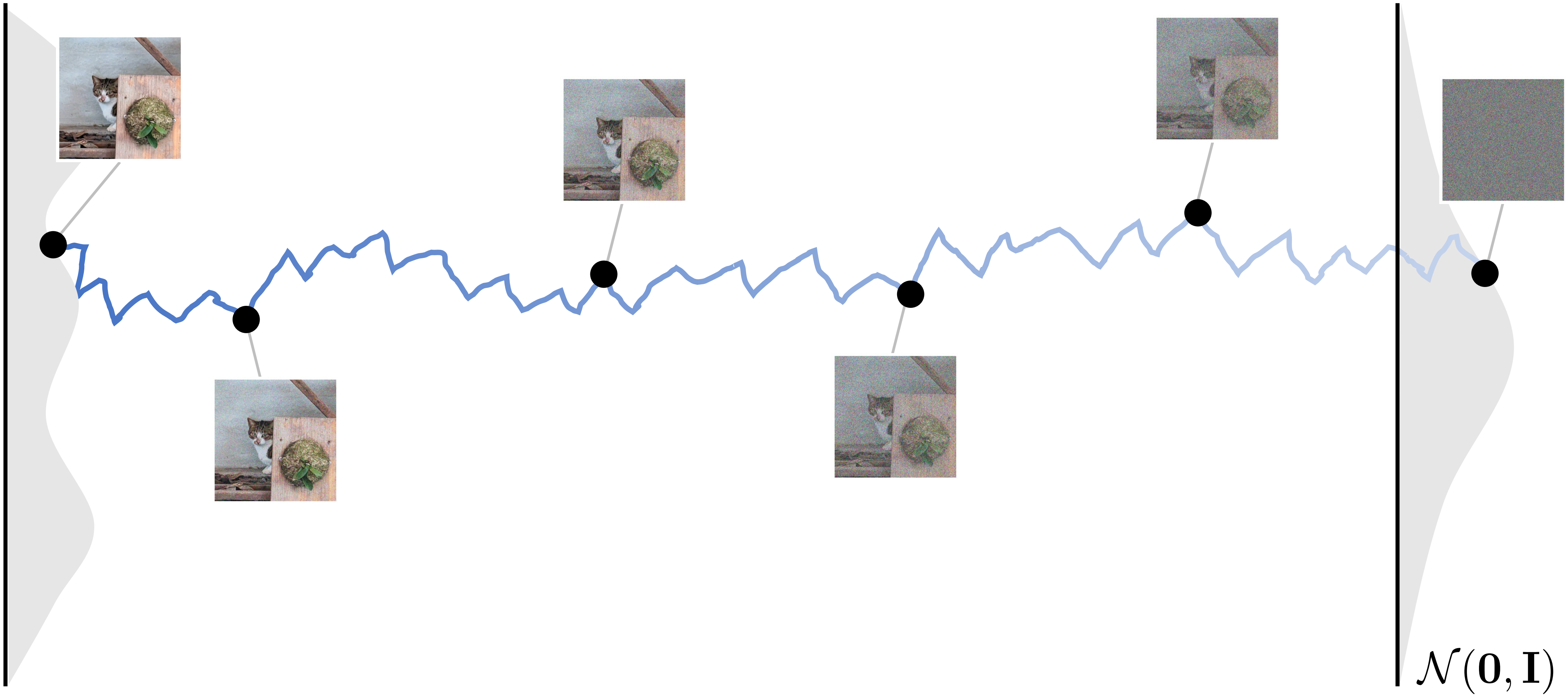
What is Diffusion Model?

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

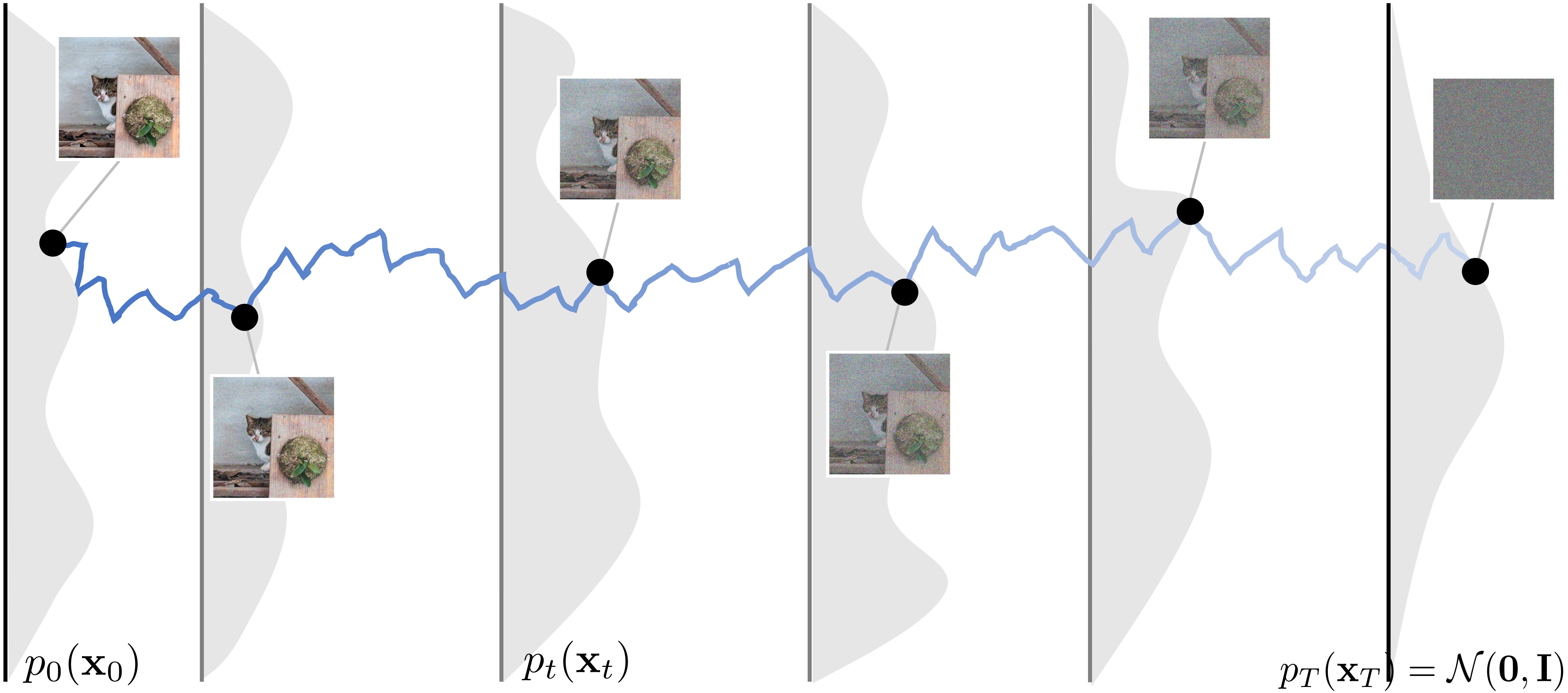
Reverse Process: generate data from pure noise by denoising

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

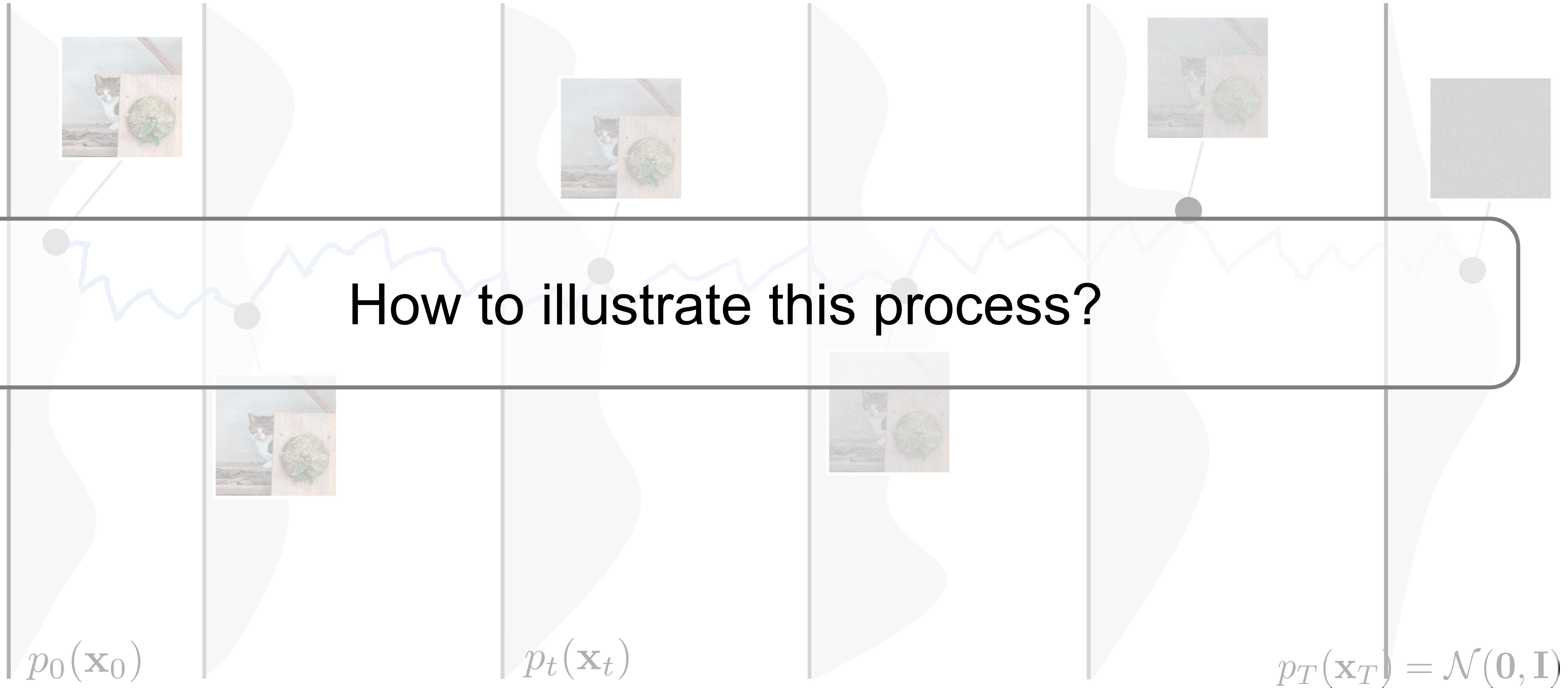
Diffusion Models



Diffusion Models

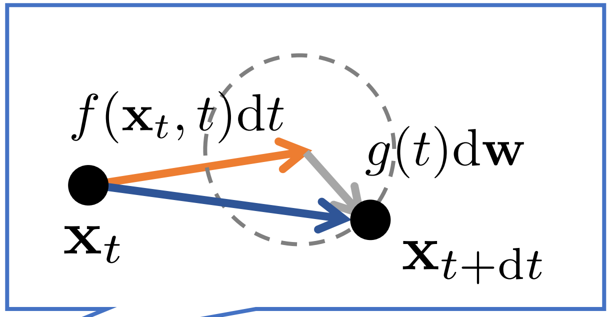


Diffusion Models



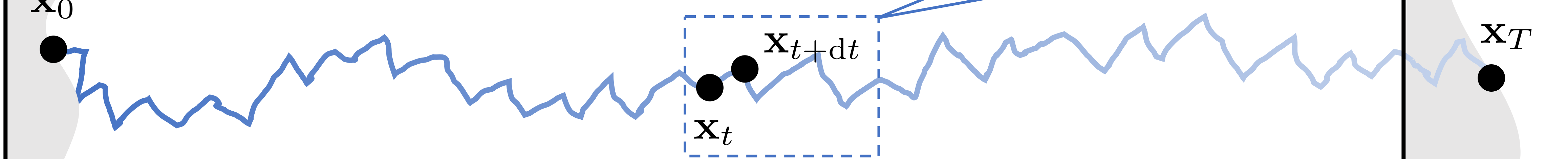
How to illustrate this process?

