Consistency Models

Yang Song

Prafulla Dhariwal

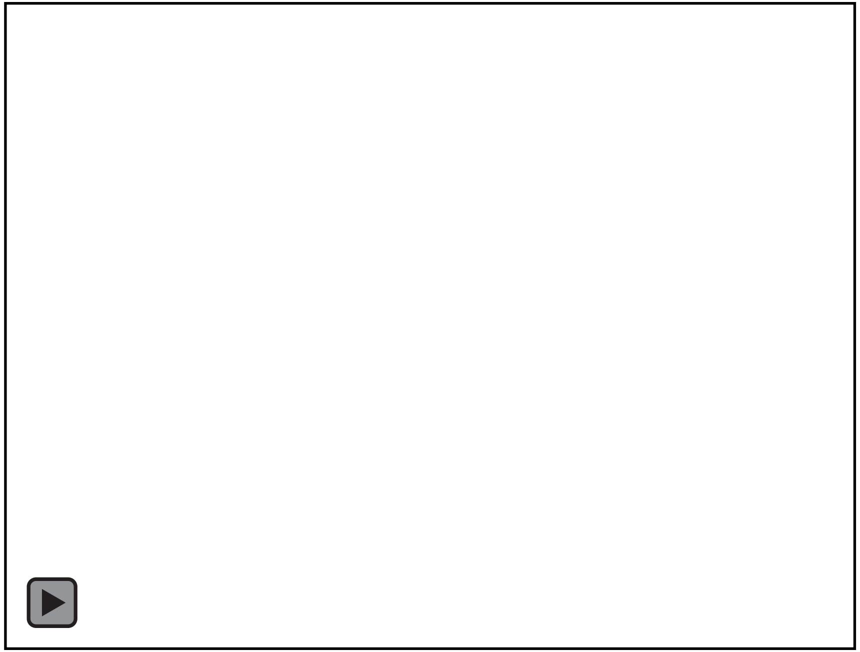
Mark Chen

Ilya Sutskever

Cheng Lu

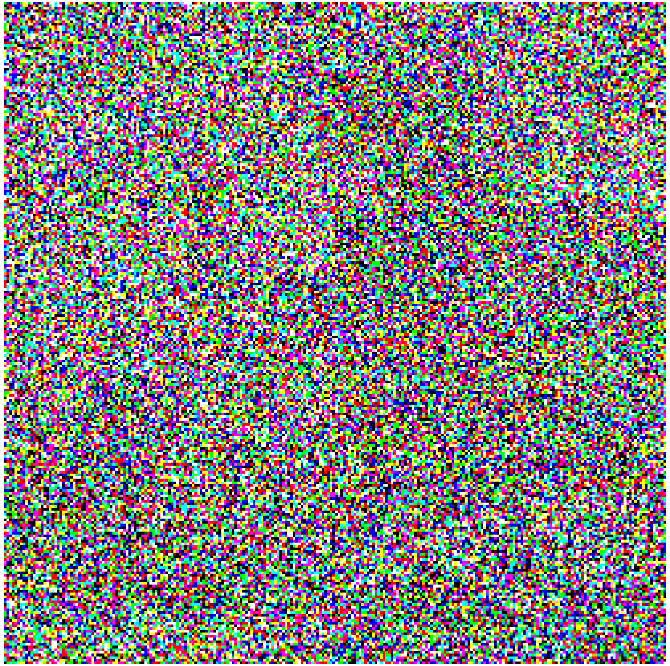
The success of diffusion models

OpenAI DALL·E 3



OpenAI Sora

Diffusion sampling is slow



- At least 10 steps for generating reasonable images.
- For best quality, often needs thousands of sampling steps.

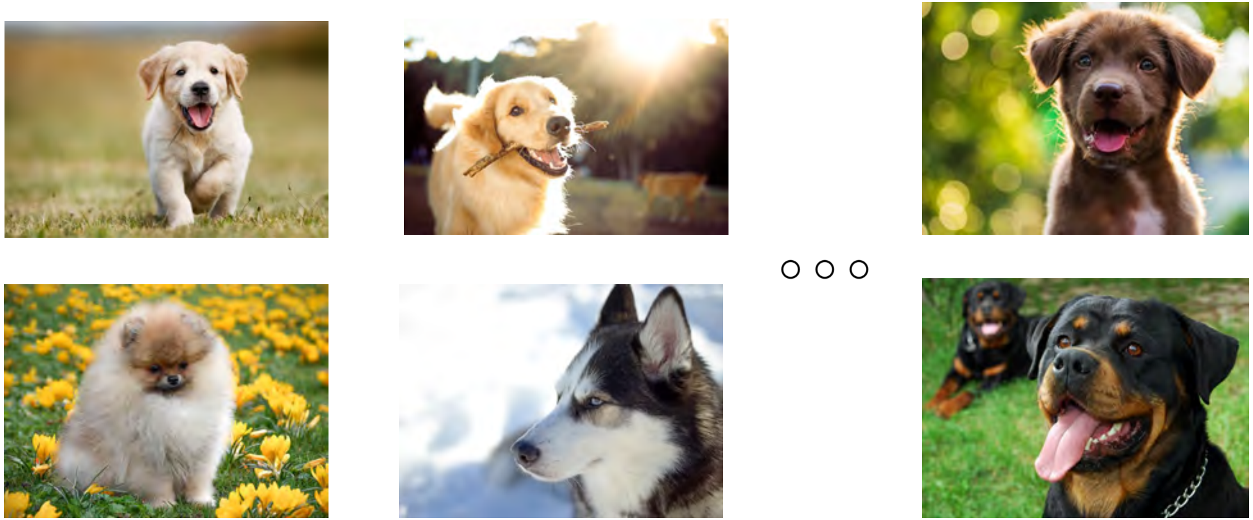
How to tackle this fundamental challenge in sampling speed?

Consistency models

Background: Continuous-time Diffusion Models

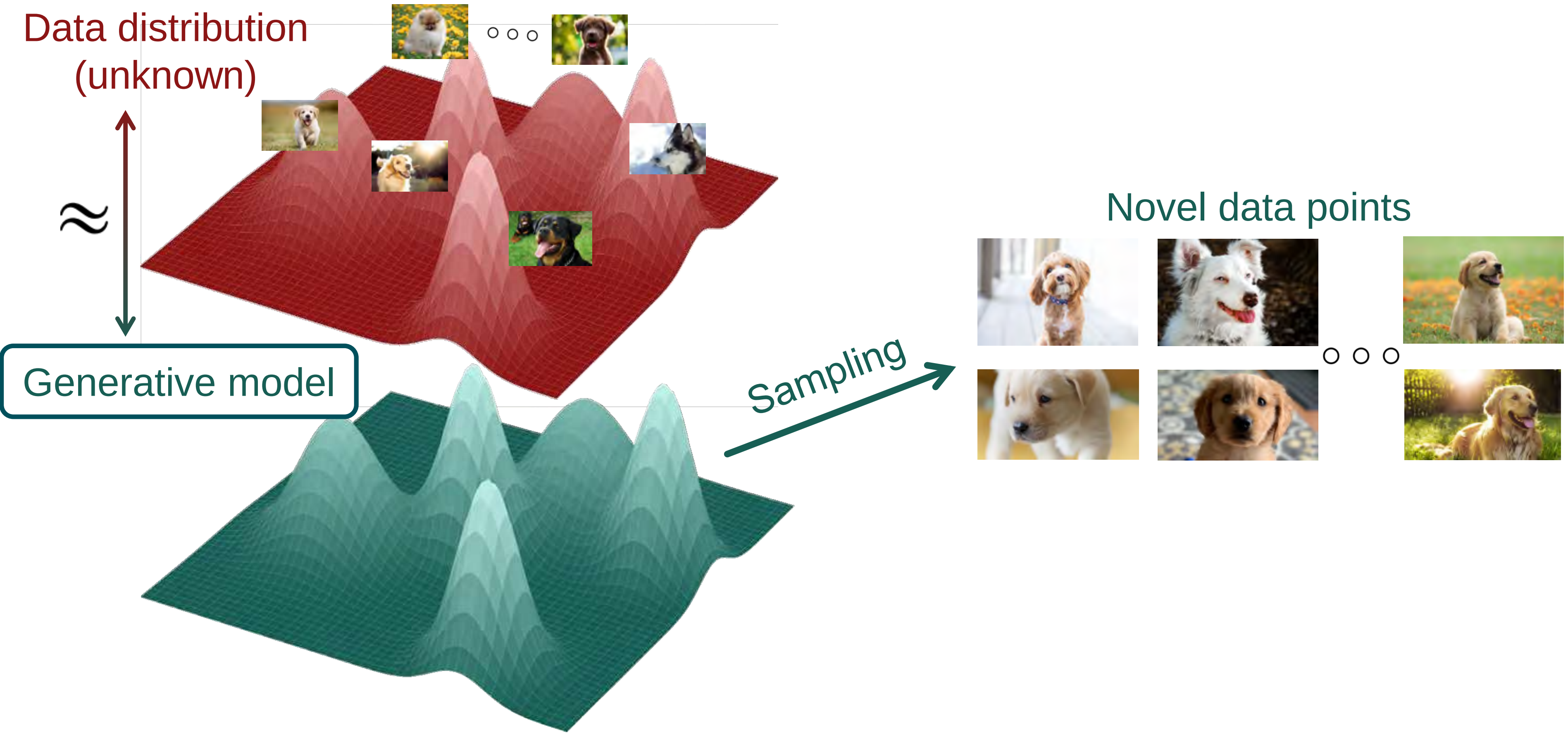
Song, et al. Score-Based Generative Modeling through Stochastic Differential Equations. ICLR 2021

