Towards One-step Diffusion and Flow

From Consistency Model to Flow Map Models

[Jia-Wei Liao]

Ph.D. Candidate in Computer Science

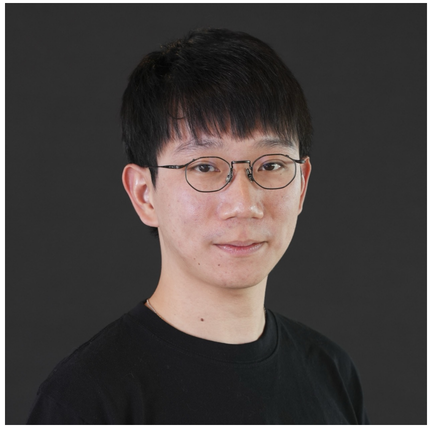
National Taiwan University

What Will We Cover Today?

- Recap Diffusion Models and Flow Matching
- **Consistency Models**
- Flow Maps Models: **Consistency Trajectory Models**, **MeanFlow**

Yang Song

Research Scientist at OpenAI



Chieh-Hsin (Jesse) Lai

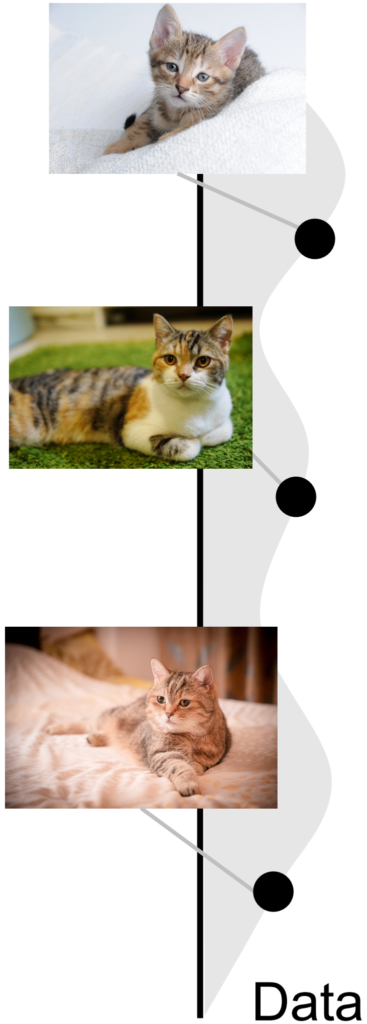
Research Scientist at Sony AI

Kaiming He

Associate Professor at MIT

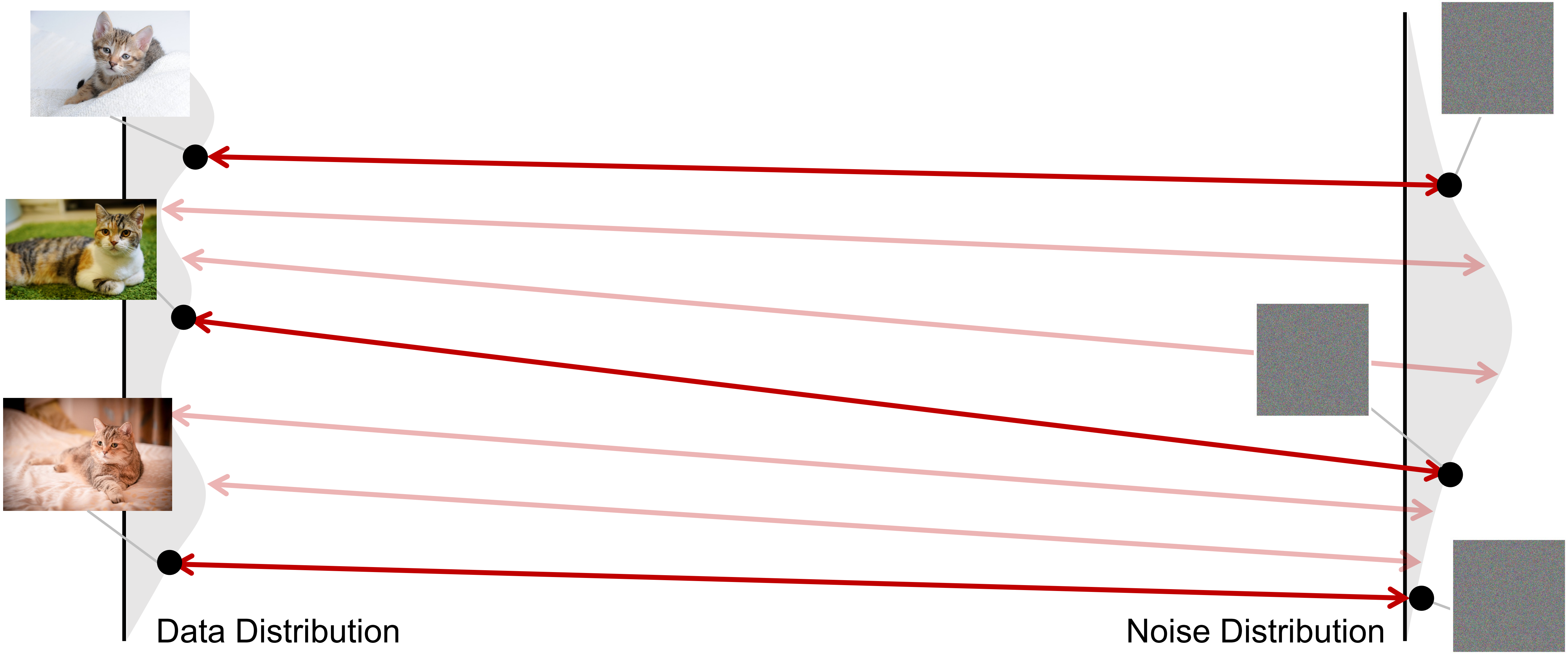
What is Generative Model Learning?

What is Generative Model Learning?