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

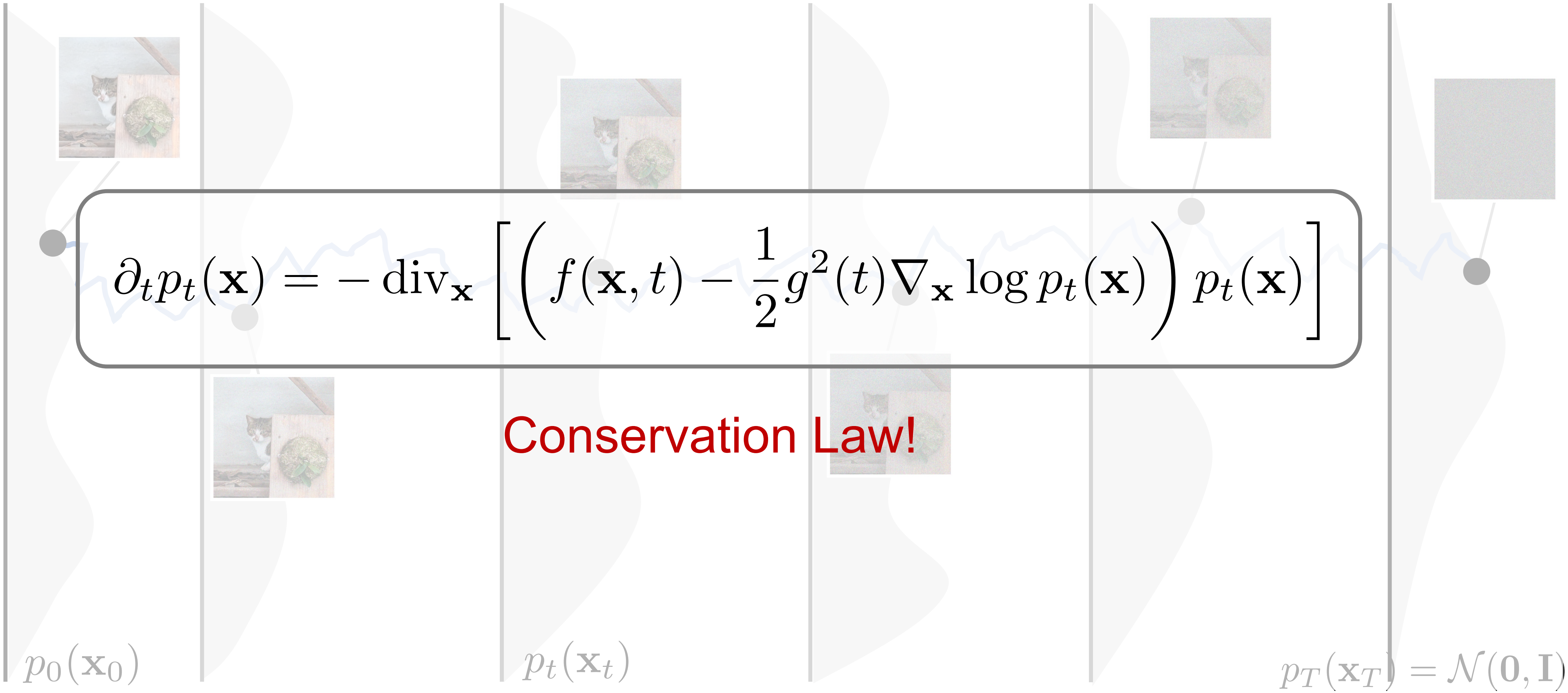
Reverse ODE (Deterministic)

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

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

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

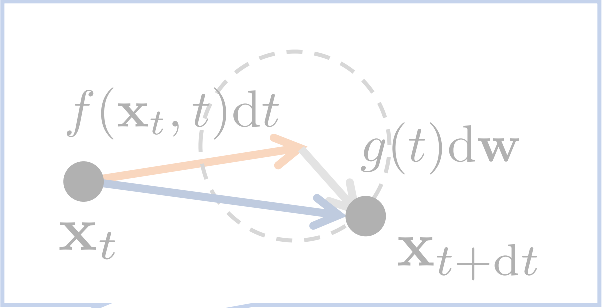
Score-based Diffusion Models [Song+ ICLR'21]



$$\partial_t p_t(\mathbf{x}) = -\operatorname{div}_{\mathbf{x}}\left[\left(f(\mathbf{x}, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right) p_t(\mathbf{x})\right]$$

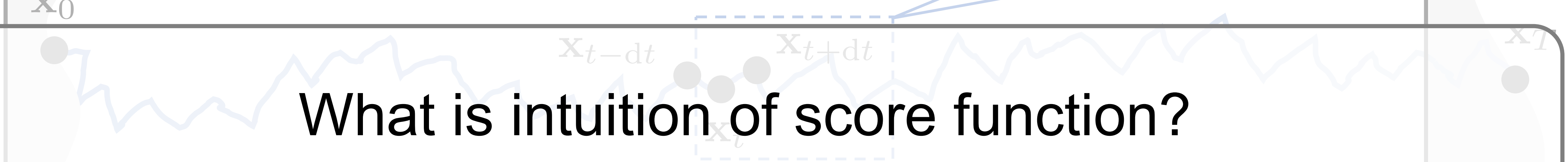
Conservation Law!

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



Forward SDE

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



What is intuition of score function?

Reverse SDE (stochastic)

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

Reverse ODE (deterministic)

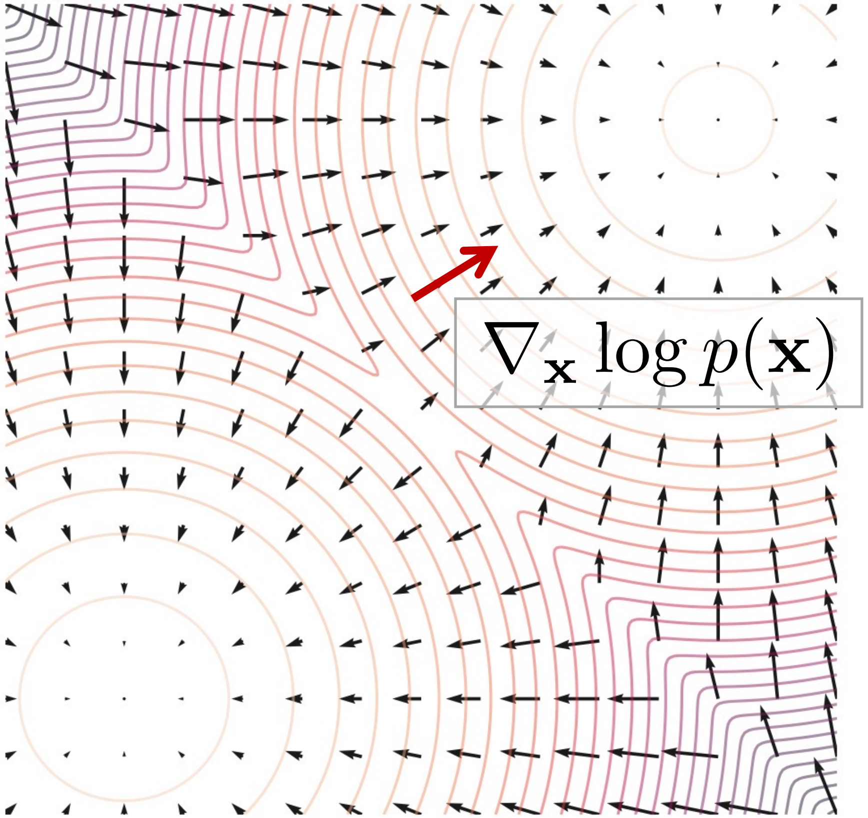
$$\mathrm{d}\mathbf{x}_t = [f(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}]\mathrm{d}t$$

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

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

Score Function

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



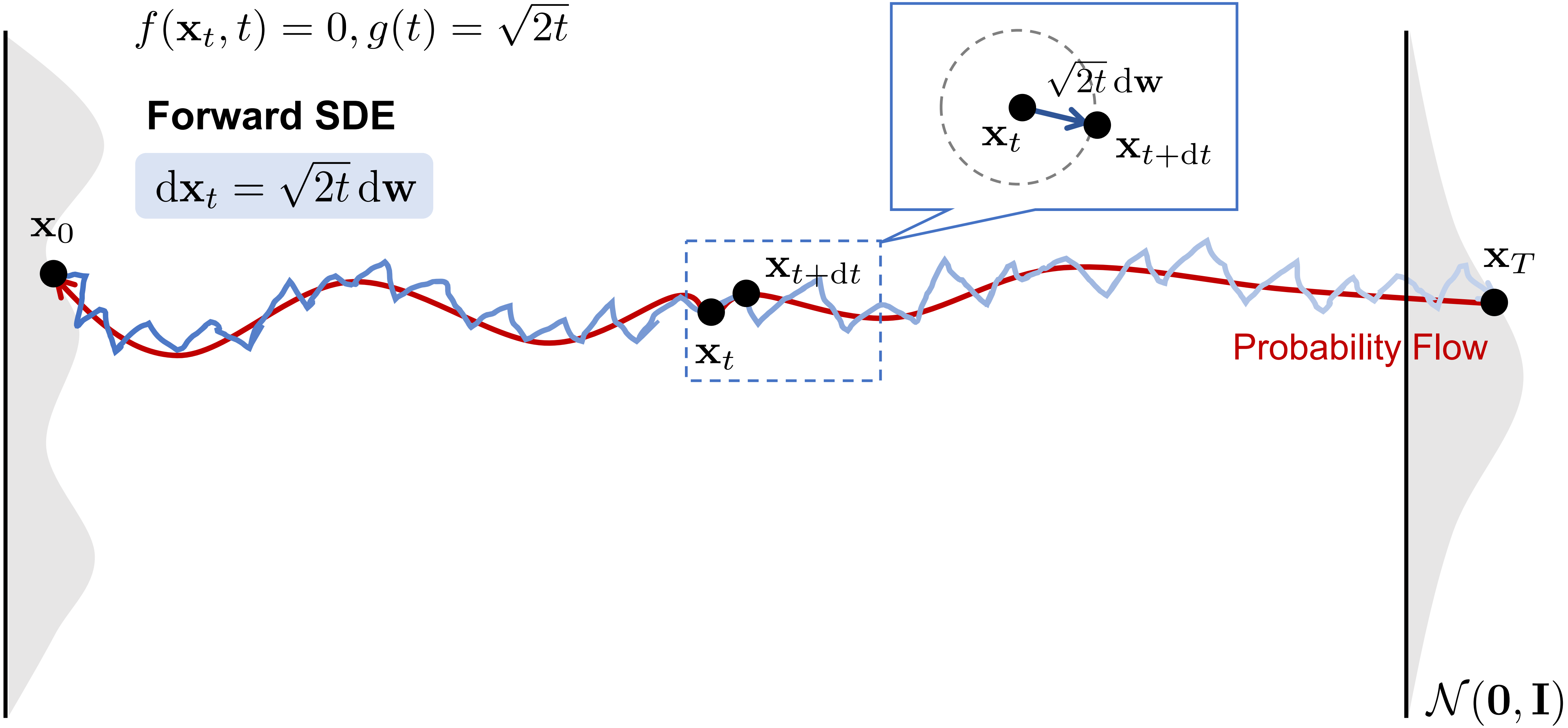
Score Function: The direction in which the probability density increases most rapidly