Estimating the probability distribution of data

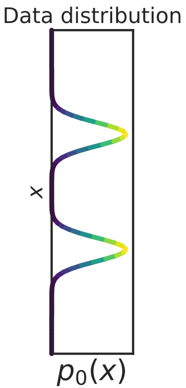


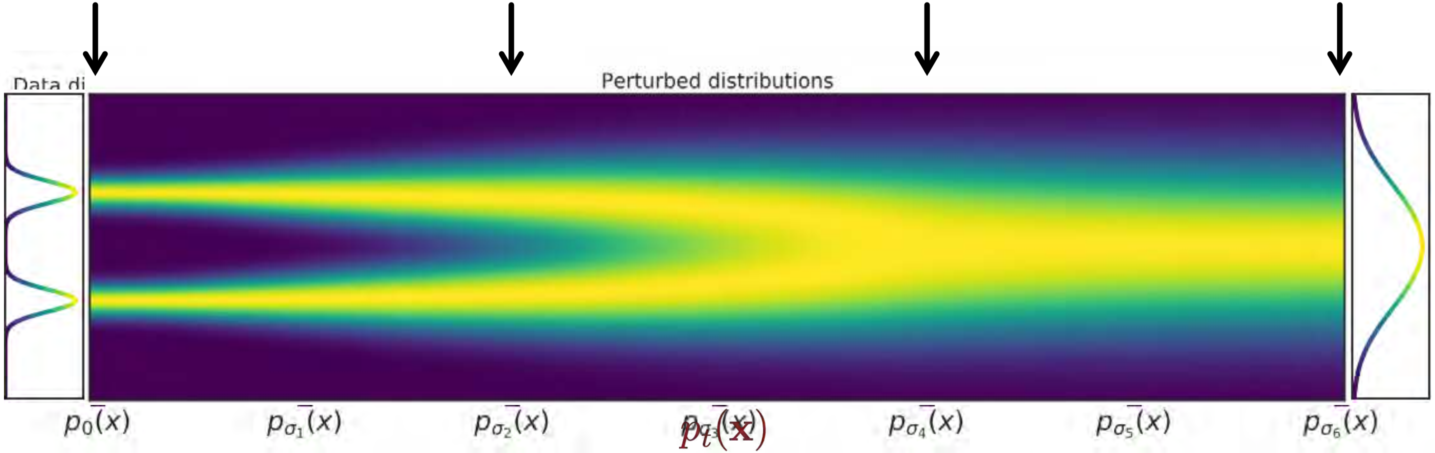
Data samples

Estimating the probability distribution of data



Deforming data distribution to Gaussian





$$p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$$

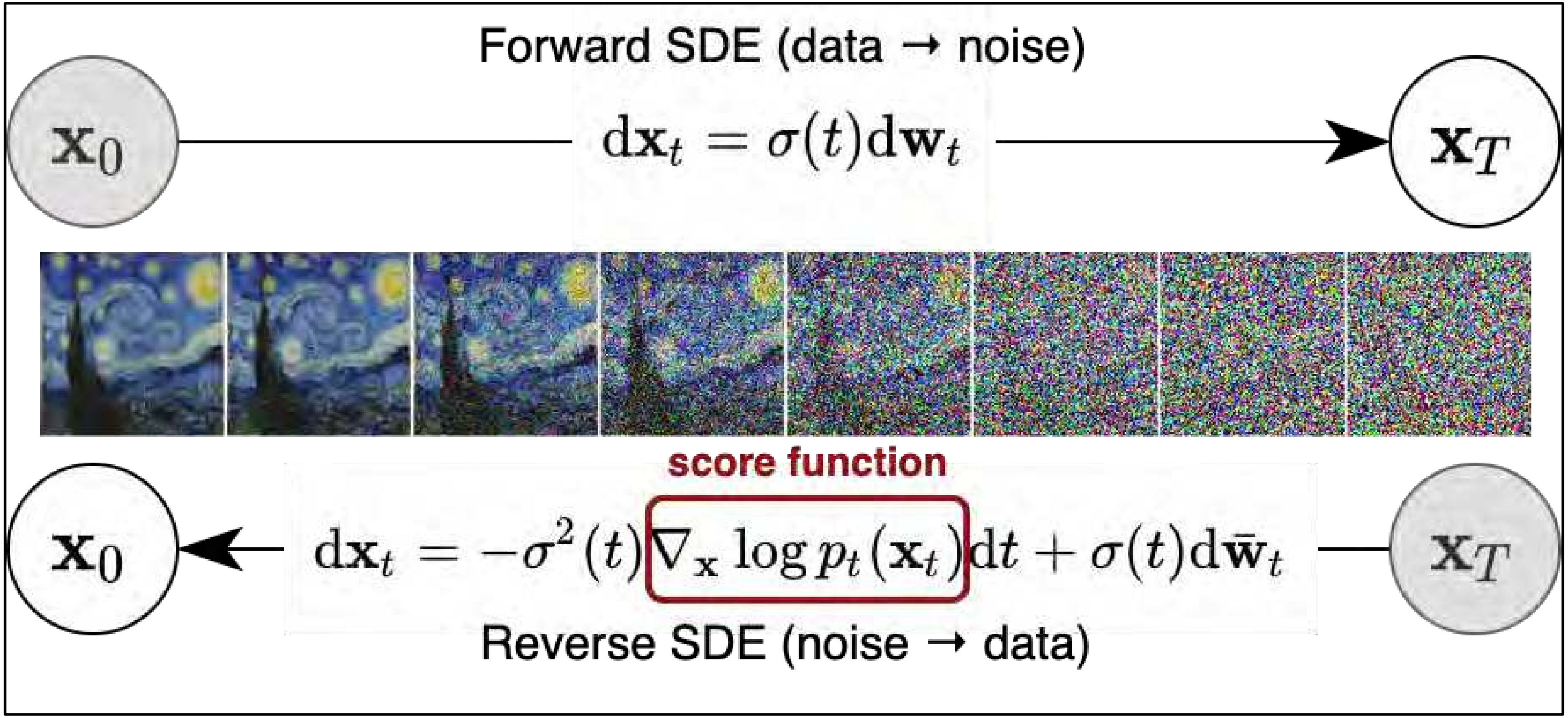
$$p_T(\mathbf{x}) \approx \pi(\mathbf{x})$$

$$p_t(\mathbf{x})$$

$$t \in [0, T]$$

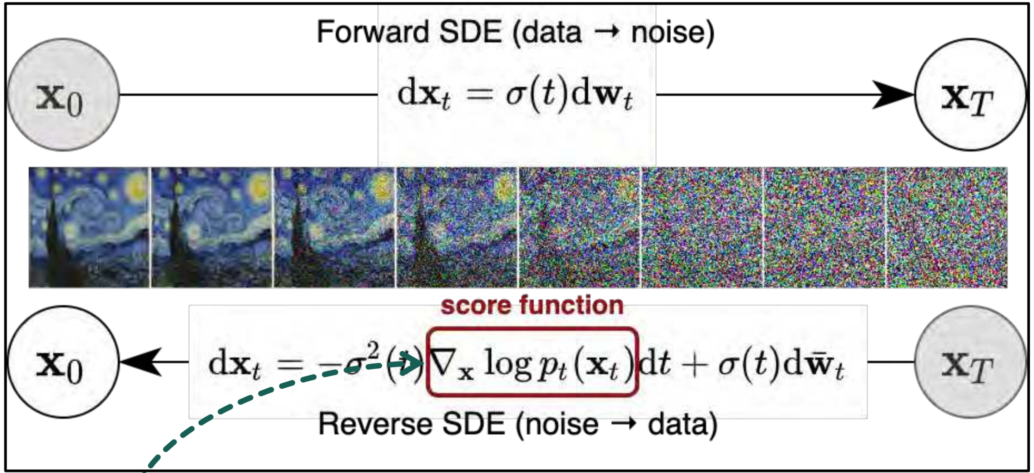
[illegible]