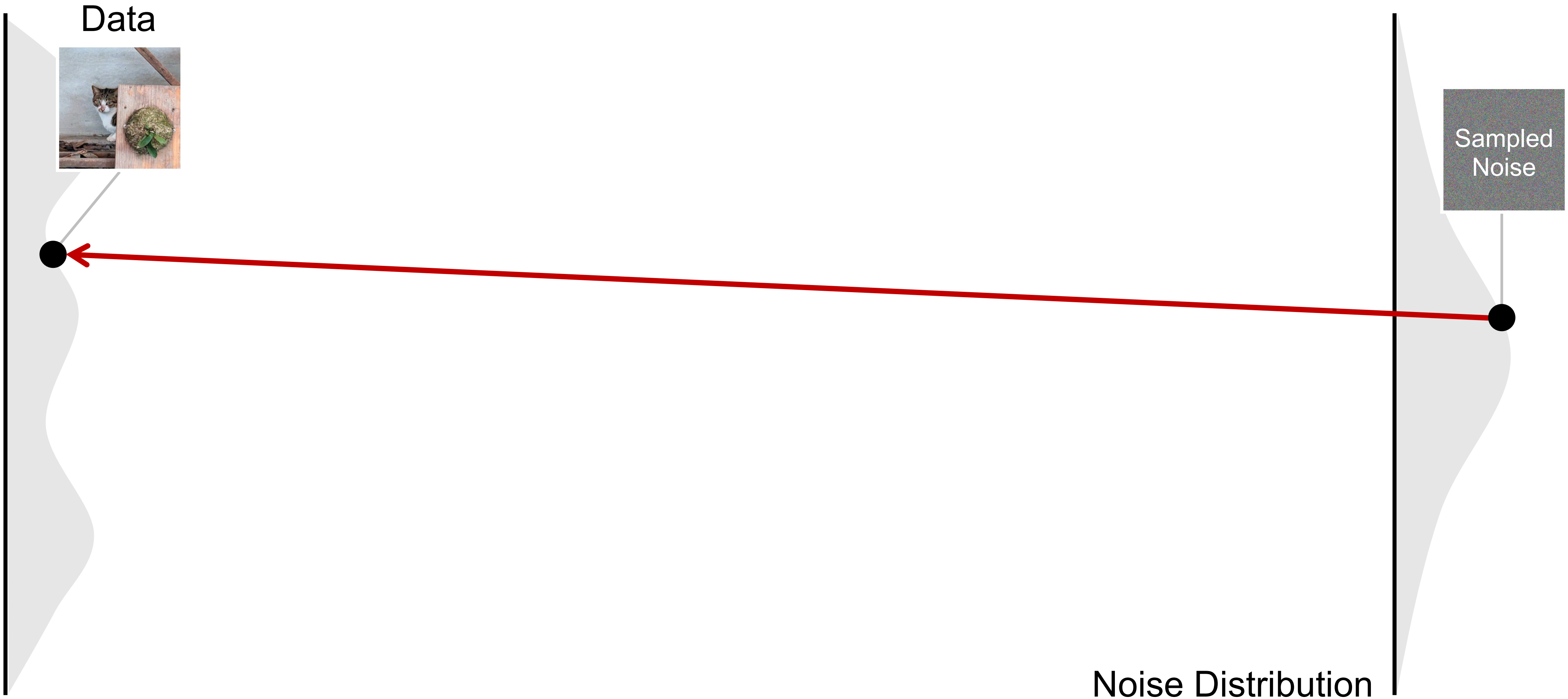


Data Distribution

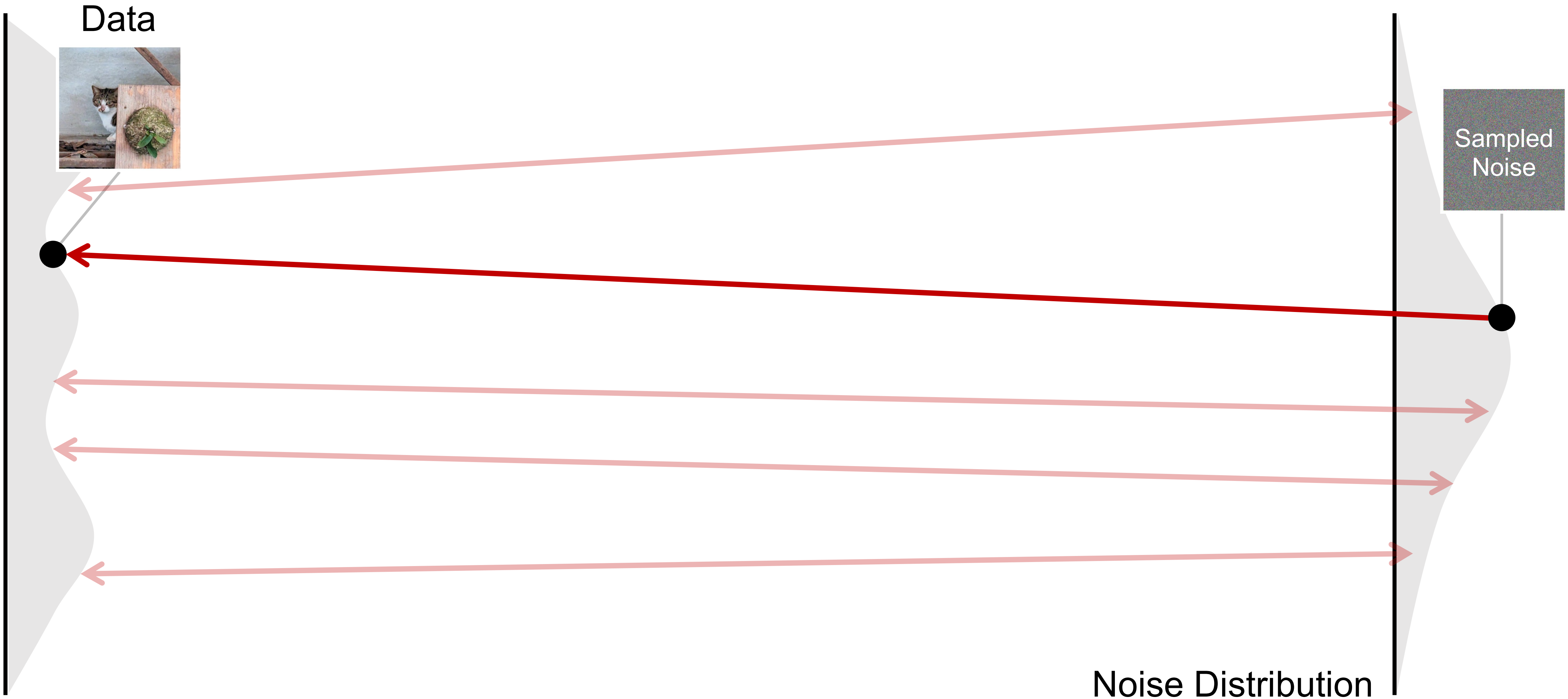
What is Generative Model Learning?



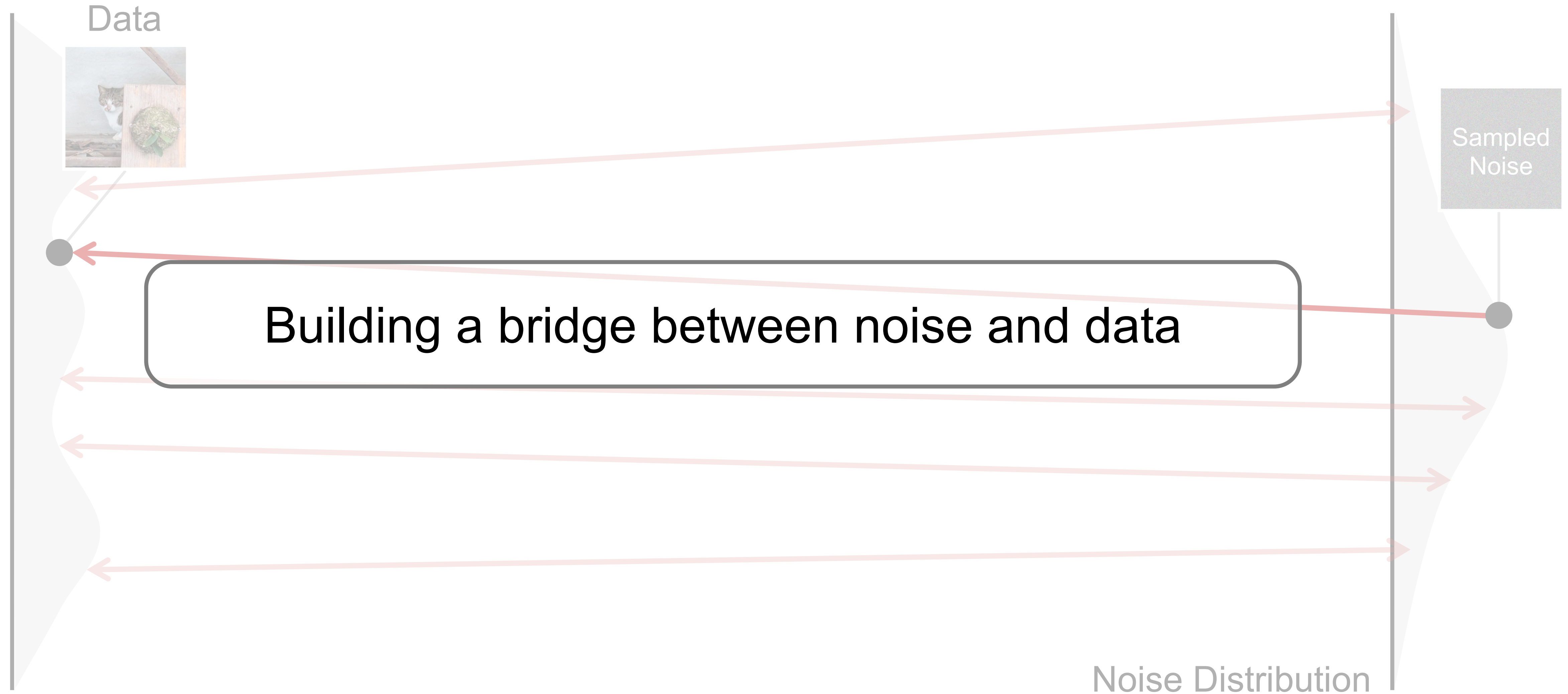
What is Generative Model Learning?