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

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

Forward SDE

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



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

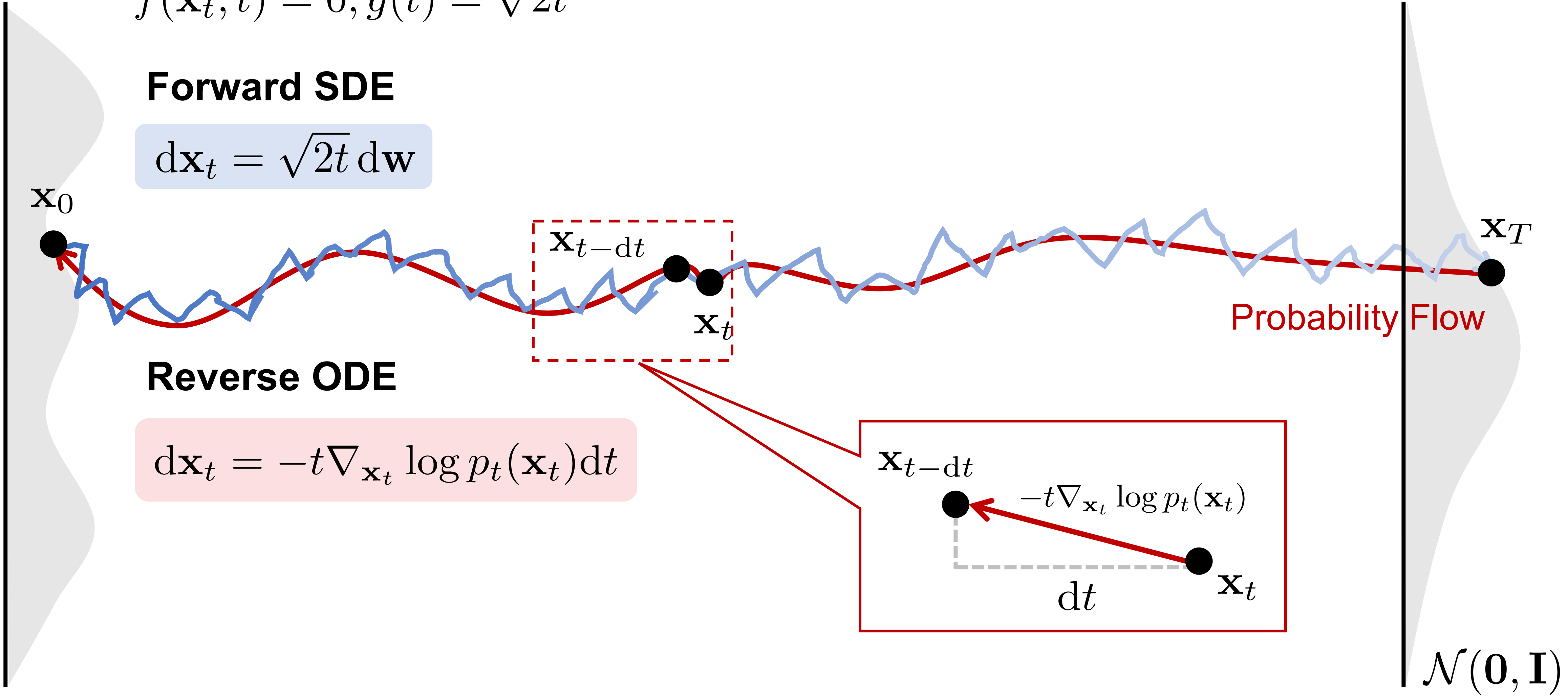
$$f(\mathbf{x}_t, t) = 0, g(t) = \sqrt{2t}$$

Forward SDE

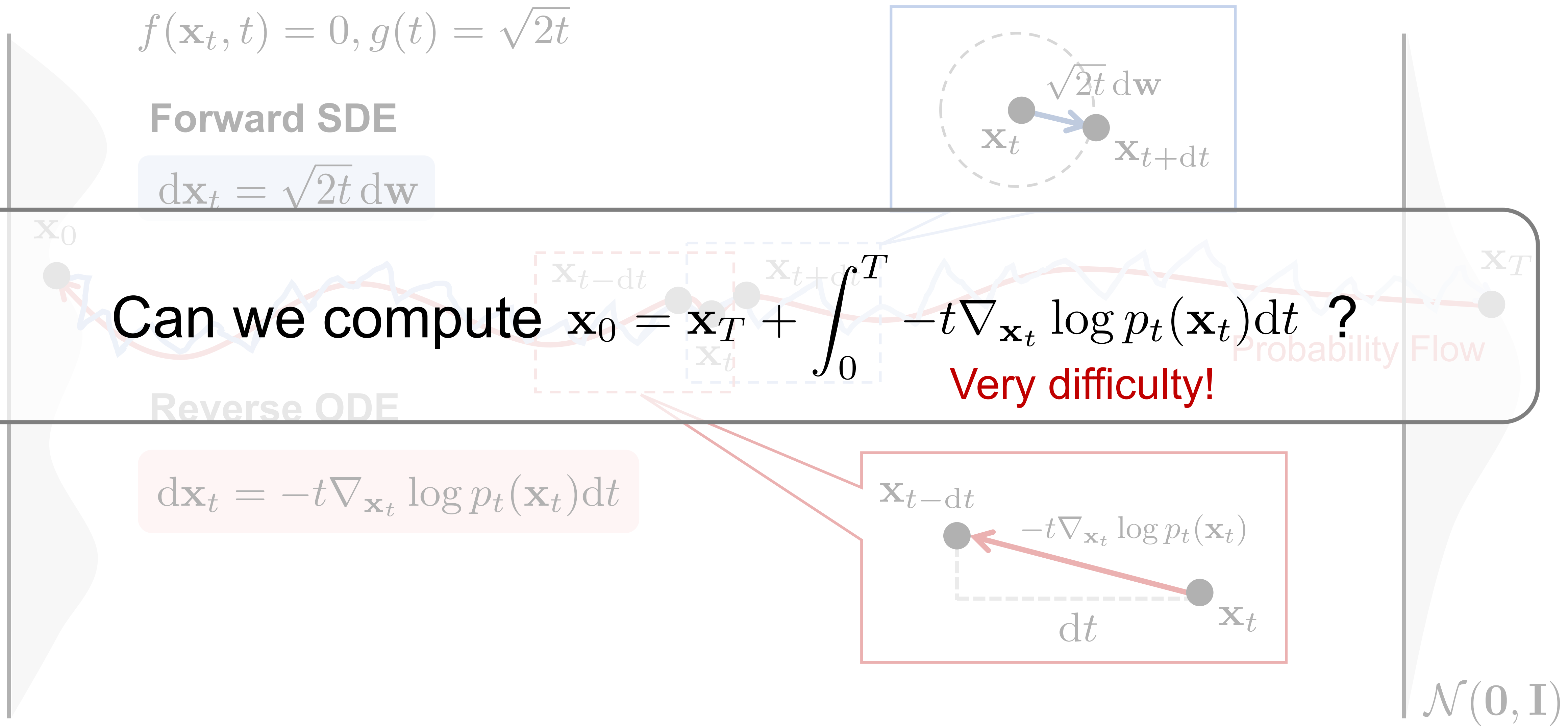
$$\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}$$

Reverse ODE

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



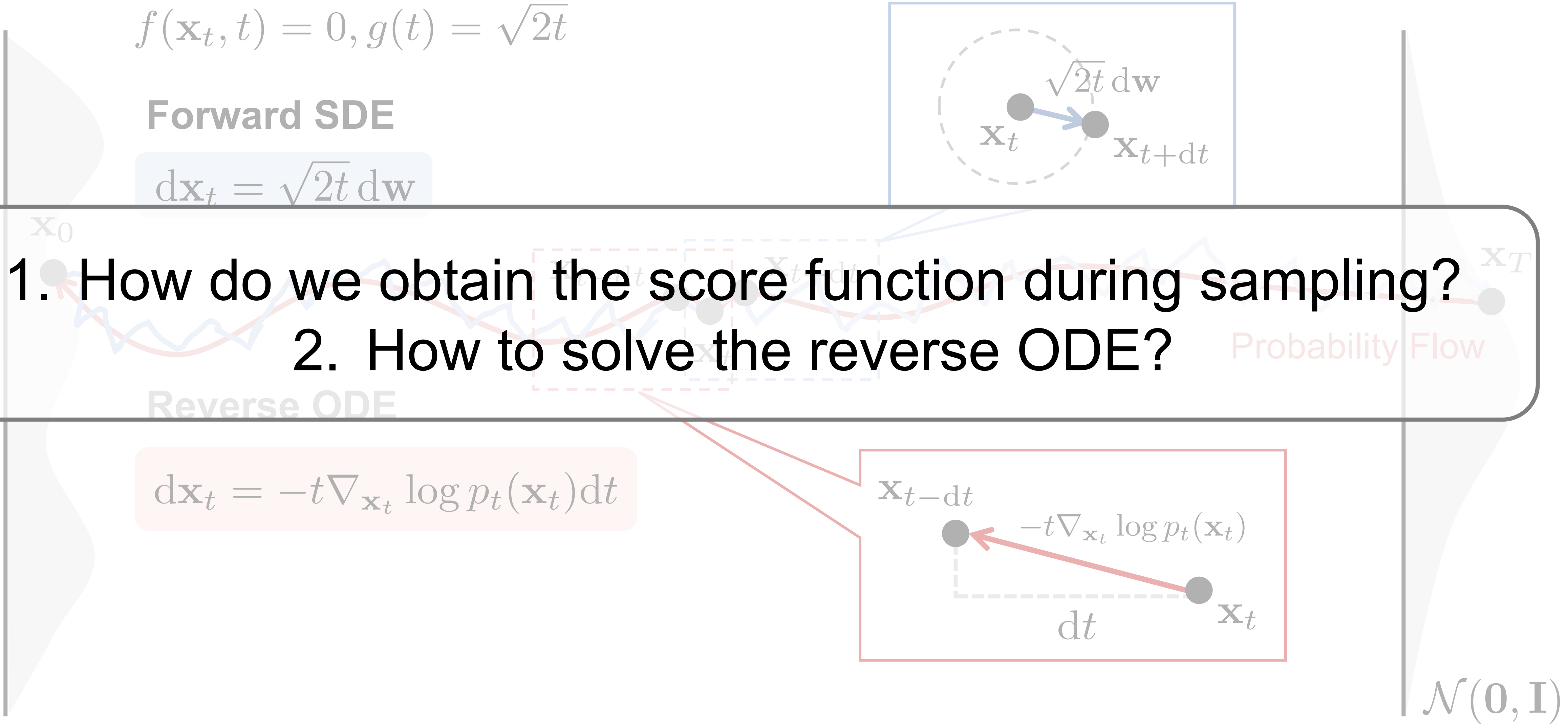
Score-based Diffusion Models [Song+ ICLR'21]