Score-based generative modeling via SDEs



[**Song** et al. ICLR 2021]

Score-based generative modeling via SDEs



Time conditional score model

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

Training objective:

$$\mathbb{E}_{t\sim \mathrm{Uniform}[0,T]}\Big[\underbrace{\lambda(t)}_{\text{Positive weighting function}}\underbrace{\mathbb{E}_{p_t(\mathbf{x})}\big[\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x}, t)\|_2^2\big]}_{\text{Score matching loss}}\Big]$$

[**Song** et al. ICLR 2021]

High-fidelity generation of 1024x1024 images

[**Song** et al. ICLR 2021]

Converting the SDE to an ODE

SDE

$$\mathrm{d}\mathbf{x}_t = \sigma(t)\,\mathrm{d}\mathbf{w}_t$$

Ordinary differential equation (probability flow ODE)

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = -\frac{1}{2}\sigma(t)^2 \boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)}$$

Score function

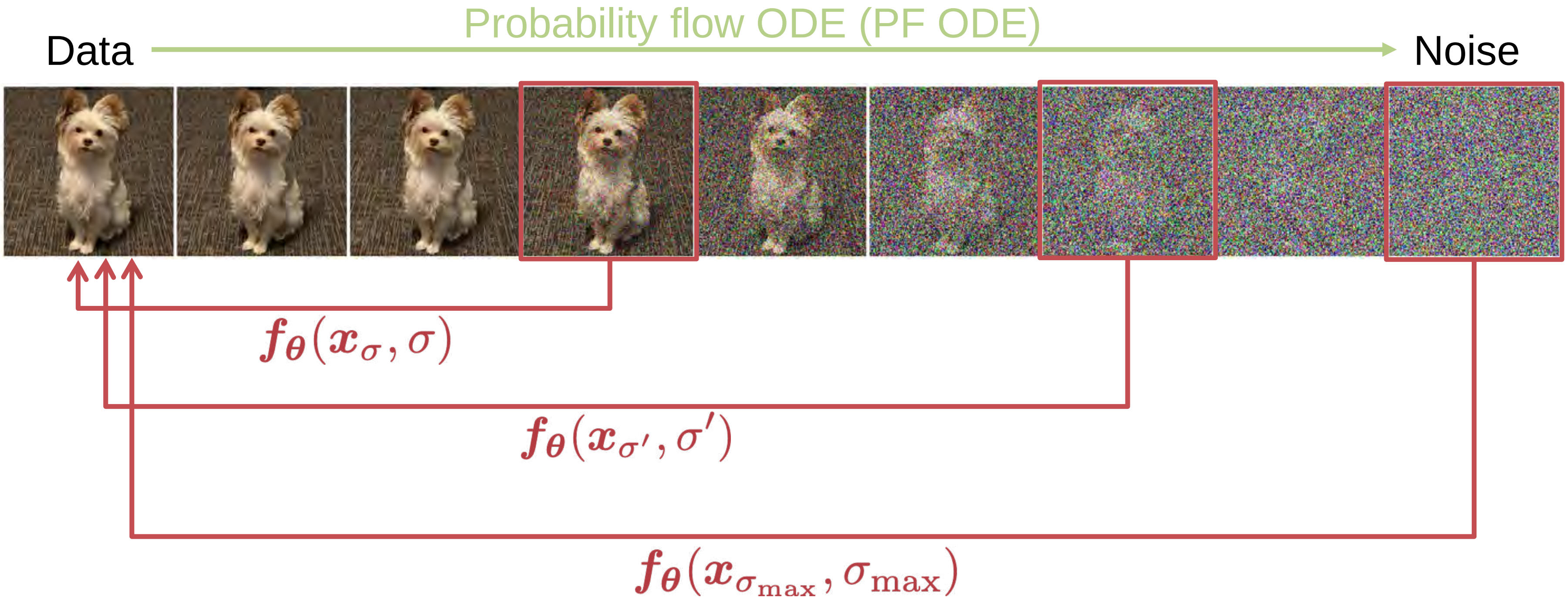
$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$$

[**Song** et al. ICLR 2021]

Basics of Consistency Models

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Consistency models are designed for one-step generation

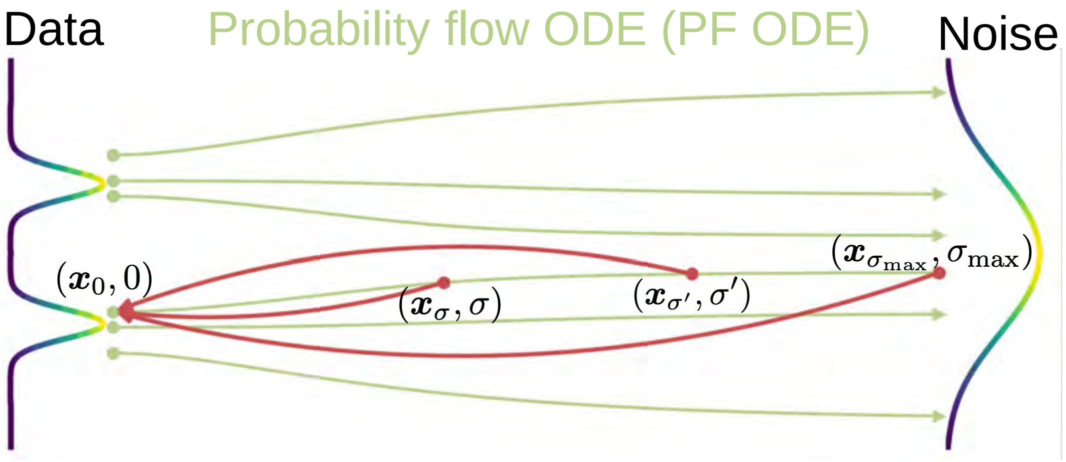


?

How does this differ from a denoiser?

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Consistency models learn this one-to-one mapping



Consistency models are trained to map points on any trajectory of the PF ODE to the trajectory's origin **in one step**.

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma, \sigma) = \boldsymbol{x}_0$$