The Goal of Generative Model



The Goal of Generative Model

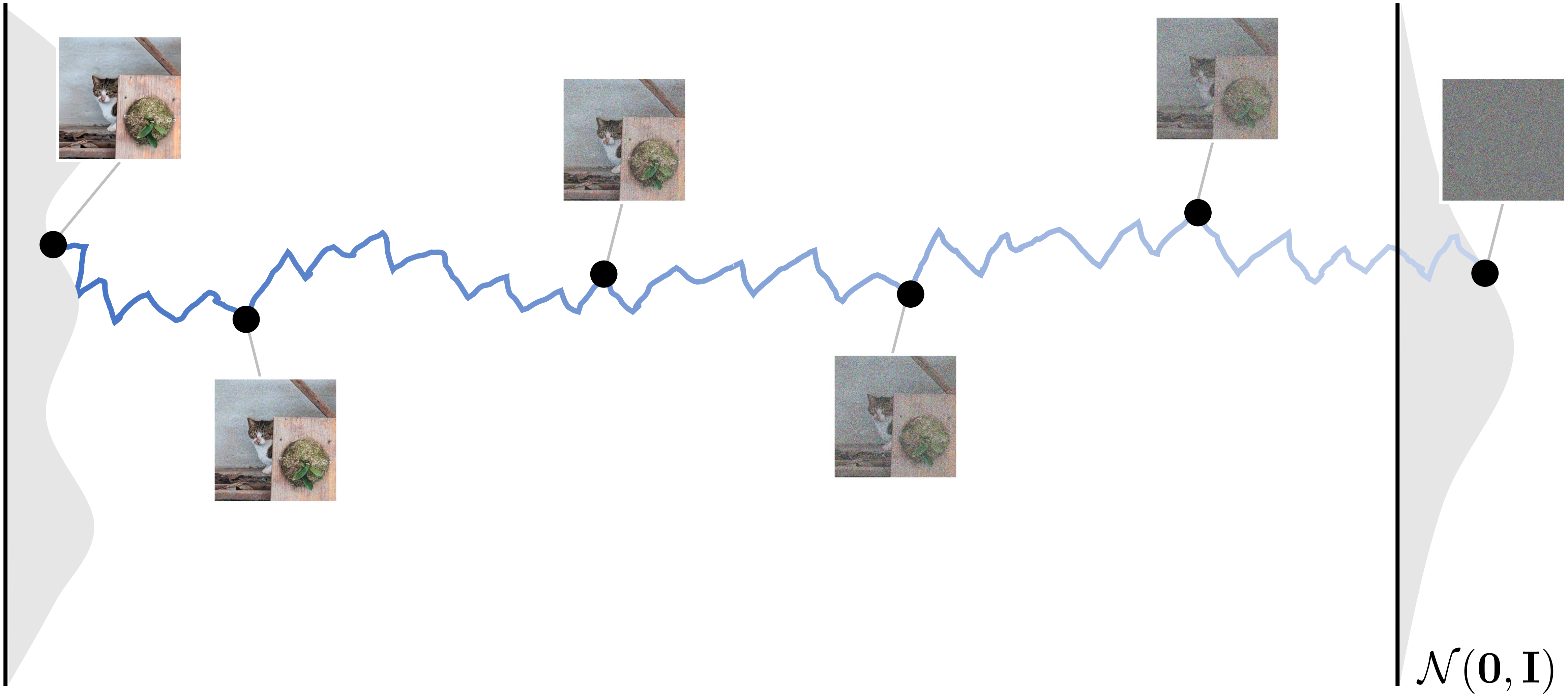


What is Diffusion Model?

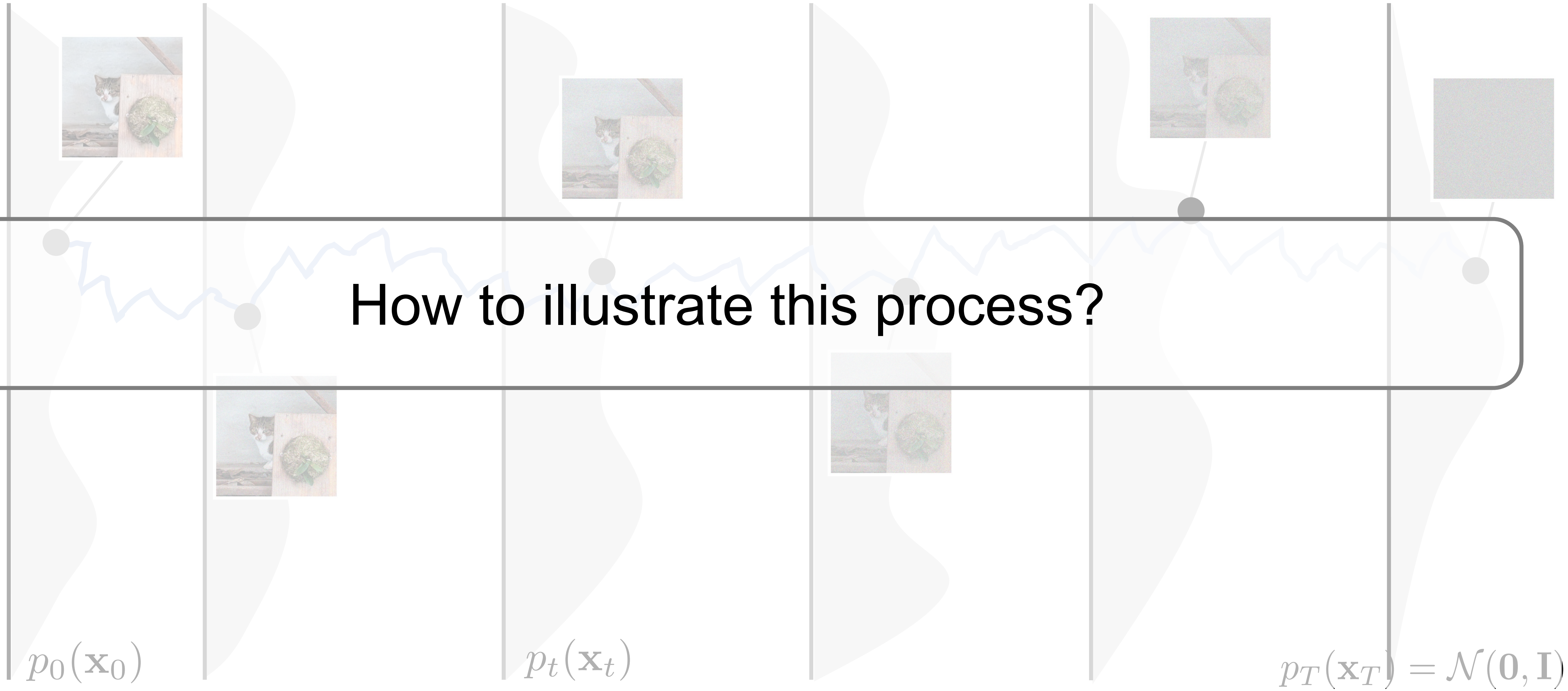
Forward Process: add noise step by step, from data to pure noise