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

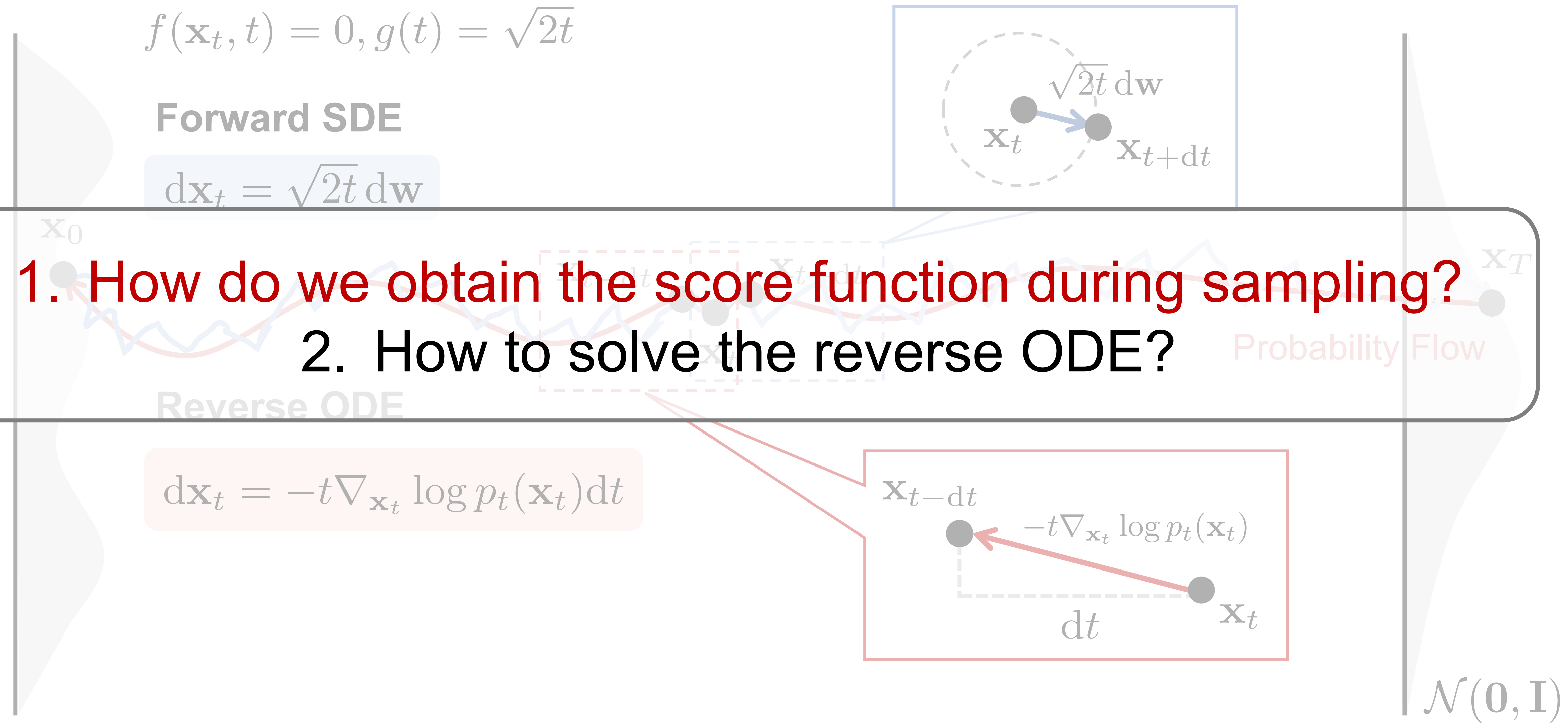
Very difficulty!

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

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

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

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

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]:

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

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) = \frac{\mathbf{x}_0 - \mathbf{x}_t}{\sigma_t^2}$$

noise level

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

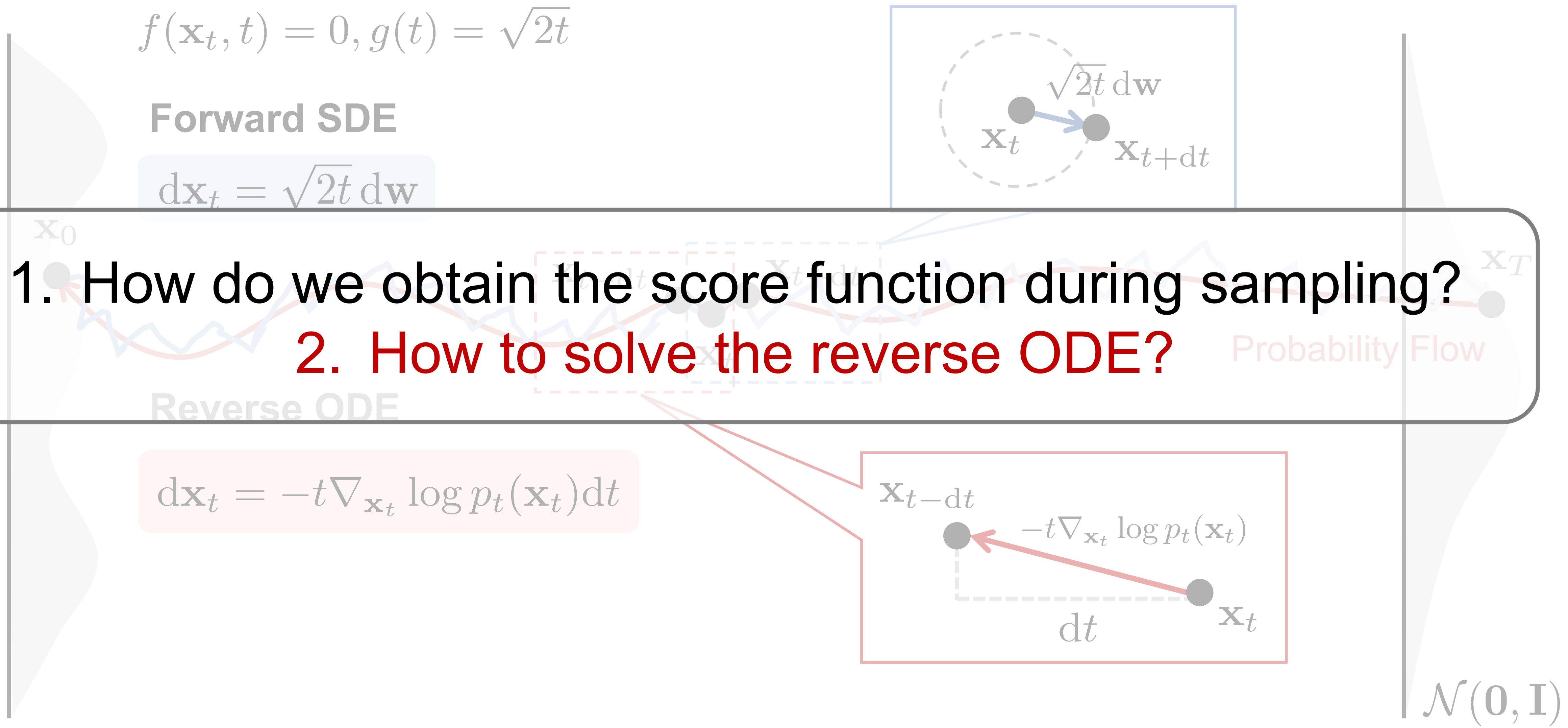
$\mathbf{x}_0 \sim p_{\mathrm{data}}$

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

$\mathbf{x}_t$

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

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



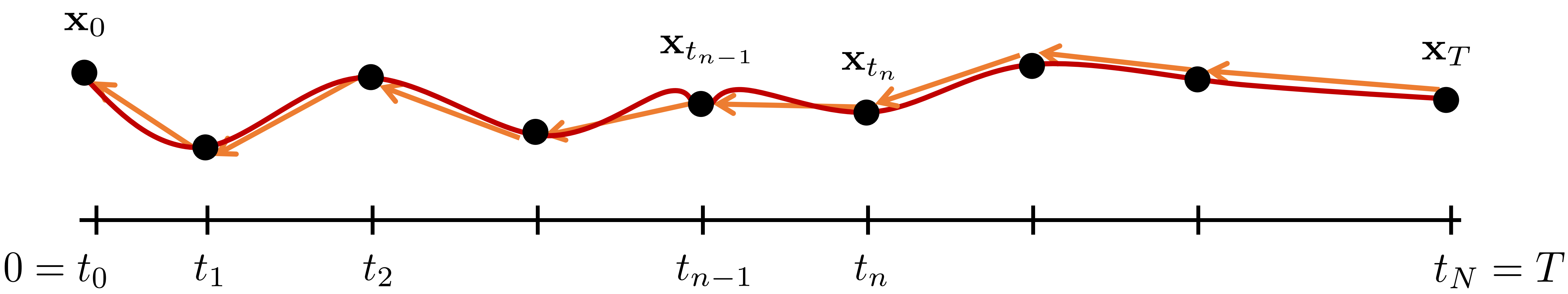
How to Solve the Reverse ODE?

Euler's Method (1st order):

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

$$\wr\wr$$

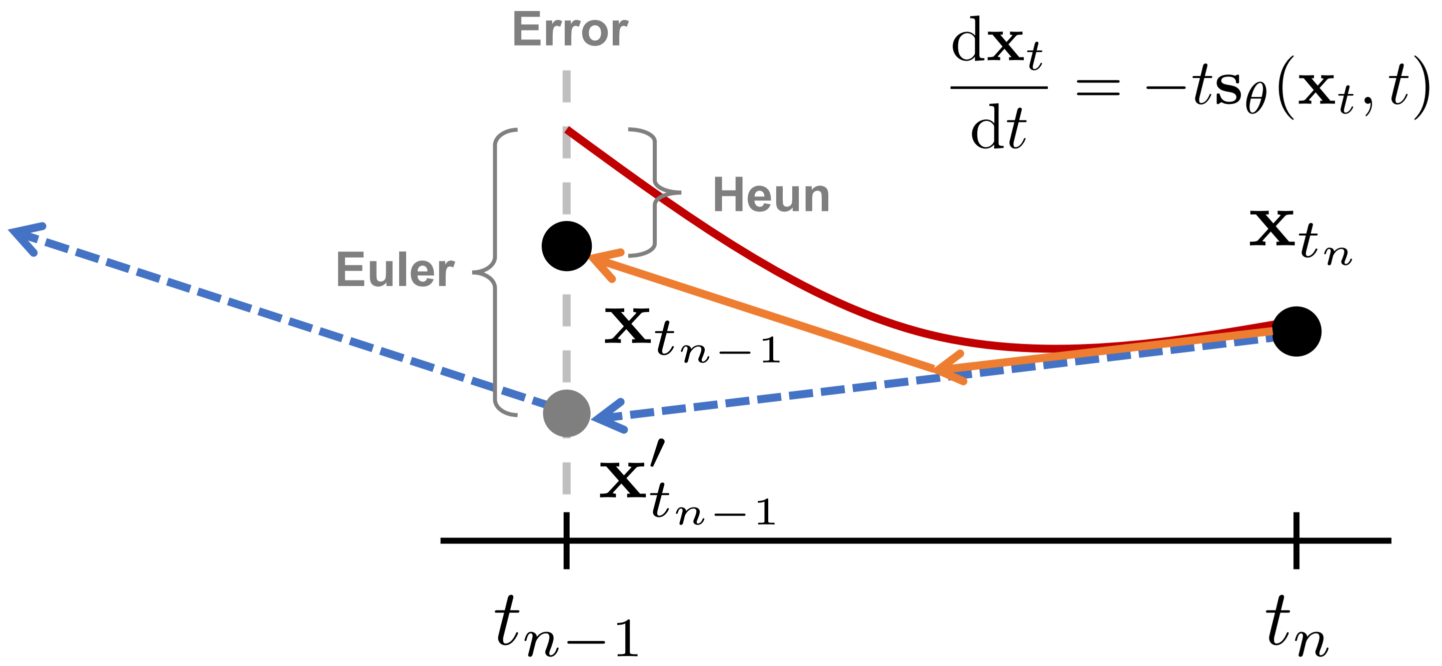
$$\frac{\mathbf{x}_{t_{n-1}} - \mathbf{x}_{t_n}}{t_{n-1} - t_n} + O(n)$$



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

How to Solve the Reverse ODE?