Boundary condition

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_0, 0) = \mathbf{x}_0$$

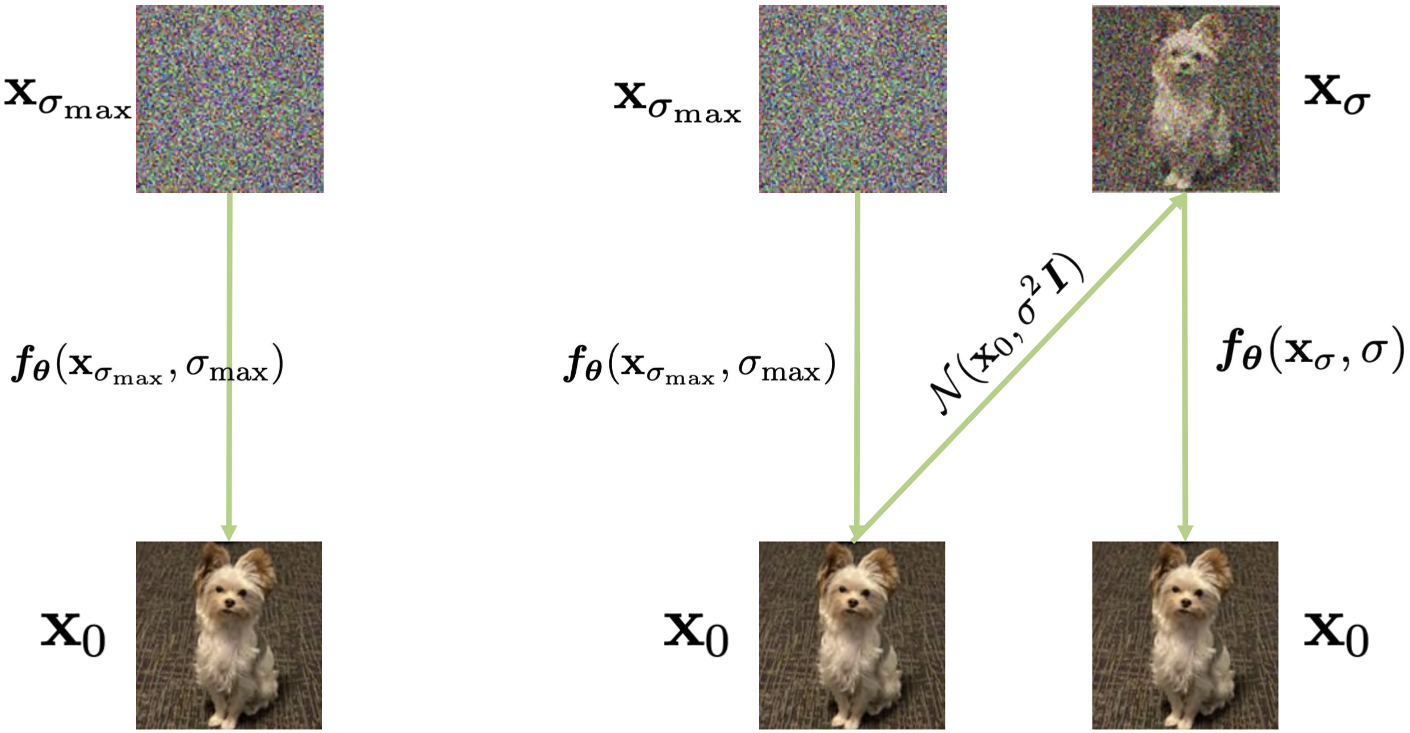
Enforced via network parameterization

Enforced via learning

Self-consistency

$$\forall \sigma, \sigma' \in [0, \sigma_{\max}] : \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma, \sigma) = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma'}, \sigma')$$

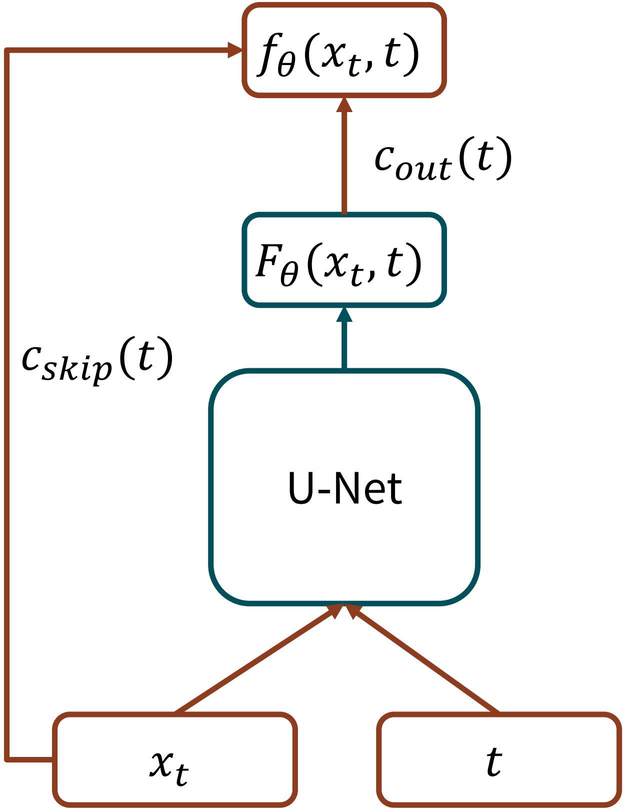
Sampling from consistency models



- Trading compute for quality
- Zero-shot image editing

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Enforcing the boundary condition



- Skip connections for enforcing the boundary condition:

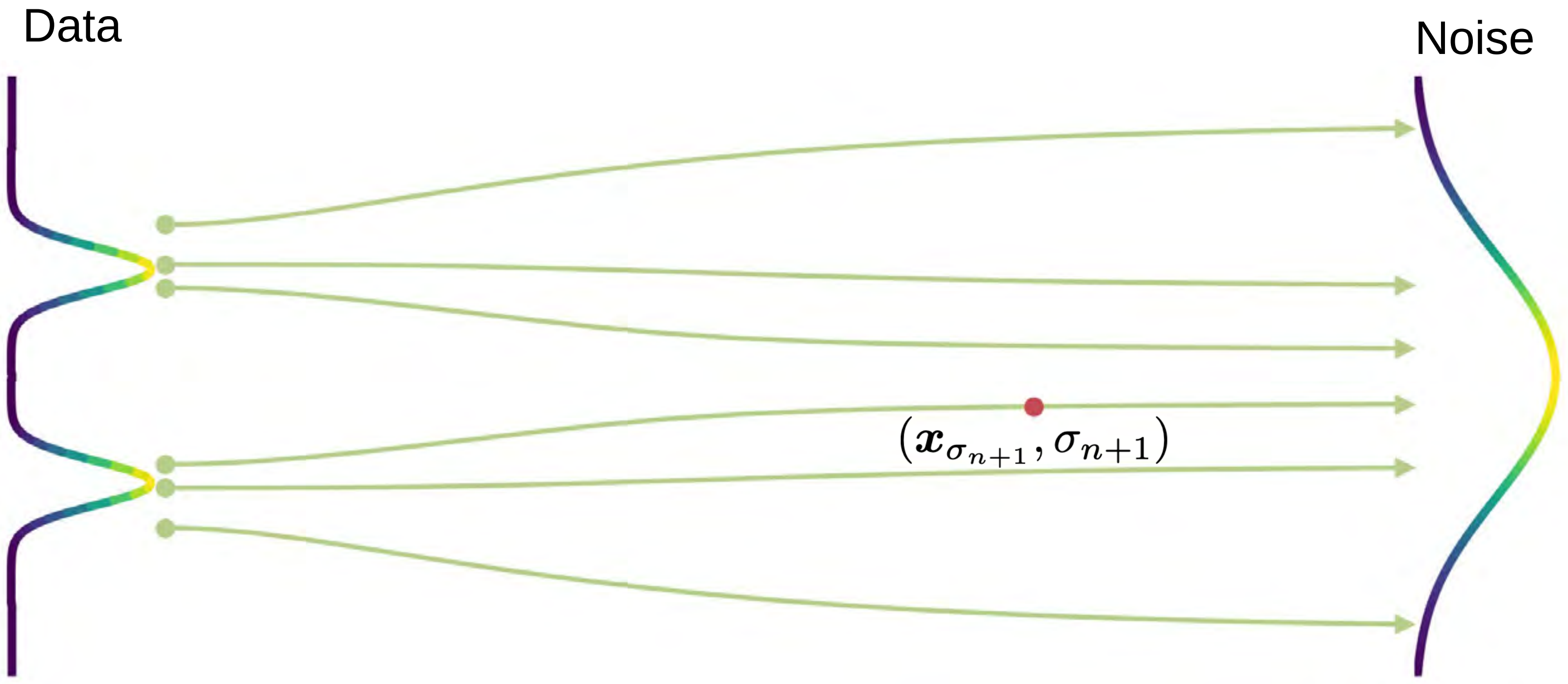
$$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)F_{\boldsymbol{\theta}}(\mathbf{x}, t)$$