Reverse Process: generate data from pure noise by denoising

Diffusion Models

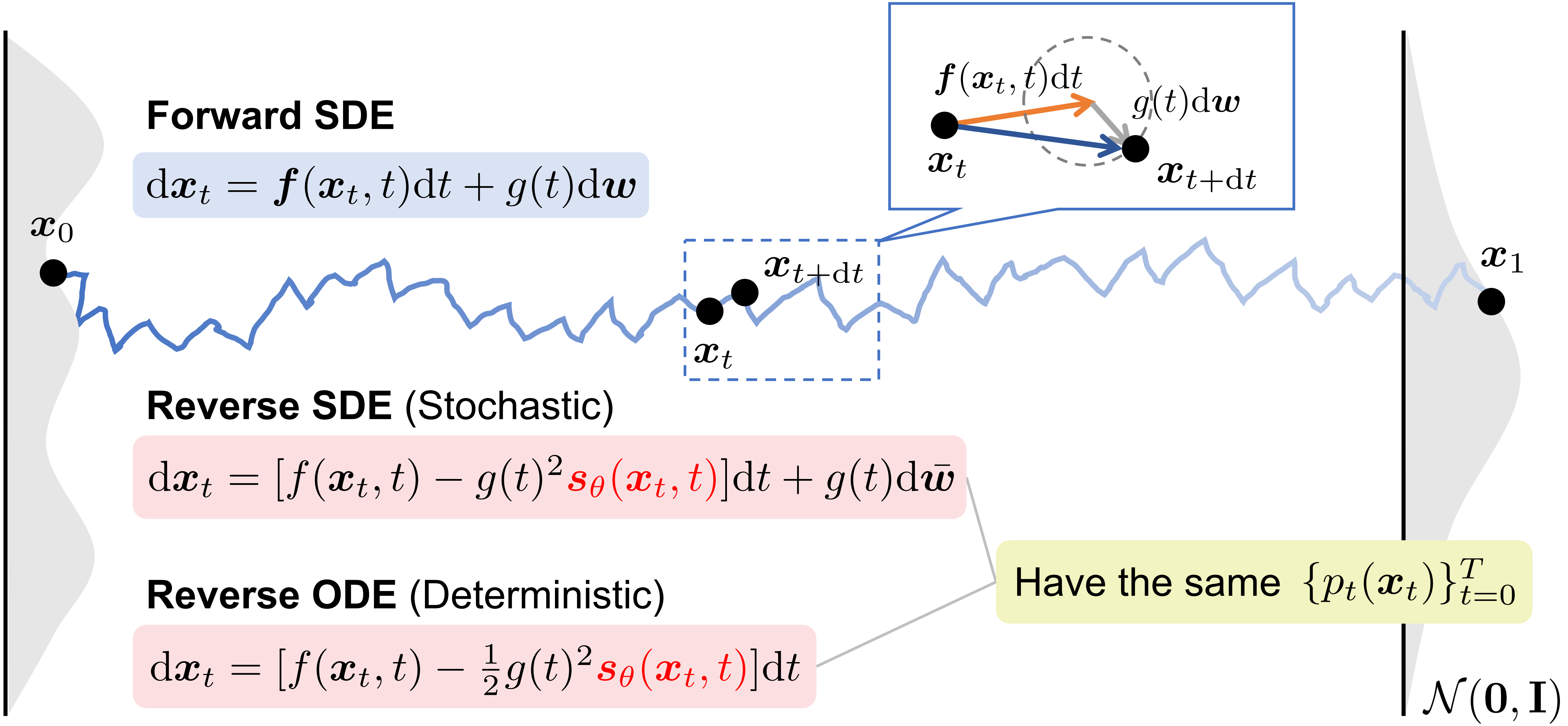


Diffusion Models



How to illustrate this process?

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



Forward SDE

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

Reverse SDE (Stochastic)

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

Reverse ODE (Deterministic)

$$\mathrm{d}\boldsymbol{x}_t = [f(\boldsymbol{x}_t, t) - \tfrac{1}{2} g(t)^2 \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)]\mathrm{d}t$$

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

Diffusion Model vs Flow Matching

- Score-based Diffusion Model

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2 \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$$

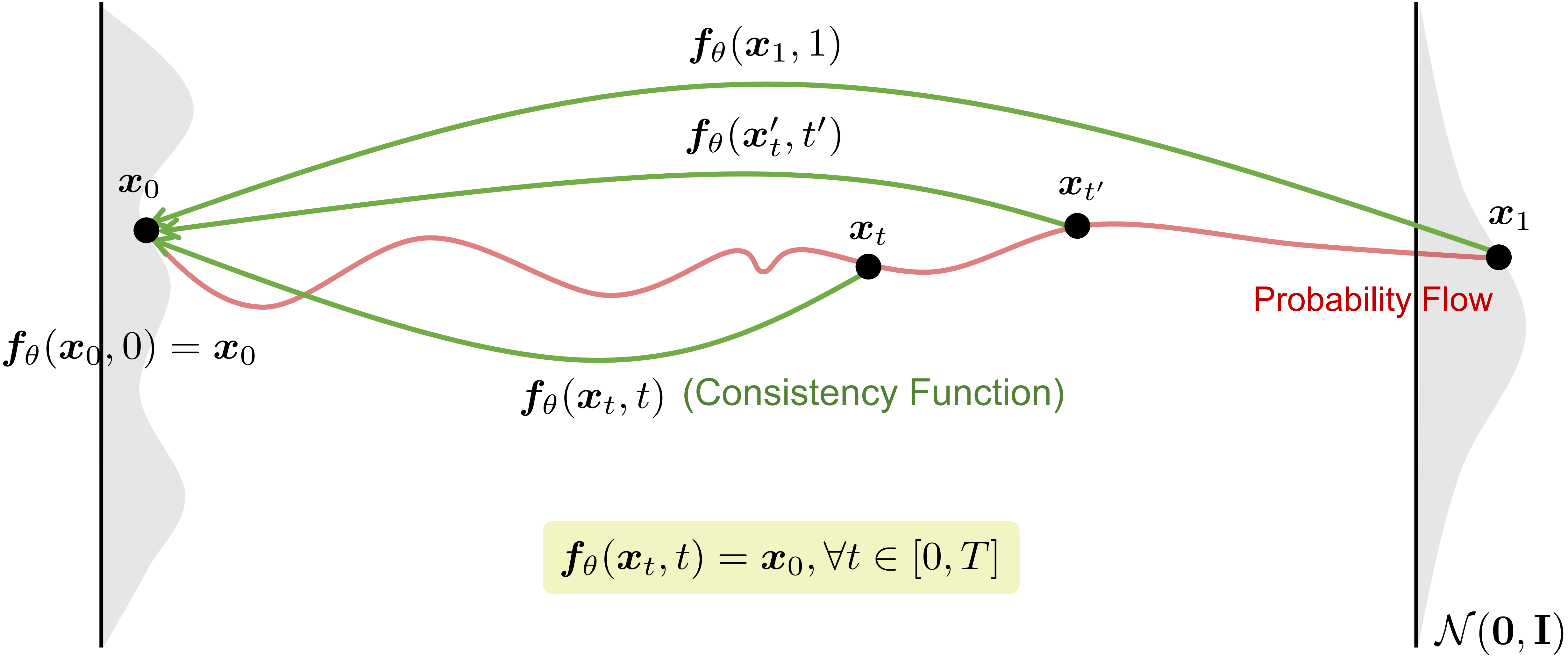
$$\mathcal{L}_{\mathrm{SM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t|\boldsymbol{x}_0} \|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \underset{\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}{\underset{\wr}{\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)}}\|_2^2$$