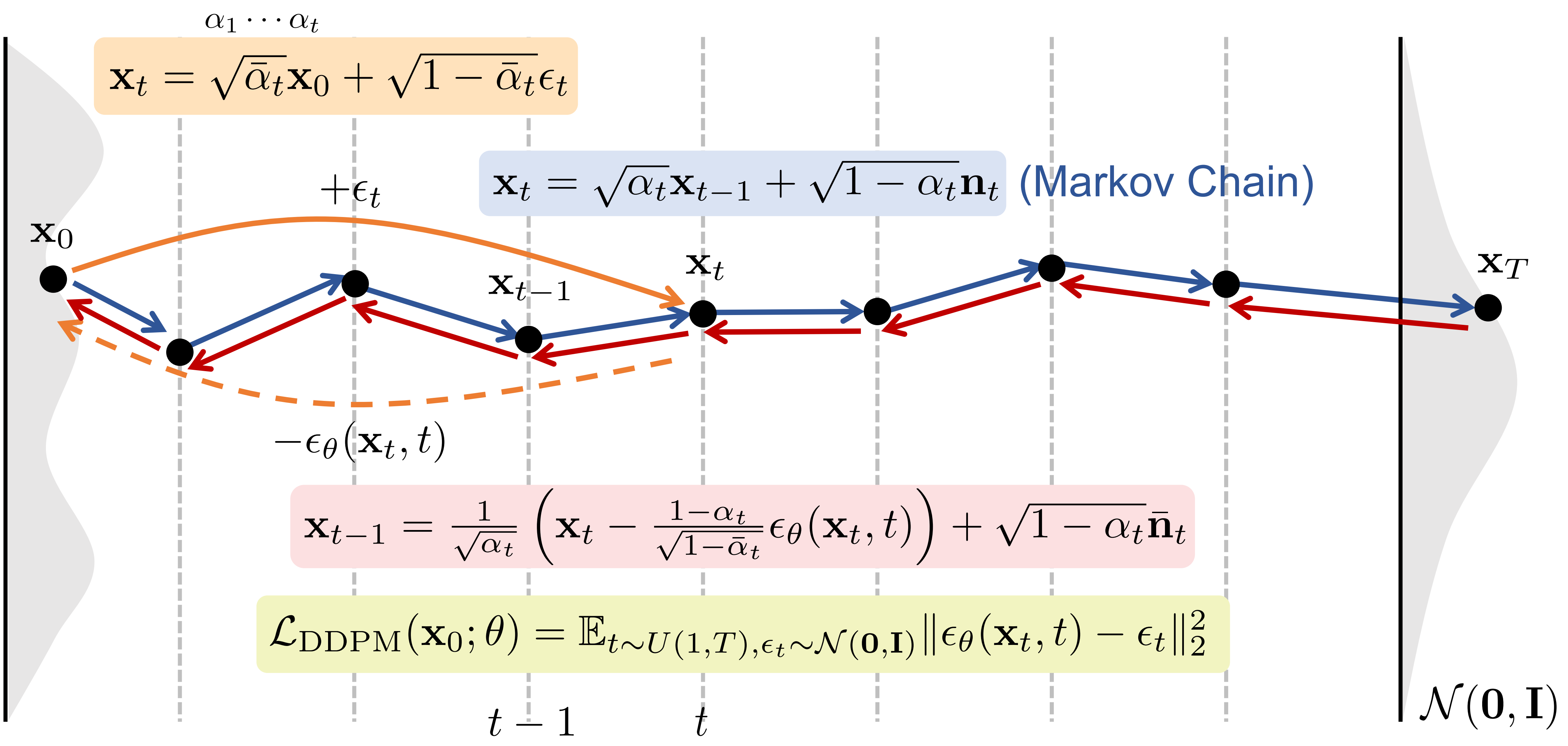
Heun's Method (2nd order):



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

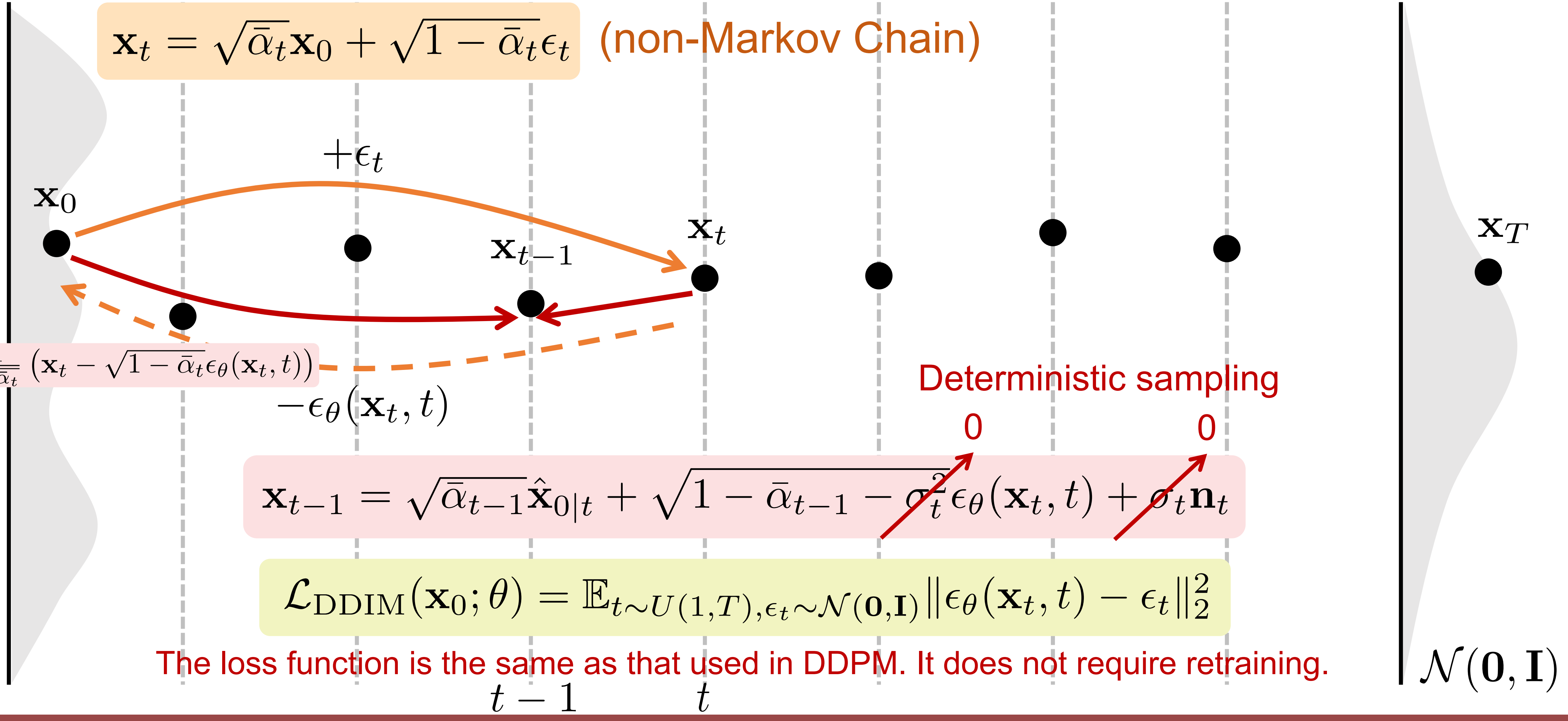
$$\mathbf{x}_{t_{n-1}} = \mathbf{x}_{t_n} - \frac{1}{2}(t_{n-1} - t_n)[t_n\mathbf{s}_\theta(\mathbf{x}_{t_n}, t_n) + t_{n-1}\mathbf{s}_\theta(\mathbf{x}'_{t_{n-1}}, t_{n-1})]$$

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



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

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$$ (non-Markov Chain)



$$\hat{\mathbf{x}}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t)\right)$$

Deterministic sampling

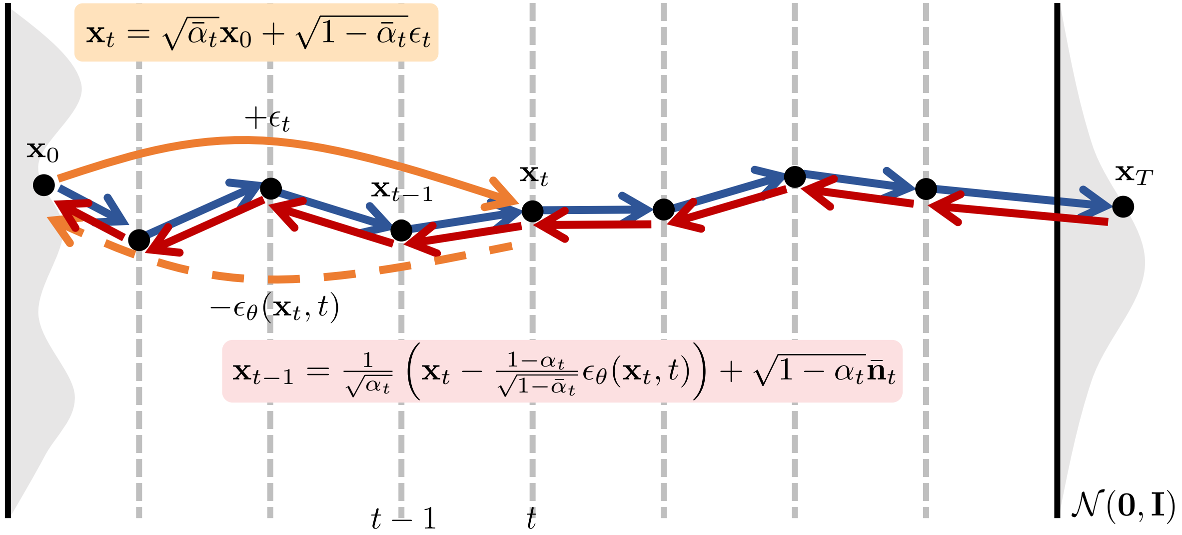
$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}-\cancelto{0}{\sigma_t^2}}\,\epsilon_\theta(\mathbf{x}_t, t) + \cancelto{0}{\sigma_t}\mathbf{n}_t$$

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

The loss function is the same as that used in DDPM. It does not require retraining.