$$c_{\text{skip}}(0) = 1$$
$$c_{\text{out}}(0) = 0$$

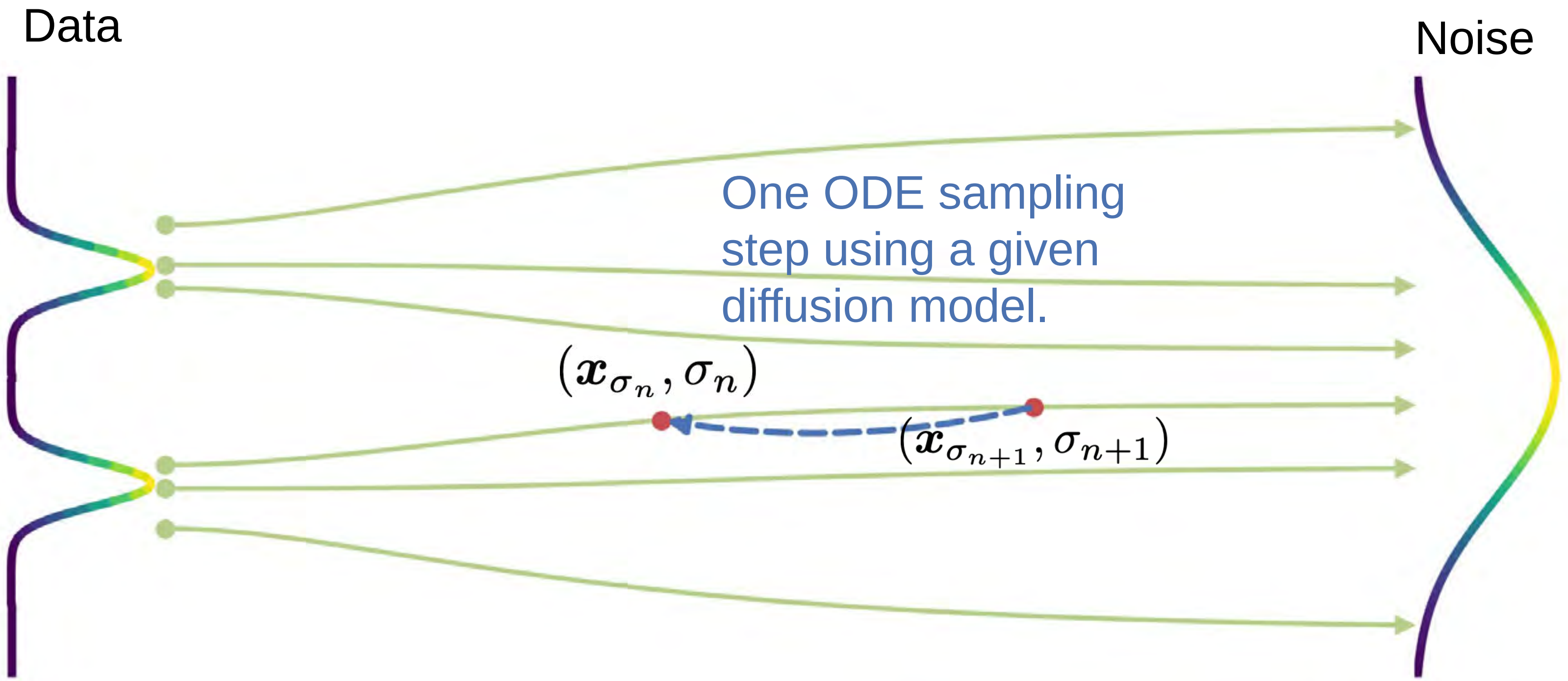
- The denoising/score network in diffusion models often has a similar parameterization (cf., EDM, v-prediction, etc.)

Karras, Tero, et al. "Elucidating the design space of diffusion-based generative models." *arXiv preprint arXiv:2206.00364* (2022).

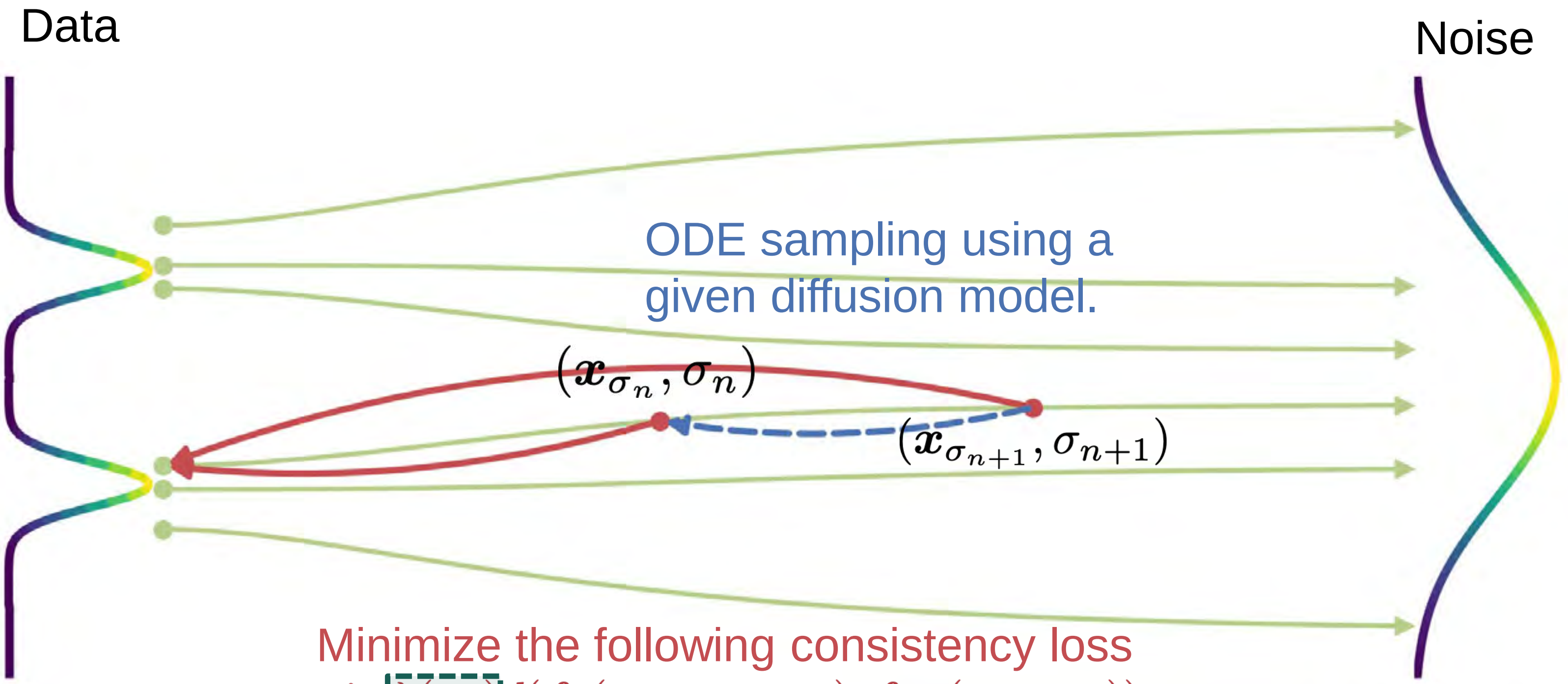
Training consistency models via distillation



Training consistency models via distillation



Training consistency models via distillation



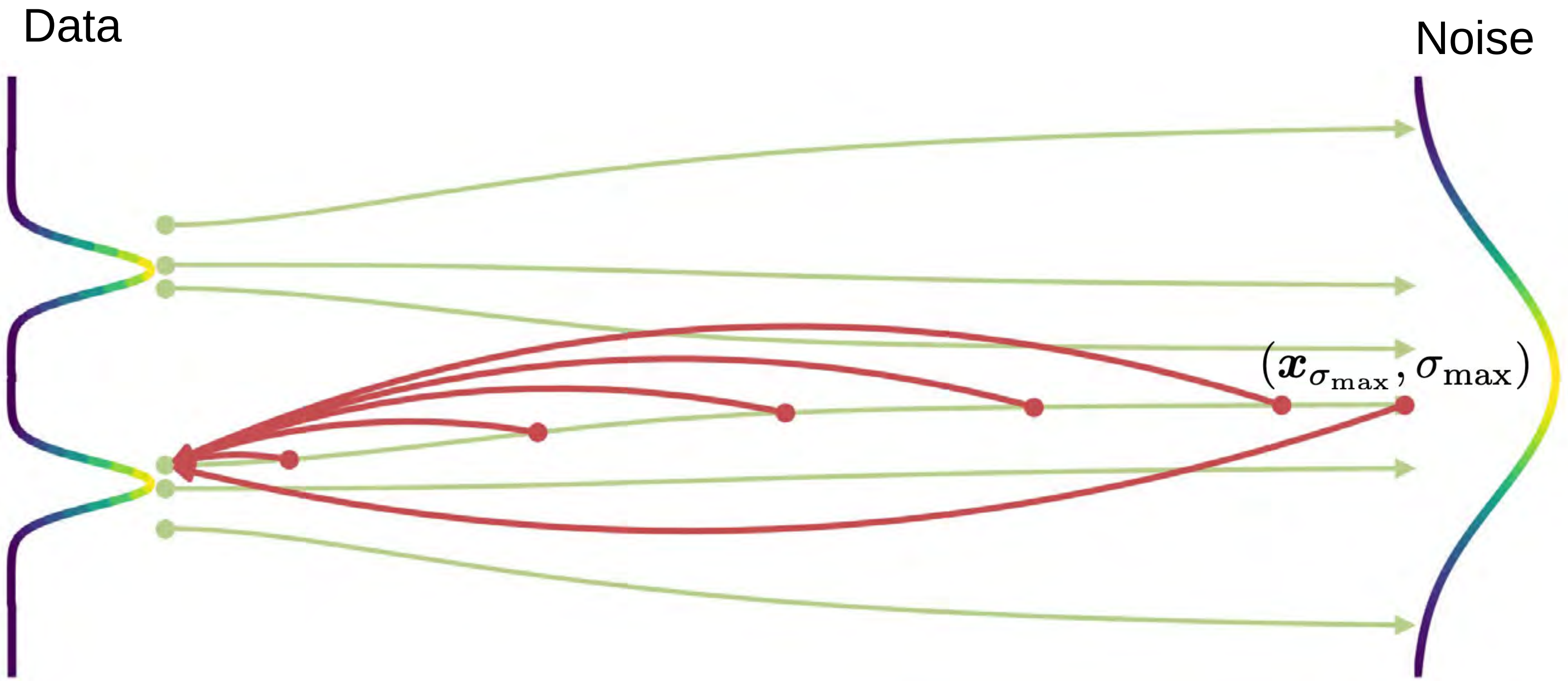
Minimize the following consistency loss

$$\min_{\boldsymbol{\theta}} \boxed{\lambda(\sigma_n)} d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_{n+1}}, \sigma_{n+1}), \boldsymbol{f}_{\boxed{\boldsymbol{\theta}^-}}(\boldsymbol{x}_{\sigma_n}, \sigma_n))$$