- Flow Matching

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

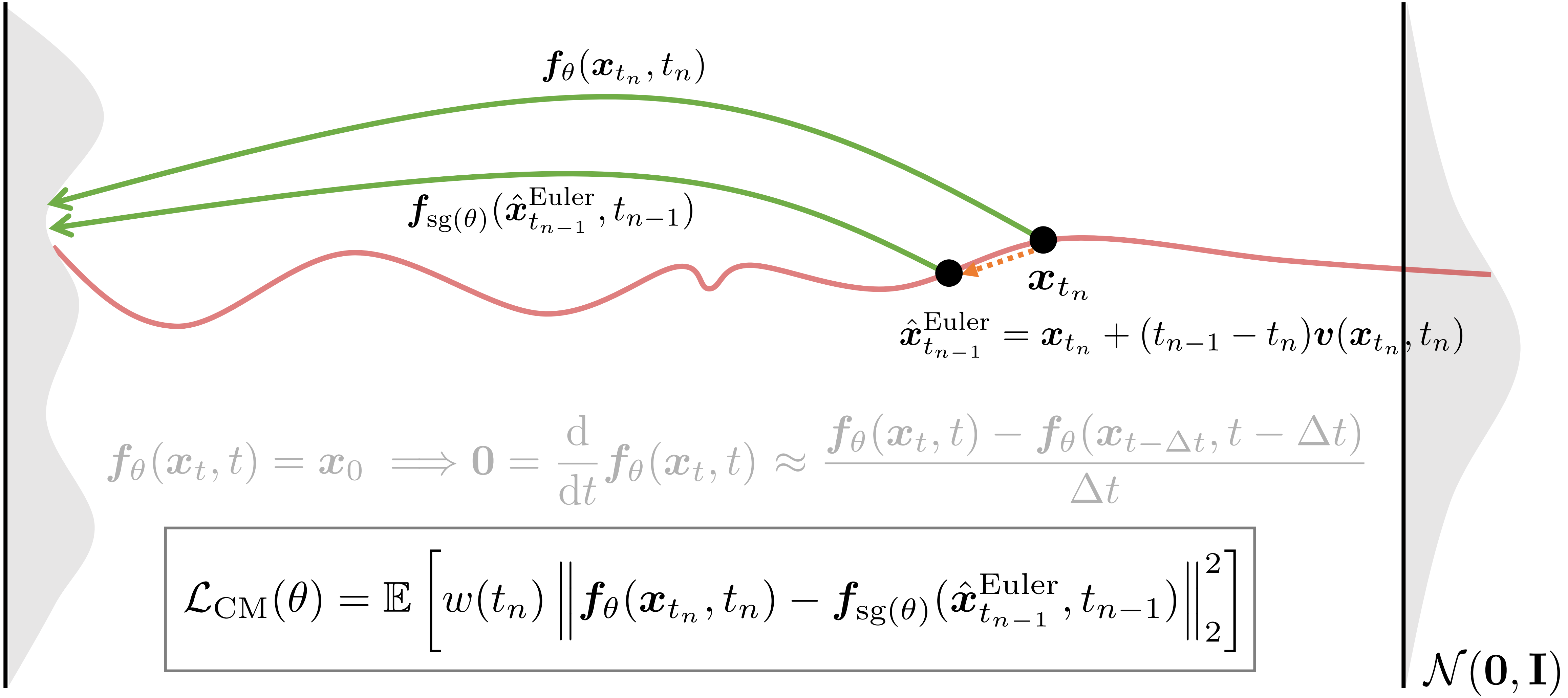
$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t|\boldsymbol{x}_0} \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_t\|_2^2$$

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



$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{x}_0, \forall t \in [0, T]$$

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



$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{x}_0 \implies \mathbf{0} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) \approx \frac{\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{f}_\theta(\boldsymbol{x}_{t-\Delta t}, t - \Delta t)}{\Delta t}$$

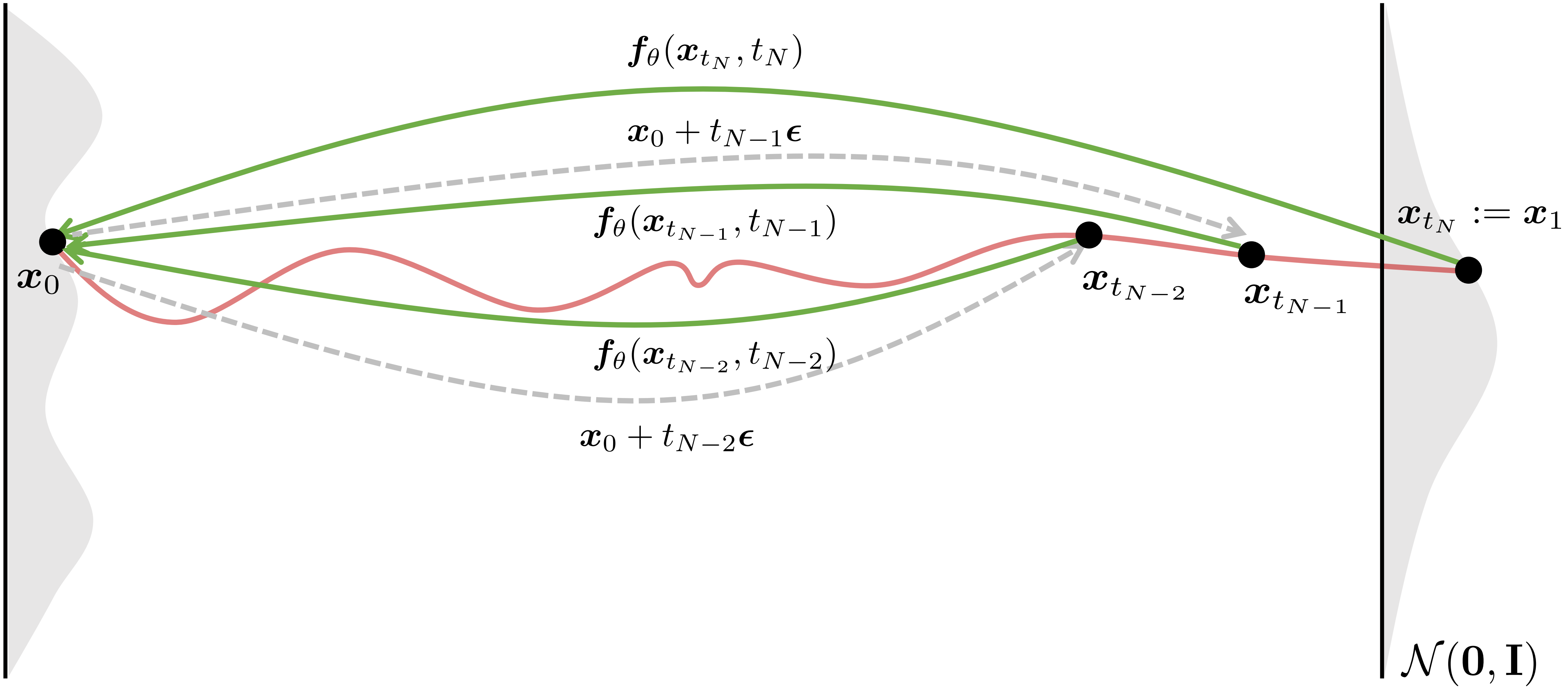
$$\boxed{\mathcal{L}_{\mathrm{CM}}(\theta) = \mathbb{E}\left[w(t_n)\left\|\boldsymbol{f}_\theta(\boldsymbol{x}_{t_n}, t_n) - \boldsymbol{f}_{\mathrm{sg}(\theta)}(\hat{\boldsymbol{x}}^{\mathrm{Euler}}_{t_{n-1}}, t_{n-1})\right\|_2^2\right]}$$