Summary: Discreate vs Continuous

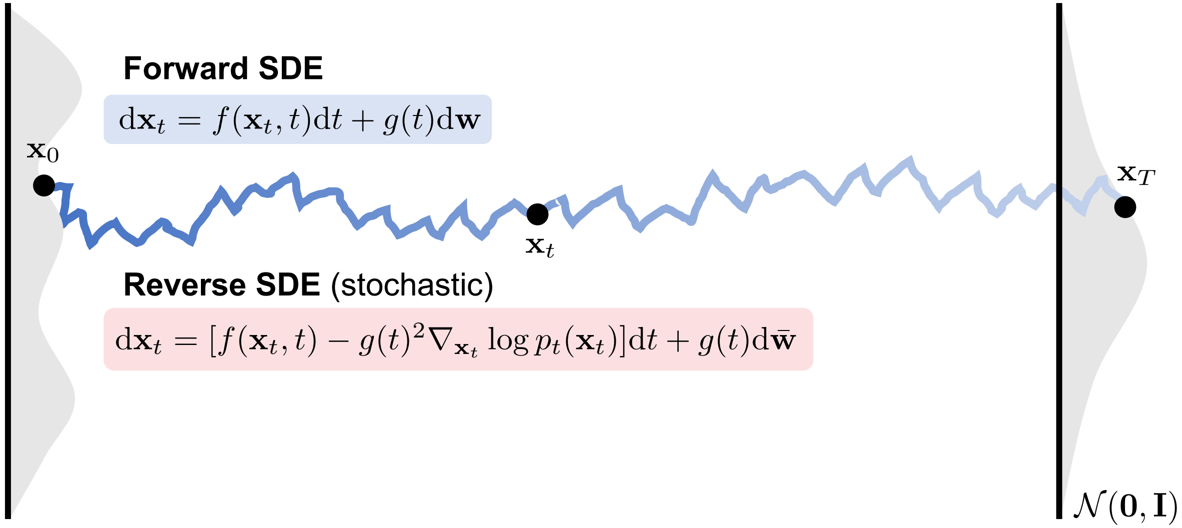
DDPM



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

Discretized in the training phase

Score-based SDE



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

Discretized during sampling

The error is only affected by the order of the solver

Experiment Results

Table 2: NLLs and FIDs (ODE) on CIFAR-10.

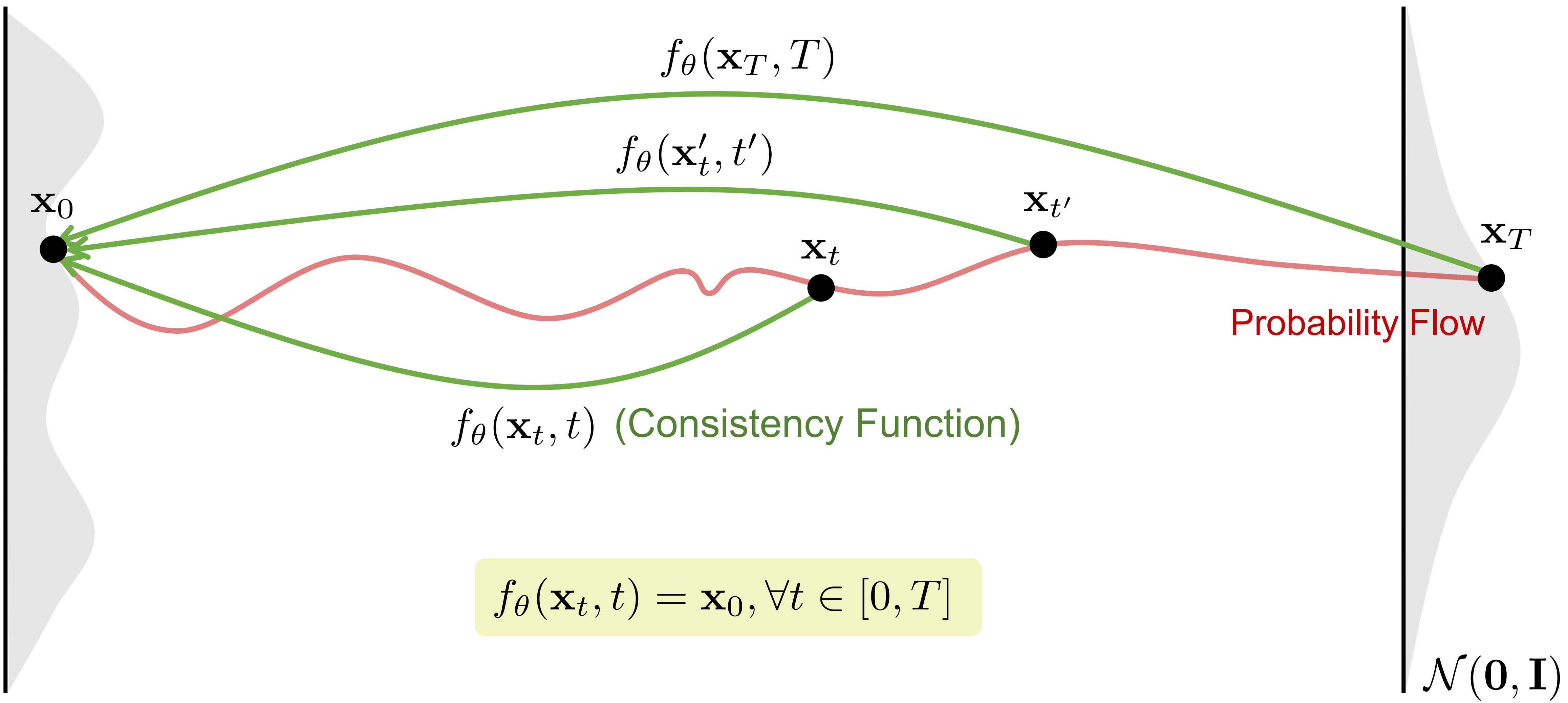
Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM (L) (Ho et al., 2020)	$\leqslant 3.70^{*}$	13.51
DDPM (L_{simple}) (Ho et al., 2020)	$\leqslant 3.75^{*}$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	**2.99**	**2.92**

Table 3: CIFAR-10 sample quality.

Model	FID↓	IS↑
Conditional		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	**2.42**	**10.14**
Unconditional		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	$8.87 \pm .12$
NCSNv2 (Song & Ermon, 2020)	10.87	$8.40 \pm .07$
DDPM (Ho et al., 2020)	3.17	$9.46 \pm .11$
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	**2.20**	**9.89**

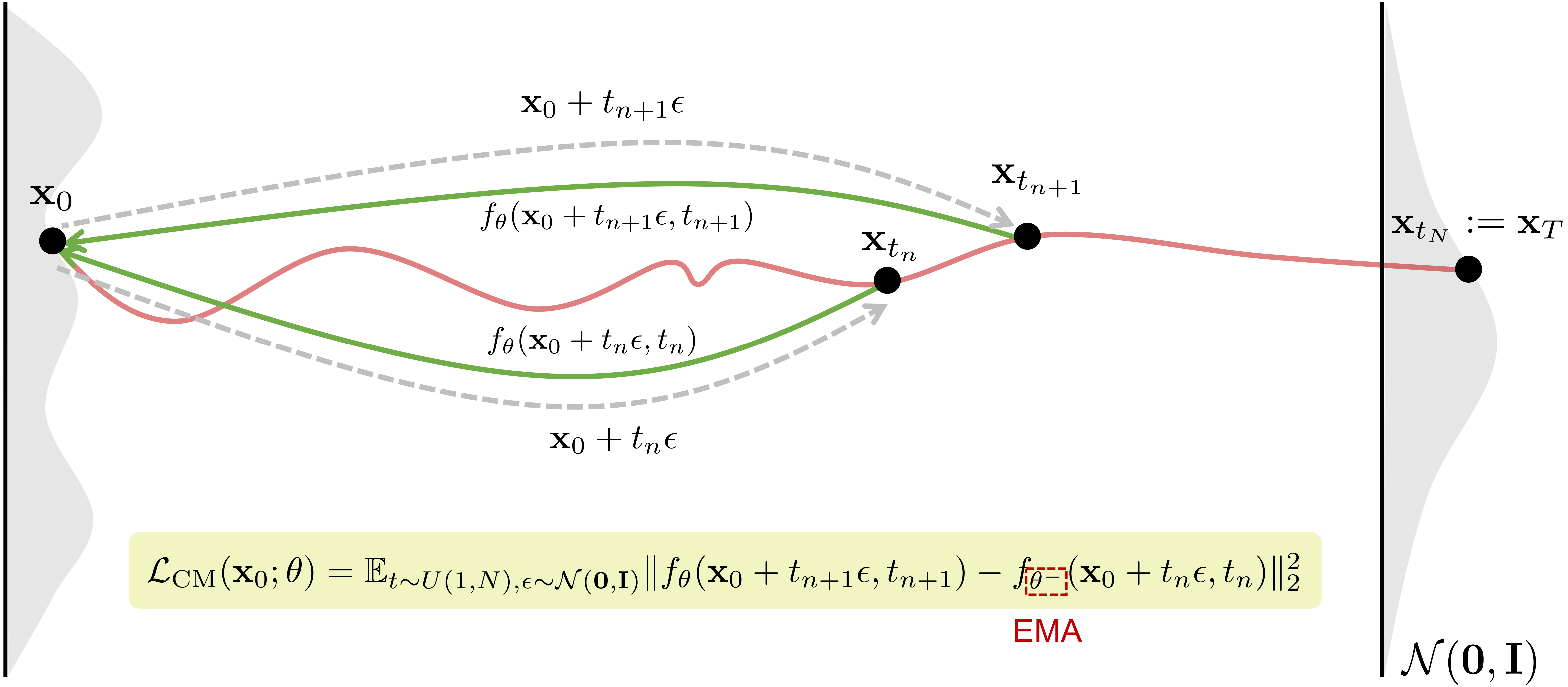
Consistency Models (CM) [Song+ ICML'23]