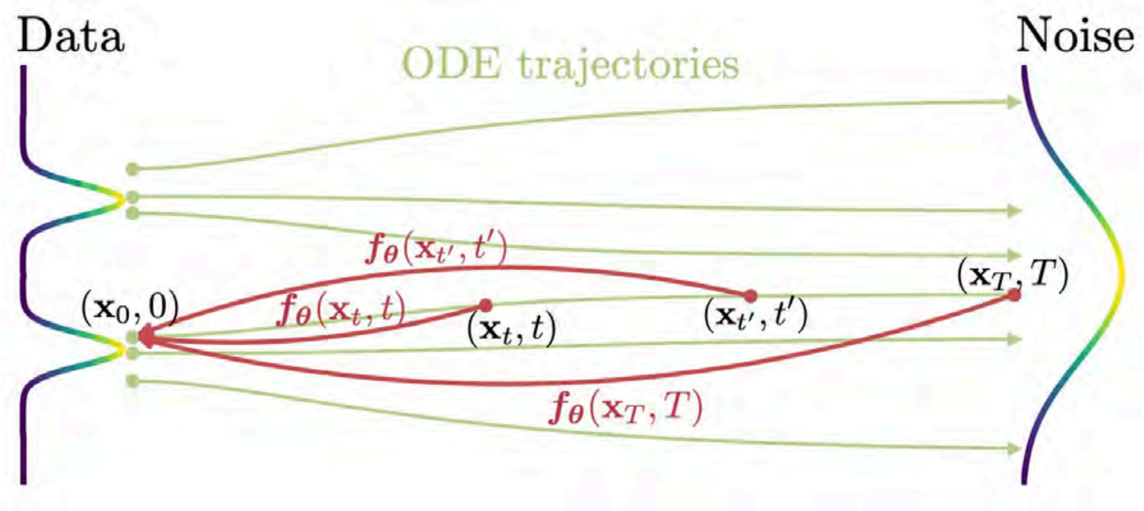
Weighting function

$$\boxed{\boldsymbol{\theta}^- = \mathrm{EMA}_\mu(\boldsymbol{\theta})}$$

Training consistency models via distillation



Enforcing self-consistency via distillation



$$\min_{\boldsymbol{\theta}} \|\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t'}, t') - \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\|_2^2$$

$$\mathbf{x}_0 \longleftarrow \mathbf{x}_t \longleftarrow \mathbf{x}_{t'} \longleftarrow \mathbf{x}_T$$

ODE solver + pretrained score function

ODE solver + pretrained score function

ODE solver + pretrained score function

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_0, 0)\|_2^2$$

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_T, T) - \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t'}, t')\|_2^2$$

Training consistency models via distillation

- Given a pre-trained score model $\boldsymbol{s}_\phi(\mathbf{x}, t)$
- With a random time step t_{n+1} and perturbed data point $\mathbf{x}_{t_{n+1}}$
  - Run one ODE step to move from time step t_{n+1} to time step t_n

$$\begin{aligned}\hat{\mathbf{x}}^{\phi}_{t_n} &:= \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}; \phi) \\ &\approx \mathbf{x}_{t_n}\end{aligned}$$

  - Minimize the consistency loss

$$\min_{\boldsymbol{\theta}} \underbrace{\lambda(t_n)}_{\text{The weighting function}} \| \underbrace{\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1})}_{\text{Student network}} - \underbrace{\boldsymbol{f}_{\boldsymbol{\theta}^-}(\underbrace{\hat{\mathbf{x}}^{\phi}_{t_n}}_{\approx \mathbf{x}_{t_n}}, t_n)}_{\text{Target network}} \|_2^2$$

Weights are obtained via exponential moving average

  - The L2 loss can be replaced with any other loss function, like LPIPS.

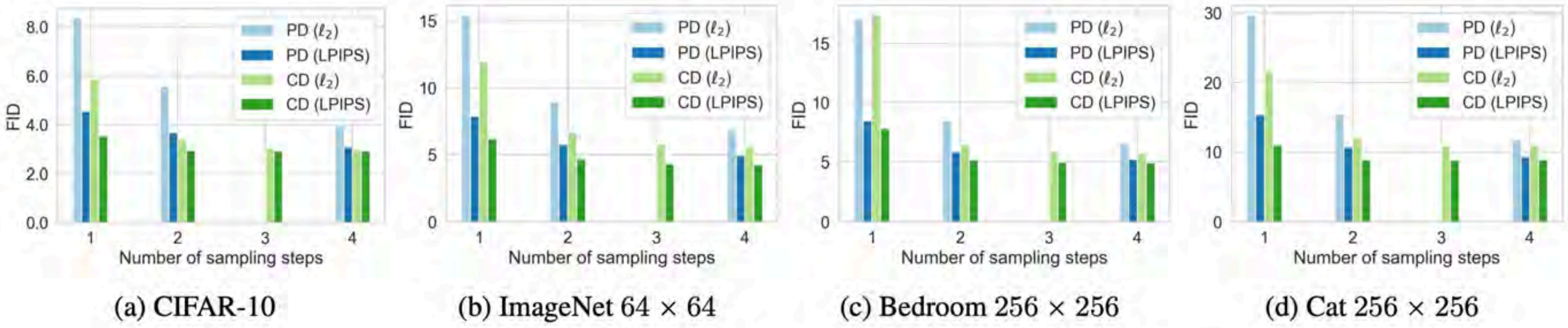
State-of-the-art few-step generation with consistency distillation (CD)

Results on the good old CIFAR-10 dataset

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**

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

State-of-the-art few-step generation with consistency distillation (CD)



Consistency distillation (CD) vs. progressive distillation (PD)

Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." *arXiv preprint arXiv:2202.00512* (2022).