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

- **Consistency Distillation (CD)**
  - Pretrained Diffusion: $\boldsymbol{v}(\boldsymbol{x}_t, t) = -t\boldsymbol{s}_\phi(\boldsymbol{x}_t, t)$
  - Pretrained Flow: $\boldsymbol{v}(\boldsymbol{x}_t, t) = \boldsymbol{v}_\phi(\boldsymbol{x}_t, t)$

Slow Convergence !!

Sampling with CM



CM Experiments

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

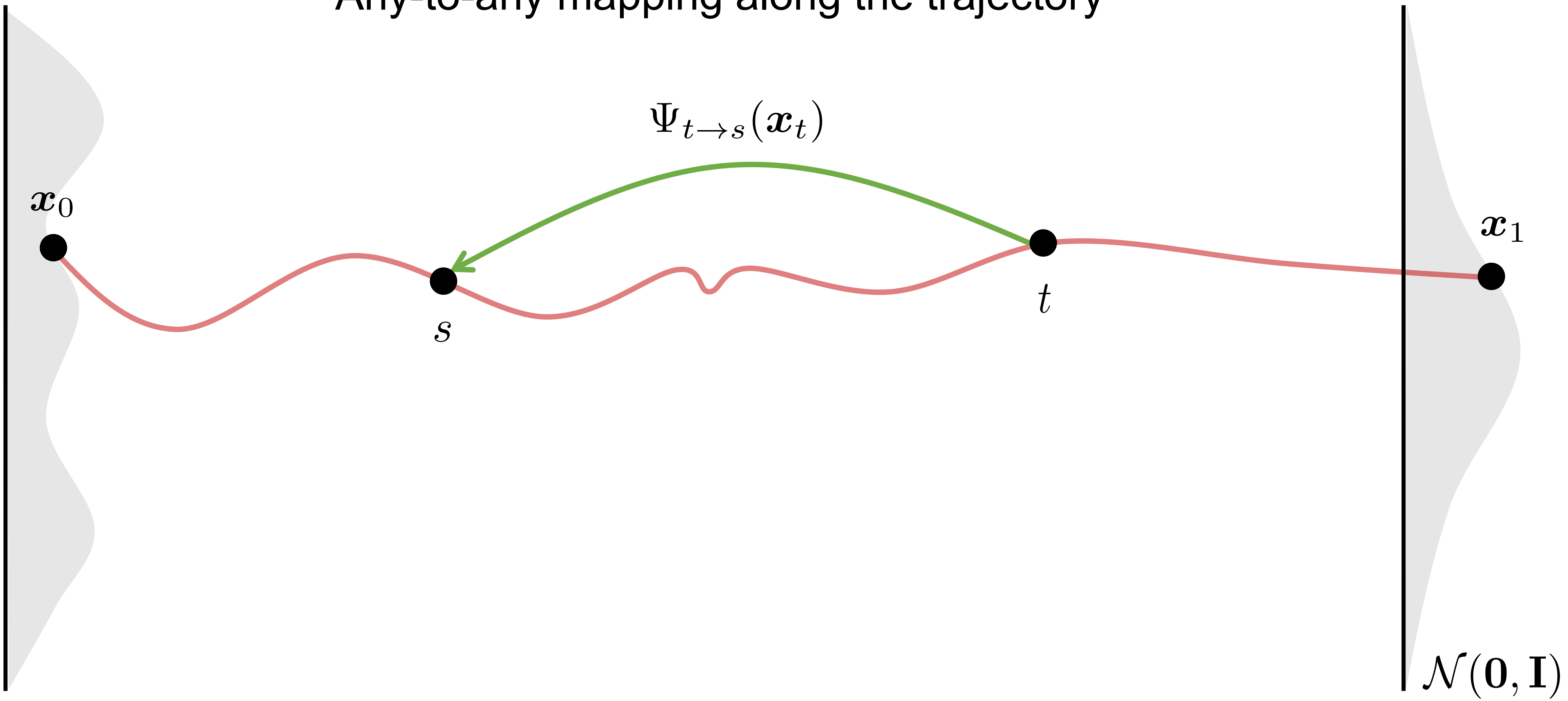
METHOD	NFE (↓)	FID (↓)	IS (↑)
Diffusion + Samplers			
DDIM (Song et al., 2020)	50	4.67	
DDIM (Song et al., 2020)	20	6.84	
DDIM (Song et al., 2020)	10	8.23	
DPM-solver-2 (Lu et al., 2022)	10	5.94	
DPM-solver-fast (Lu et al., 2022)	10	4.70	
3-DEIS (Zhang & Chen, 2022)	10	**4.17**	
Diffusion + Distillation			
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.