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



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

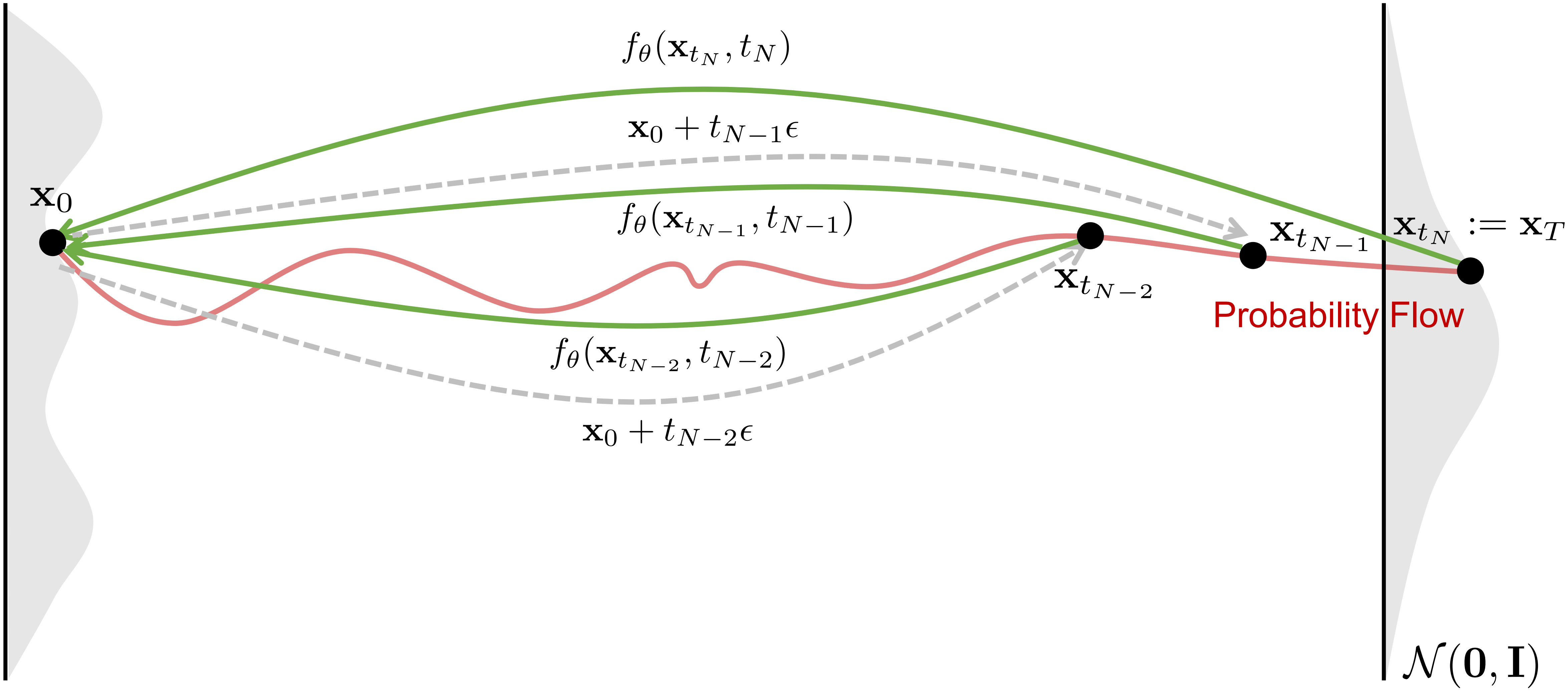
Training CM



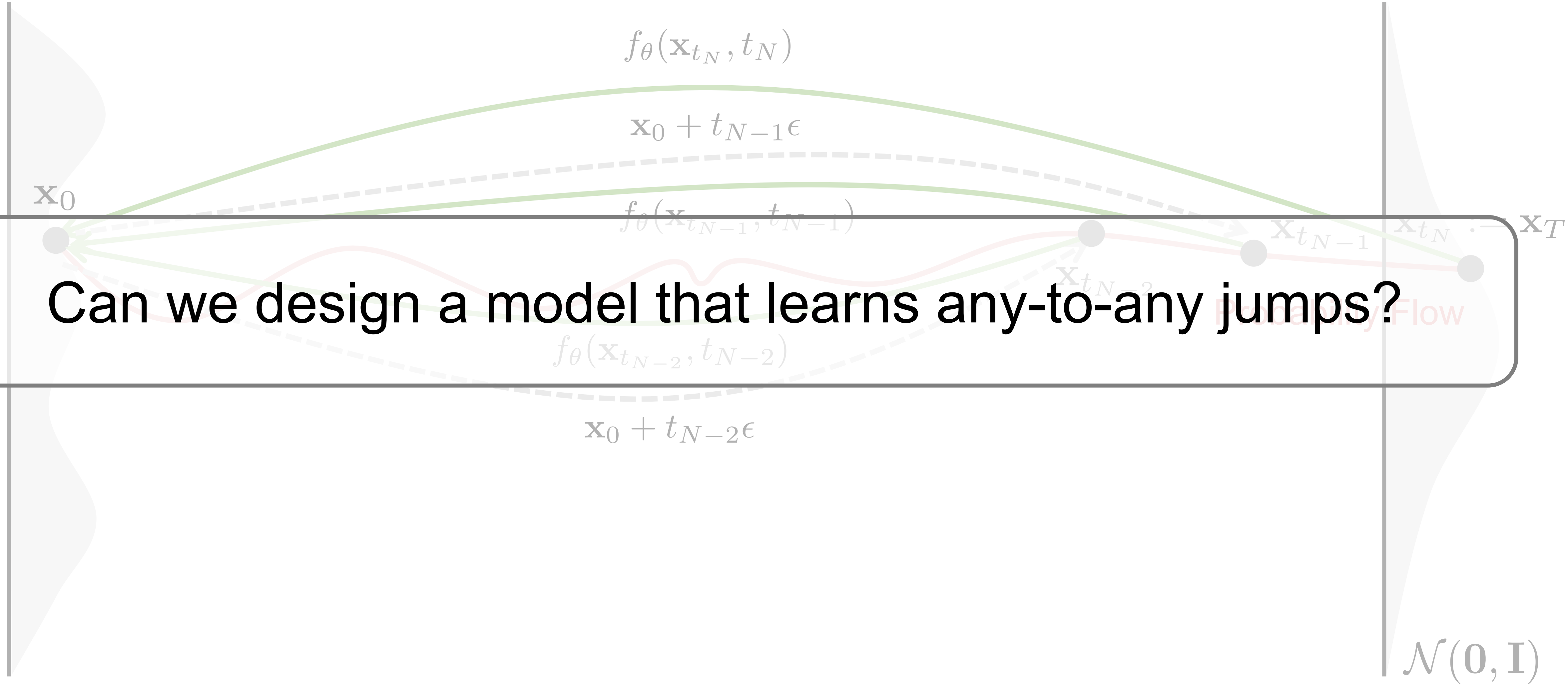
$$\mathcal{L}_{\mathrm{CM}}(\mathbf{x}_0;\theta) = \mathbb{E}_{t\sim U(1,N),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\|f_\theta(\mathbf{x}_0 + t_{n+1}\epsilon, t_{n+1}) - f_{\boxed{\theta^-}}(\mathbf{x}_0 + t_n\epsilon, t_n)\|_2^2$$