Consistency models distilled from diffusion models

EDM
FID = 2.44
NFE = 79

One step
FID = 6.20
NFE = 1

Two steps
FID = 4.70
NFE = 2

Consistency models distilled from diffusion models

EDM
FID = 3.57
NFE = 79

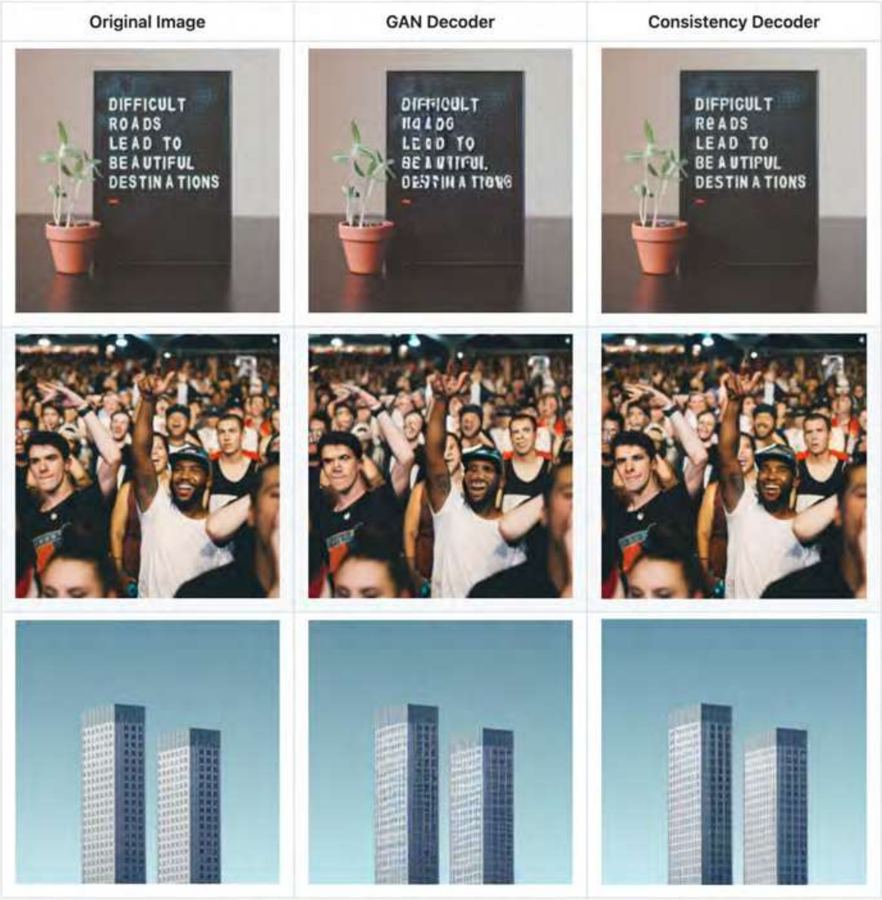
One step
FID = 7.80
NFE = 1

Two steps
FID = 5.22
NFE = 2

Consistency decoder in DALLE 3

Gabriel Goh @gabeeegoooh

if you get the chance, do try decoding your Stable Diffusion generations with this decoder. You should see some improvements in text, small faces and straight lines. Made with @_tim_brooks , @DrYangSong , @model_mechanic , @txhf , @neonbjb github.com/openai/consist...



Latent consistency models

Demo

Luo, S., Tan, Y., Huang, L., Li, J. and Zhao, H., 2023. Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference. *arXiv preprint arXiv:2310.04378*.

Training consistency models directly from data

- **Consistency training**
  - Sample a random noise level σ_{n+1}, a data point $\boldsymbol{x}$, and Gaussian noise $\boldsymbol{z}$
  - Minimize the following objective

$$\min_{\boldsymbol{\theta}} \mathbb{E}[\lambda(\sigma_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma_{n+1}\boldsymbol{z}, \sigma_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\boldsymbol{x} + \sigma_n \boldsymbol{z}, \sigma_n))]$$

- **Theoretical justification**

> When $|\Delta\sigma_n| = |\sigma_{n+1} - \sigma_n| \to 0$, the above objective converges to the distillation objective that uses the **ground truth diffusion model**.

- **No need to pretrain a diffusion model!**
- Can be generalized to non-L2 losses.

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Continuous-time consistency training

- **Motivation:** removing the potential bias in finite time steps.
- **Continuous-time consistency training**
  - Sample a random time step t, a data point $\mathbf{x}$, and a perturbed data point $\mathbf{x}_t$
  - Minimize the following objective

$$\mathbb{E}\left[\lambda(t)\boldsymbol{f}_{\boldsymbol{\theta}}^{\top}(\mathbf{x}_t, t)\,\mathrm{stopgrad}\left(\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} + \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \cdot \frac{\mathbf{x}_t - \mathbf{x}}{t}\right)\right]$$

- Can be generalized to non-l2 losses.
- No need to choose discrete time steps.
- **Pseudo-objective:** loss value is meaningless, but provides the right gradients.

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Catalog of one-step generative models

- **VAEs**
  - Stable training (maximum likelihood)
  - Tractable likelihood estimation
  - Low sample quality
- **GANs**
  - Unstable training (adversarial games)
  - High sample quality
  - No likelihoods
- **Normalizing flows**
  - Stable training (maximum likelihood)
  - Exact likelihood computation
  - Restricted model architecture
  - Low sample quality
- **Consistency models**
  - Stable training (pseudo-objective)
  - High sample quality
  - No likelihoods
  - Moderate architecture constraints.

Consistency models as new generative models

Results on CIFAR-10

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**

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Consistency models as new generative models

METHOD	NFE (↓)	FID (↓)	Prec. (↑)	Rec. (↑)
ImageNet 64 × 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				
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

METHOD	NFE (↓)	FID (↓)	Prec. (↑)	Rec. (↑)
LSUN Cat 256 × 256				
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

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Consistency models as new generative models

EDM
FID = 2.44
NFE = 79

One step
FID = 12.96
NFE = 1

Two steps
FID = 11.12
NFE = 2

Consistency models as new generative models

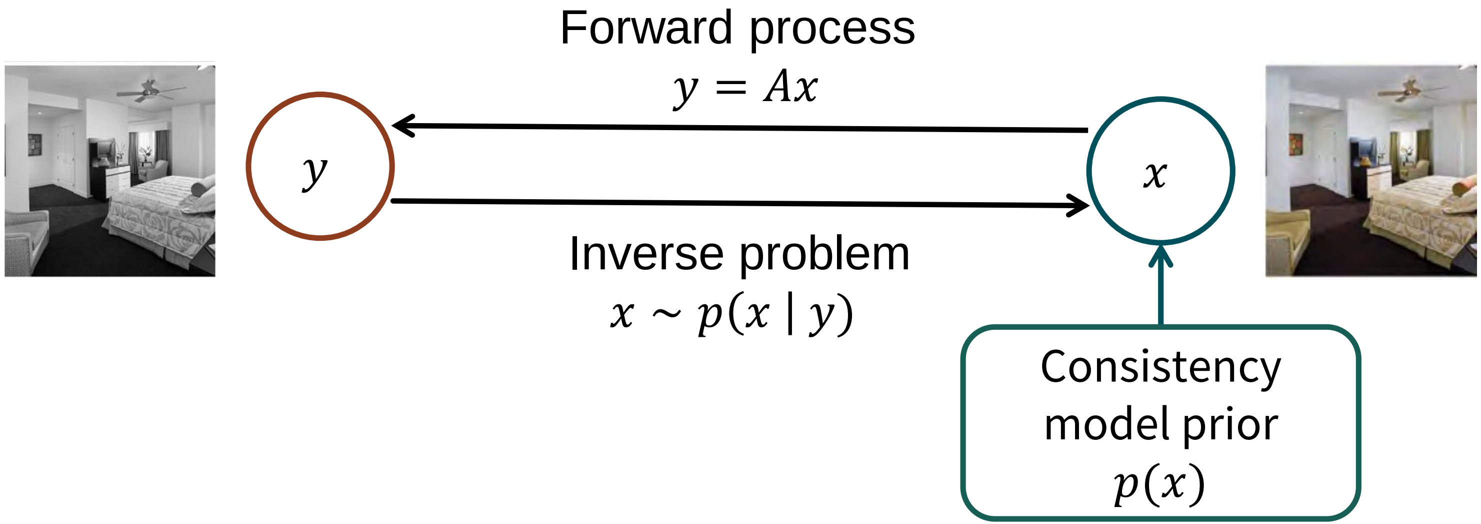
EDM
FID = 3.57
NFE = 79

One step
FID = 16.00
NFE = 1

Two steps
FID = 7.80
NFE = 2

Solving linear inverse problems with consistency models

- **Problem:** solving linear inverse problems with a consistency model prior.