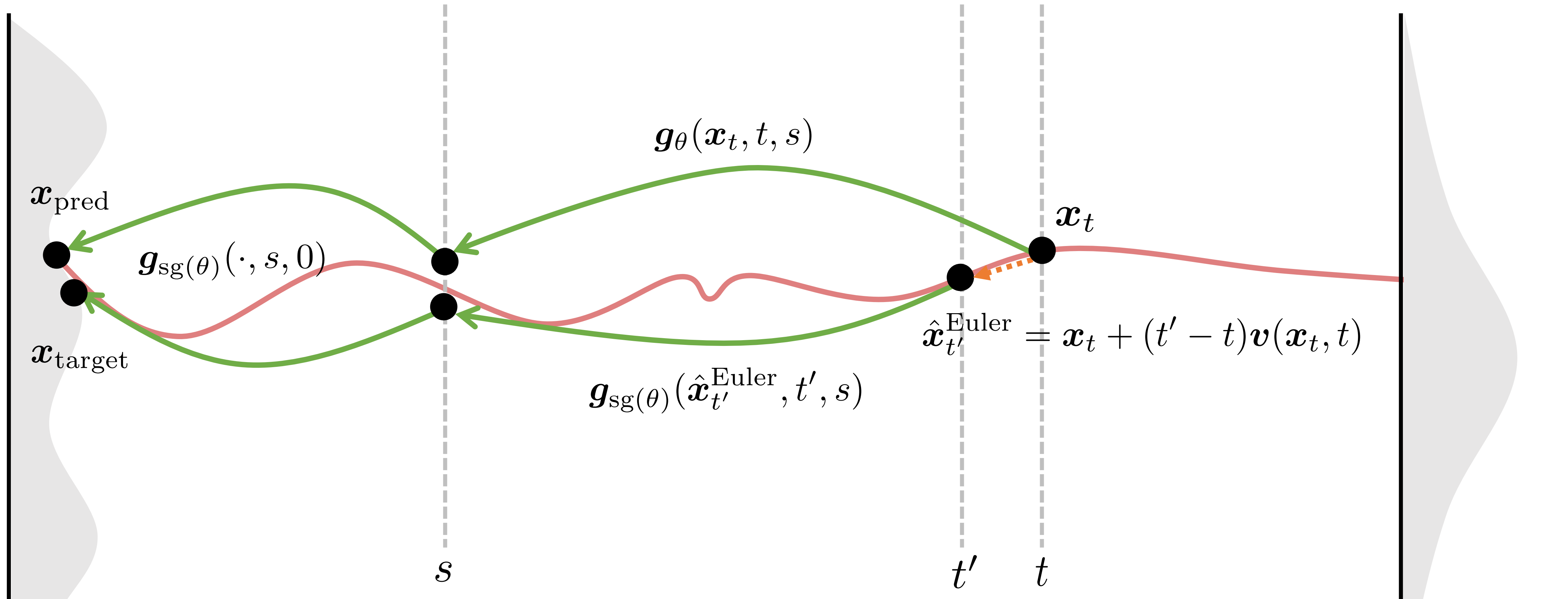
METHOD	NFE (↓)	FID (↓)	Prec. (↑)	Rec. (↑)
ImageNet 64 × 64				
PD† (Salimans & Ho, 2022)	1	15.39	0.59	0.62
DFNO† (Zheng et al., 2022)	1	8.35		
CD†	1	6.20	0.68	0.63
PD† (Salimans & Ho, 2022)	2	8.95	0.63	**0.65**
CD†	2	**4.70**	**0.69**	0.64
ADM (Dhariwal & Nichol, 2021)	250	**2.07**	0.74	0.63
EDM (Karras 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

Flow Map

Any-to-any mapping along the trajectory



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



$$\mathcal{L}_{\text{CTM}}(\theta) = \mathbb{E}\left[w(t_n)\left\|\boldsymbol{g}_{\text{sg}(\theta)}(\boldsymbol{g}_\theta(\boldsymbol{x}_t, t, s), s, 0) - \boldsymbol{g}_{\text{sg}(\theta)}(\boldsymbol{g}_{\text{sg}(\theta)}(\hat{\boldsymbol{x}}_{t'}^{\text{Euler}}, t', s), s, 0)\right\|_2^2\right]$$

CTM Losses

- **DM Loss:** When *t* an *s* are very close, the gradients from the CTM loss become weak, leading to slow learning. Incorporating the DM loss provides a stronger local training signal and stabilizes optimization.
- **GAN Loss:** CTM and DSM losses can yield overly smooth outputs; therefore, an adversarial term can be added to encourage sharper and more realistic samples by aligning the generator distribution with the data distribution.
- **Total Loss:** $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CTM}} + \lambda_{\text{DM}}\mathcal{L}_{\text{DM}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}$

CTM Experiments

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	[illegible]	-
Score SDE ([illegible])	[illegible]	[illegible]	[illegible]	[illegible]
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	-	-

Table 2: Performance comparisons on ImageNet 64 × 64.

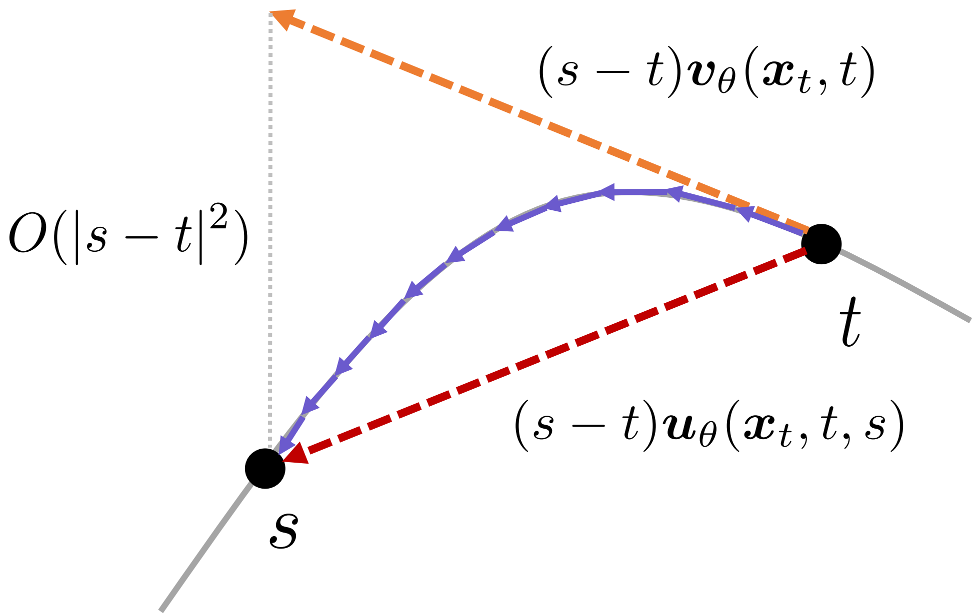
Model	NFE	FID↓	IS↑	Rec↑
Validation Data		1.41	64.10	0.67
ADM (Dhariwal & Nichol, 2021)	250	2.07	-	0.63
EDM (Karras et al., 2022)	79	2.44	48.88	**0.67**
BigGAN-deep (Brock et al., 2018)	1	4.06	-	0.48
StyleGAN-XL (Sauer et al., 2022)	1	2.09	**82.35**	0.52
Diffusion Models – Distillation Sampling				
PD (Salimans & Ho, 2021)	1	15.39	-	0.62
BOOT (Gu et al., 2023)	1	16.3	-	0.36
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
CTM (ours)	[illegible]	[illegible]	[illegible]	[illegible]
PD (Salimans & Ho, 2021)	2	8.95	-	0.65
CD (Song et al., 2023)	2	4.70	-	0.64
CTM (ours)	2	**1.73**	64.29	0.57

Is there a more efficient and simplified training strategy?