EMA

Sampling with CM



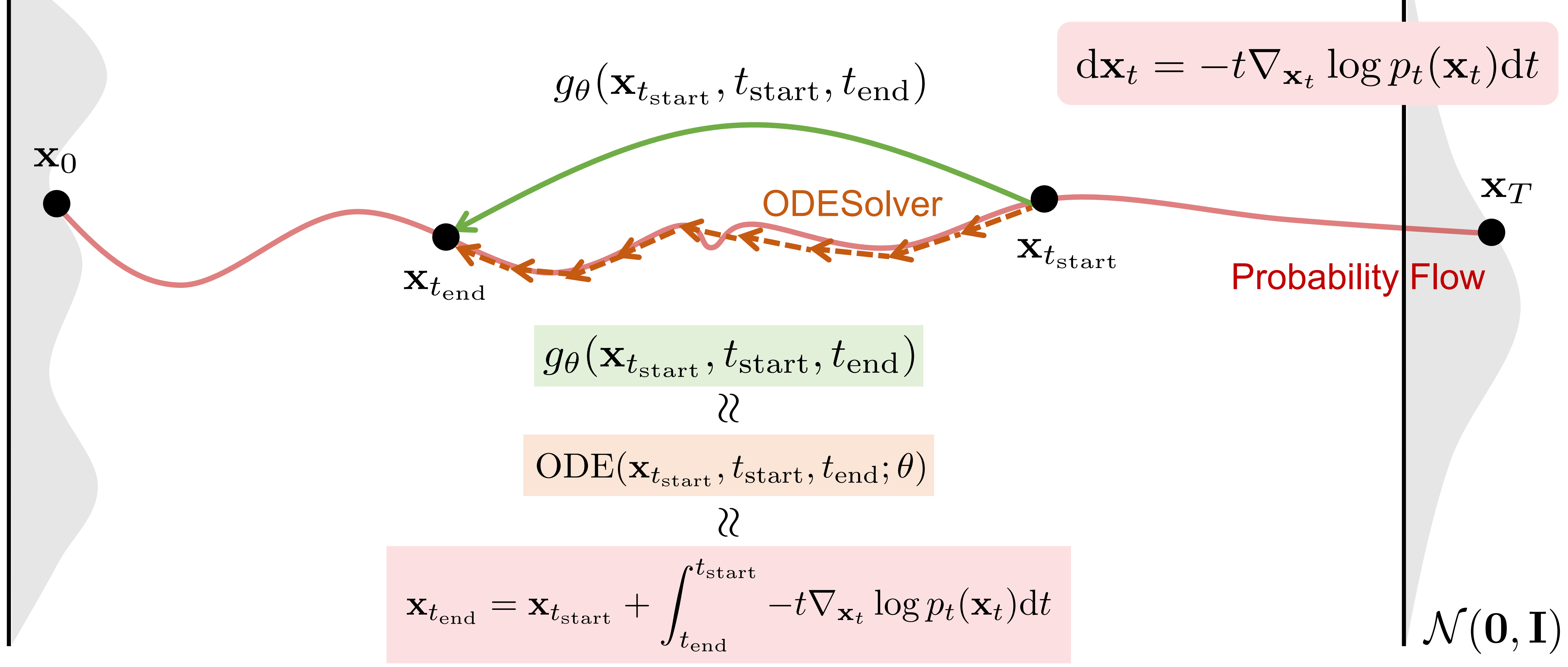
Sampling with CM



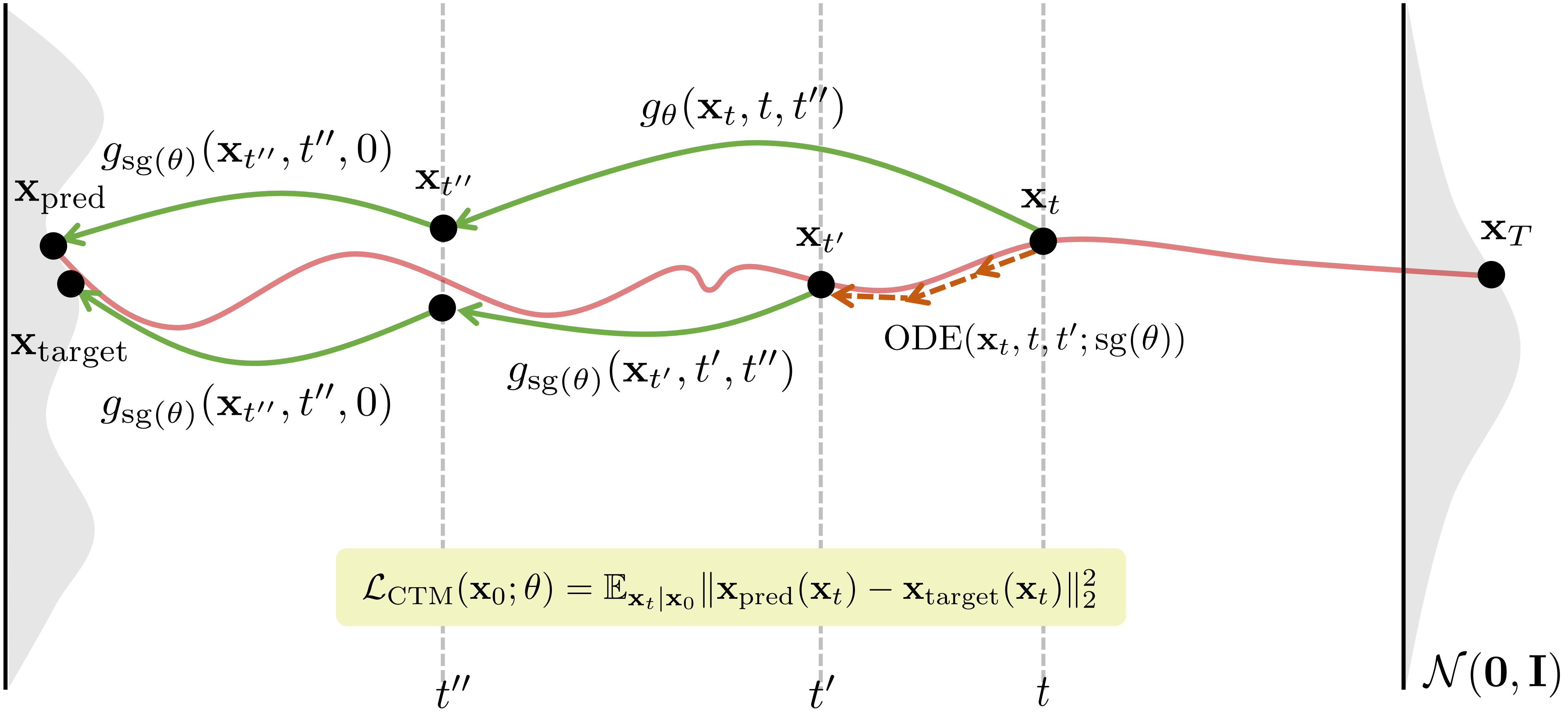
Can we design a model that learns any-to-any jumps?

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

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



Training CTM



$$\mathcal{L}_{\text{CTM}}(\mathbf{x}_0; \theta) = \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \|\mathbf{x}_{\text{pred}}(\mathbf{x}_t) - \mathbf{x}_{\text{target}}(\mathbf{x}_t)\|_2^2$$

Experiment Results

Table 1: Sample quality on CIFAR-10. *Methods that require synthetic data construction for distillation.

METHOD	NFE (↓)	FID (↓)	IS (↑)
Diffusion + Samplers			
DDIM (Song et al., 2020)	50	4.67	
DDIM (Song et al., 2020)	20	6.84	
DDIM (Song et al., 2020)	10	8.23	
DPM-solver-2 (Lu et al., 2022)	10	5.94	
DPM-solver-fast (Lu et al., 2022)	10	4.70	
3-DEIS (Zhang & Chen, 2022)	10	**4.17**	
Diffusion + Distillation			
Knowledge Distillation* (Luhman & Luhman, 2021)	1	9.36	
DFNO* (Zheng et al., 2022)	1	4.12	
1-Rectified Flow (+distill)* (Liu et al., 2022)	1	6.18	9.08
2-Rectified Flow (+distill)* (Liu et al., 2022)	1	4.85	9.01
3-Rectified Flow (+distill)* (Liu et al., 2022)	1	5.21	8.79
PD (Salimans & Ho, 2022)	1	8.34	8.69
CD	1	**3.55**	**9.48**
PD (Salimans & Ho, 2022)	2	5.58	9.05
CD	2	**2.93**	**9.75**
Direct Generation			
BigGAN (Brock et al., 2019)	1	14.7	9.22
Diffusion GAN (Xiao et al., 2022)	1	14.6	8.93
AutoGAN (Gong et al., 2019)	1	12.4	8.55
E2GAN (Tian et al., 2020)	1	11.3	8.51
ViTGAN (Lee et al., 2021)	1	6.66	9.30
TransGAN (Jiang et al., 2021)	1	9.26	9.05
StyleGAN2-ADA (Karras et al., 2020)	1	2.92	**9.83**
StyleGAN-XL (Sauer et al., 2022)	1	**1.85**	
Score SDE (Song et al., 2021)	2000	2.20	**9.89**
DDPM (Ho et al., 2020)	1000	3.17	9.46
LSGM (Vahdat et al., 2021)	147	2.10	
PFGM (Xu et al., 2022)	110	2.35	9.68
EDM (Karras et al., 2022)	35	**2.04**	9.84
1-Rectified Flow (Liu et al., 2022)	1	378	1.13
Glow (Kingma & Dhariwal, 2018)	1	48.9	3.92
Residual Flow (Chen et al., 2019)	1	46.4	
GLFlow (Xiao et al., 2019)	1	44.6	
DenseFlow (Grcić et al., 2021)	1	34.9	
DC-VAE (Parmar et al., 2021)	1	17.9	8.20
CT	1	**8.70**	**8.49**
CT	2	**5.83**	**8.85**

Table 2: Sample quality on ImageNet 64 × 64, and LSUN Bedroom & Cat 256 × 256. †Distillation techniques.

METHOD	NFE (↓)	FID (↓)	Prec. (↑)	Rec. (↑)
ImageNet 64 × 64				
PD† (Salimans & Ho, 2022)	1	15.39	0.59	0.62
DFNO† (Zheng et al., 2022)	1	8.35		
CD†	1	6.20	0.68	0.63
PD† (Salimans & Ho, 2022)	2	8.95	0.63	**0.65**
CD†	2	**4.70**	**0.69**	0.64
ADM (Dhariwal & Nichol, 2021)	250	**2.07**	0.74	0.63
EDM (Karras et al., 2022)	79	2.44	0.71	**0.67**
BigGAN-deep (Brock et al., 2019)	1	4.06	**0.79**	0.48
CT	1	13.0	0.71	0.47
CT	2	11.1	0.69	0.56
LSUN Bedroom 256 × 256				
PD† (Salimans & Ho, 2022)	1	16.92	0.47	0.27
PD† (Salimans & Ho, 2022)	2	8.47	0.56	**0.39**
CD†	1	7.80	0.66	0.34
CD†	2	**5.22**	**0.68**	**0.39**
DDPM (Ho et al., 2020)	1000	4.89	0.60	0.45
ADM (Dhariwal & Nichol, 2021)	1000	**1.90**	0.66	**0.51**
EDM (Karras et al., 2022)	79	3.57	0.66	0.45
PGGAN (Karras et al., 2018)	1	8.34		
PG-SWGAN (Wu et al., 2019)	1	8.0		
TDPM (GAN) (Zheng et al., 2023)	1	5.24		
StyleGAN2 (Karras et al., 2020)	1	2.35	0.59	0.48
CT	1	16.0	0.60	0.17
CT	2	7.85	**0.68**	0.33
LSUN Cat 256 × 256				
PD† (Salimans & Ho, 2022)	1	29.6	0.51	0.25
PD† (Salimans & Ho, 2022)	2	15.5	0.59	0.36
CD†	1	11.0	0.65	0.36
CD†	2	**8.84**	**0.66**	**0.40**
DDPM (Ho et al., 2020)	1000	17.1	0.53	0.48
ADM (Dhariwal & Nichol, 2021)	1000	**5.57**	0.63	**0.52**
EDM (Karras et al., 2022)	79	6.69	**0.70**	0.43
PGGAN (Karras et al., 2018)	1	37.5		
StyleGAN2 (Karras et al., 2020)	1	7.25	0.58	0.43
CT	1	20.7	0.56	0.23
CT	2	11.7	0.63	0.36

Table 1: Performance comparisons on CIFAR-10[9].

Model	NFE	Unconditional		Conditional
		FID↓	NLL↓	FID↓
GAN Models				
BigGAN (Brock et al., 2018)	1	8.51	✗	-
StyleGAN-Ada (Karras et al., 2020)	1	2.92	✗	2.42
StyleGAN-D2D (Kang et al., 2021)	1	-	✗	2.26
StyleGAN-XL (Sauer et al., 2022)	1	-	✗	1.85
Diffusion Models – Score-based Sampling				
DDPM (Ho et al., 2020)	1000	3.17	3.75	-
DDIM (Song et al., 2020a)	100	4.16	-	-
	10	13.36	-	-
Score SDE (Song et al., 2020a)	2000	2.20	3.45	-
VDM (Kingma et al., 2021)	1000	7.41	2.49	-
LSGM (Vahdat et al., 2021)	138	2.10	3.43	-
EDM (Karras et al., 2022)	35	2.01	2.56	1.82
Diffusion Models – Distillation Sampling				
KD (Luhman & Luhman, 2021)	1	9.36	✗	-
DFNO (Zheng et al., 2023)	1	3.78	✗	-
2-Rectified Flow (Liu et al., 2022)	1	4.85	✗	-
PD (Salimans & Ho, 2021)	1	9.12	✗	-
CD (official report) (Song et al., 2023)	1	3.55	✗	-
CD (retrained)	1	10.53	✗	-
CD + GAN (Lu et al., 2023)	1	2.65	✗	-
CTM (ours)	1	1.98	**2.43**	1.73
PD (Salimans & Ho, 2021)	2	4.51	-	-
CD (Song et al., 2023)	2	2.93	-	-
CTM (ours)	2	**1.87**	**2.43**	**1.63**
Models without Pre-trained DM – Direct Generation				
CT	1	8.70	✗	-
CTM (ours)	1	2.39	-	-

Takeaway

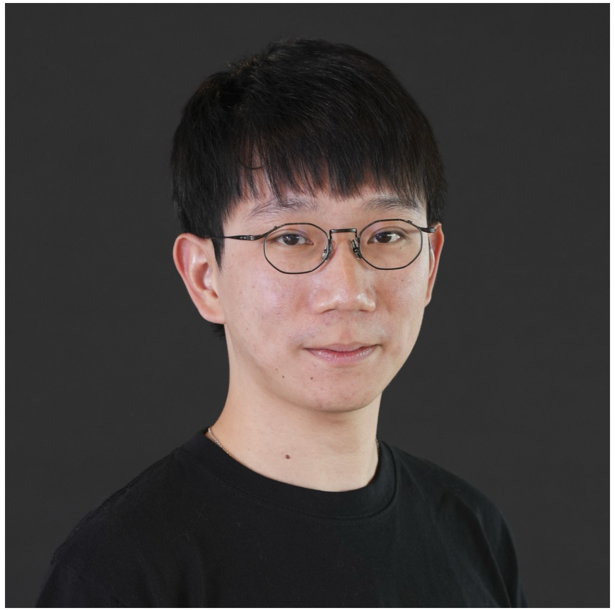
- Diffusion models build a bridge between noise and data, forming a powerful generative modeling framework.
- Score-based models leverage SDE/ODE formulations and score functions to guide the reverse process.
- Variants like DDIM, CM, CTM offer trade-offs in speed, quality, and control.

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

Reference

1. Jonathan Ho et al. Denoising Diffusion Probabilistic Models, *NeurIPS 2020.*
2. Jiaming Song et al. Denoising Diffusion Implicit Models, *ICLR 2021.*
3. Yang Song et al. Score-Based Generative Modeling through Stochastic Differential Equations, *ICLR 2021.*
4. Yang Song et al. Consistency Models, *ICML 2023.*
5. Dongjun Kim et al. Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion, *ICLR 2024.*

Thank you