MeanFlow: Average Velocity

- What we want: $$\boldsymbol{x}_s = \boldsymbol{x}_t + \int_t^s \boldsymbol{v}_\theta(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau$$
- But we do: $$\boldsymbol{x}_s = \boldsymbol{x}_t + (s-t)\cancel{\boldsymbol{v}_\theta(\boldsymbol{x}_\tau, \tau) + O(|s-t|^2)}$$

$$\boldsymbol{u}_\theta(\boldsymbol{x}_t, t, s) \approx \frac{1}{s-t}\int_t^s \boldsymbol{v}_t(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau$$



MeanFlow Identity

Differential

$$\boxed{\boldsymbol{u}(\boldsymbol{x}_t, t, s) = \frac{1}{s-t}\int_t^s \boldsymbol{v}(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(s-t)\boldsymbol{u}(\boldsymbol{x}_t, t, s) = \frac{\mathrm{d}}{\mathrm{d}t}\int_t^s \boldsymbol{v}(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau$$

$$-\boldsymbol{u}(\boldsymbol{x}_t, t, s) + (s-t)\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(\boldsymbol{x}_t, t, s) = -\boldsymbol{v}(\boldsymbol{x}_t, t)$$

$$\boxed{\boldsymbol{u}(\boldsymbol{x}_t, t, s) = \boldsymbol{v}(\boldsymbol{x}_t, t) + (s-t)\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(\boldsymbol{x}_t, t, s)}$$

Integral

MeanFlow: Time Derivative

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(\boldsymbol{x}_t, t, s) = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}_t} \cdot \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \frac{\partial \boldsymbol{u}}{\partial t} \cdot \frac{\mathrm{d}t}{\mathrm{d}t} + \frac{\partial \boldsymbol{u}}{\partial s} \cdot \frac{\mathrm{d}s}{\mathrm{d}t}$$

$$= \boldsymbol{v}_t(\boldsymbol{x}_t, t)\partial_{\boldsymbol{x}_t}\boldsymbol{u} + \partial_t \boldsymbol{u}$$

$$= \left[\frac{\partial \boldsymbol{u}(\boldsymbol{x}_t, t, s)}{\partial(\boldsymbol{x}_t, t, s)}\right]\begin{bmatrix}\boldsymbol{v}_t(\boldsymbol{x}_t, t) & 1 & 0\end{bmatrix}^\top$$ (Jacobian-Vector Product)

$$\boldsymbol{u}(\boldsymbol{x}_t, t, s) = \boldsymbol{v}_t(\boldsymbol{x}_t, t) + (s - t)(\boldsymbol{v}(\boldsymbol{x}_t, t)\partial_{\boldsymbol{x}_t}\boldsymbol{u} + \partial_t \boldsymbol{u})$$

MeanFlow: Training Objective

$$\mathcal{L}_{\mathrm{MF}}(\theta) = \|\boldsymbol{u}_\theta(\boldsymbol{x}_t, t, s) - \boldsymbol{u}_{\mathrm{target}}\|_2^2$$

$$\boldsymbol{u}_{\mathrm{target}}(\boldsymbol{x}_t, t, s) = \boldsymbol{v}_t(\boldsymbol{x}_t, t) + (s - t)(\boldsymbol{v}(\boldsymbol{x}_t, t)\partial_{\boldsymbol{x}_t}\boldsymbol{u}_{\mathrm{sg}(\theta)} + \partial_t \boldsymbol{u}_{\mathrm{sg}(\theta)})$$

MeanFlow: Sampling

- Multi-step Sampling

$$\boldsymbol{x}_{t_i} = \boldsymbol{x}_{t_{i+1}} + (t_i - t_{i+1})\boldsymbol{u}_\theta(\boldsymbol{x}_{t_{i+1}}, t_{i+1}, t_i)$$

- One-step Sampling

$$\boldsymbol{x}_0 = \boldsymbol{x}_1 + \boldsymbol{u}_\theta(\boldsymbol{x}_1, 1, 0)$$

MeanFlow Experiments

Result on ImageNet-256 x 256

method	params	NFE	FID
1-NFE diffusion/flow from scratch			
iCT-XL/2 [43]†	675M	1	34.24
Shortcut-XL/2 [13]	675M	1	10.60
MeanFlow-B/2	131M	1	6.17
MeanFlow-M/2	308M	1	5.01
MeanFlow-L/2	459M	1	3.84
MeanFlow-XL/2	676M	1	**3.43**
2-NFE diffusion/flow from scratch			
iCT-XL/2 [43]†	675M	2	20.30
iMM-XL/2 [52]	675M	1×2	7.77
MeanFlow-XL/2	676M	2	2.93
MeanFlow-XL/2+	676M	2	**2.20**

method	params	NFE	FID
GANs			
BigGAN [5]	112M	1	6.95
GigaGAN [21]	569M	1	3.45
StyleGAN-XL [40]	166M	1	2.30
autoregressive/masking			
AR w/ VQGAN [10]	227M	1024	26.52
MaskGIT [6]	227M	8	6.18
VAR-*d*30 [47]	2B	10×2	1.92
MAR-H [27]	943M	256×2	1.55
diffusion/flow			
ADM [8]	554M	250×2	10.94
LDM-4-G [37]	400M	250×2	3.60
SimDiff [20]	2B	512×2	2.77
DiT-XL/2 [34]	675M	250×2	2.27
SiT-XL/2 [33]	675M	250×2	2.06
SiT-XL/2+REPA [51]	675M	250×2	**1.42**

Summary

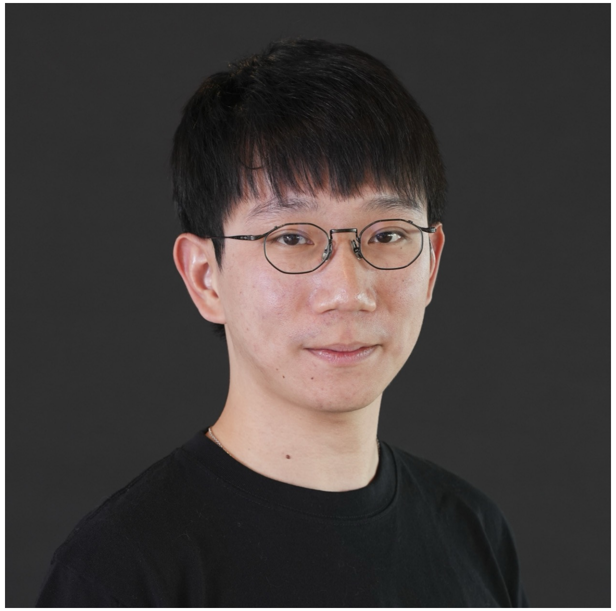
- Consistency Models formulate diffusion in a one-step manner and support both distillation and direct training.
- Consistency Trajectory Models extend Consistency Models by learning the flow map along the trajectory and introducing multiple loss terms to improve generation quality.
- MeanFlow introduces integral-based velocity averaging to further improve sample quality and stability, moving us closer to fast and high-fidelity generative models.

Further Follow-up

- **Consistency Models**
  - Consistency Model (CM)
  - Improved Consistency Training (iCT)
  - Easy Consistency Model (ECM)
  - Simple/stable/scalable Consistency Model (sCM)
- **Flow Maps Models**
  - Consistency Trajectory Model (CTM)
  - Shortcut Model
  - MeanFlow (MF)
  - Improved MeanFlow (iMF)
  - Consistency Mid-Training (CMT)

Recommended Reading

Some concepts and insights in these slides are inspired by Jesse



Chieh-Hsin (Jesse) Lai

Research Scientist at Sony AI

The Principles of Diffusion Models

From Origins to Advances

Chieh-Hsin Lai
Sony AI

Yang Song
OpenAI

Dongjun Kim
Stanford University

Yuki Mitsufuji
Sony Corporation, Sony AI

Stefano Ermon
Stanford University

Thank you