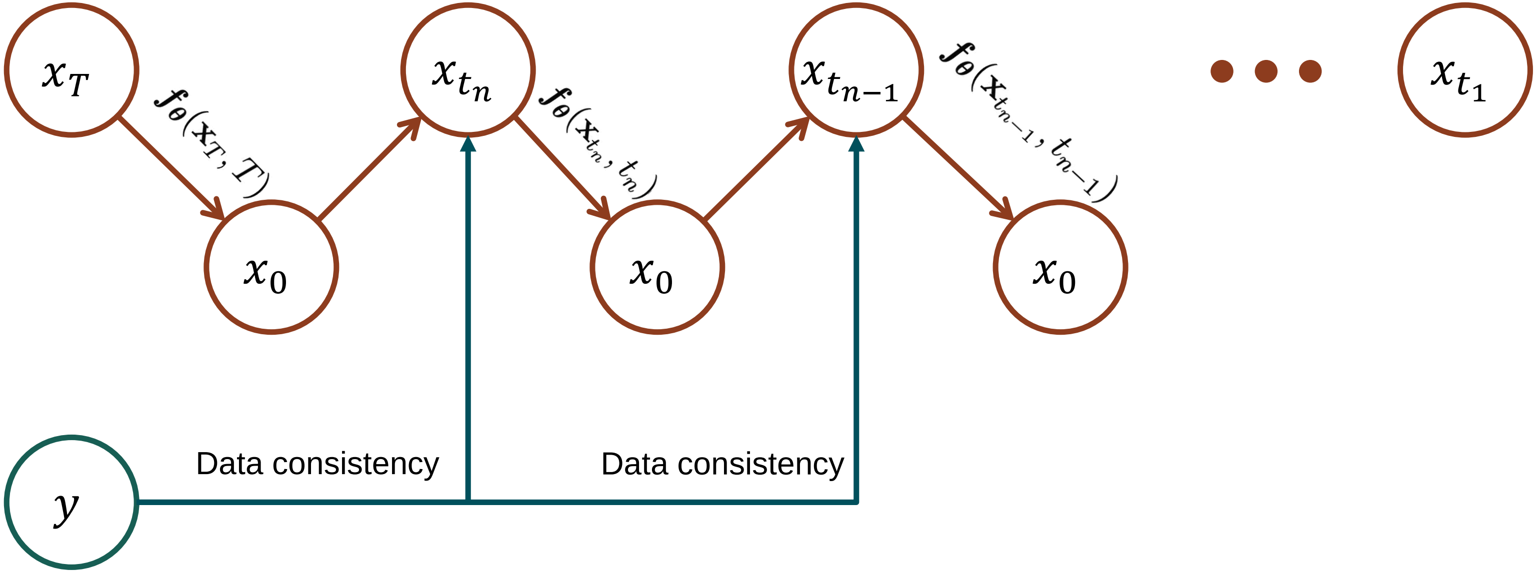


- **Examples:** colorization, inpainting, super-resolution, computed tomography, magnetic resonance imaging, cryo-electon tomography, photoacoustic tomography, …

Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Solving linear inverse problems with consistency models

- **Algorithm:** alternating data generation and data consistency steps.



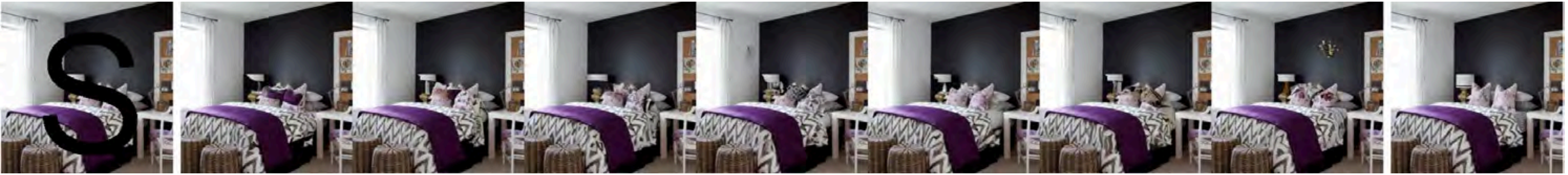
Song, Dhariwal, Chen, Sutskever. Consistency Models. ICML 2023

Zero-shot image editing

Colorization

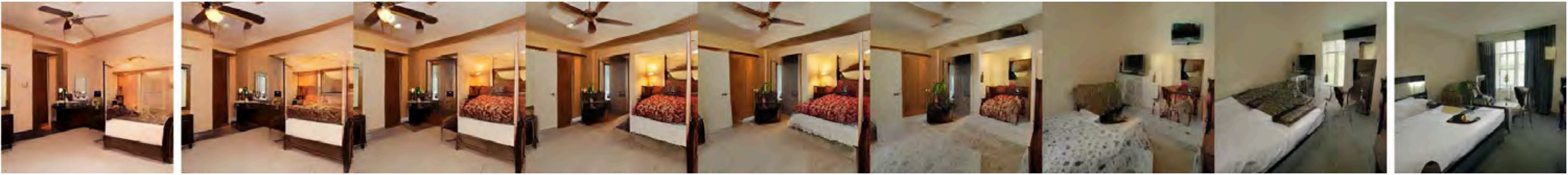
Super-resolution

Inpainting

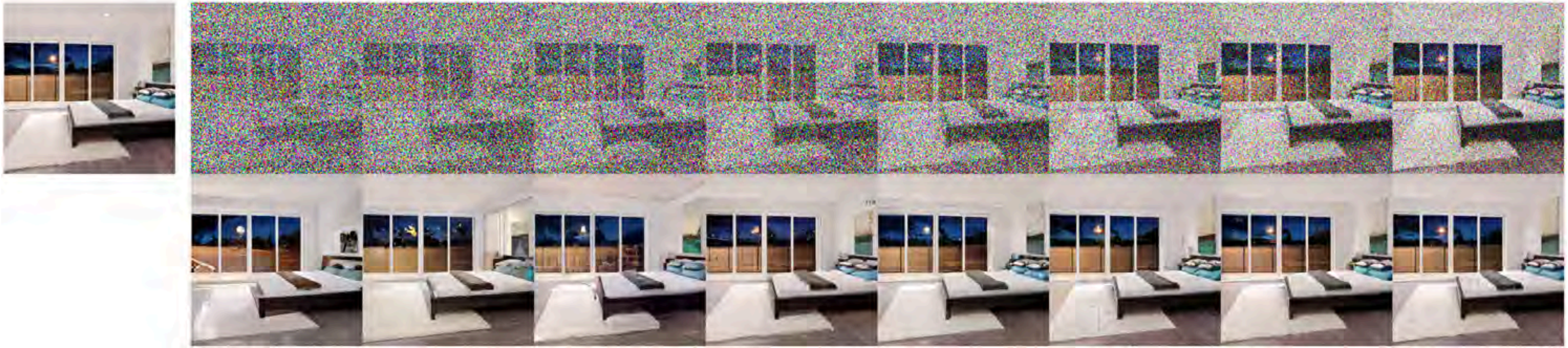


Zero-shot image editing

Interpolation



One-step denoising



Zero-shot image editing

Stroke-guided image generation

Summary

- Consistency models have native support for one-step generation.
- Consistency models allow multistep generation and zero-shot image editing.
- Consistency models are both a diffusion distillation technique, and a new generative model.

Improved Techniques for Consistency Training

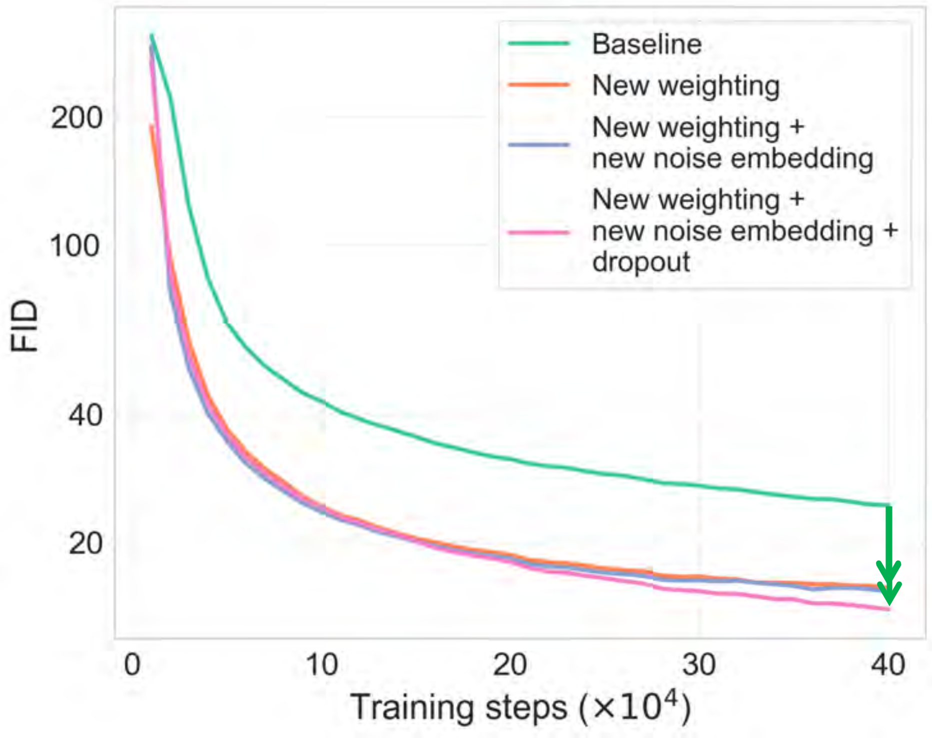
Song, Dhariwal. Improved Techniques for Consistency Training, ICLR 2024

Weighting functions, noise embeddings, and dropout

- ~~Uniform weighting~~ → Larger weighting for smaller noise

$$\lambda(\sigma_n) \propto \frac{1}{\sigma_{n+1} - \sigma_n}$$

- Reducing the sensitivity of noise embeddings for better training stability
- Larger dropout than diffusion models results in higher one-step quality



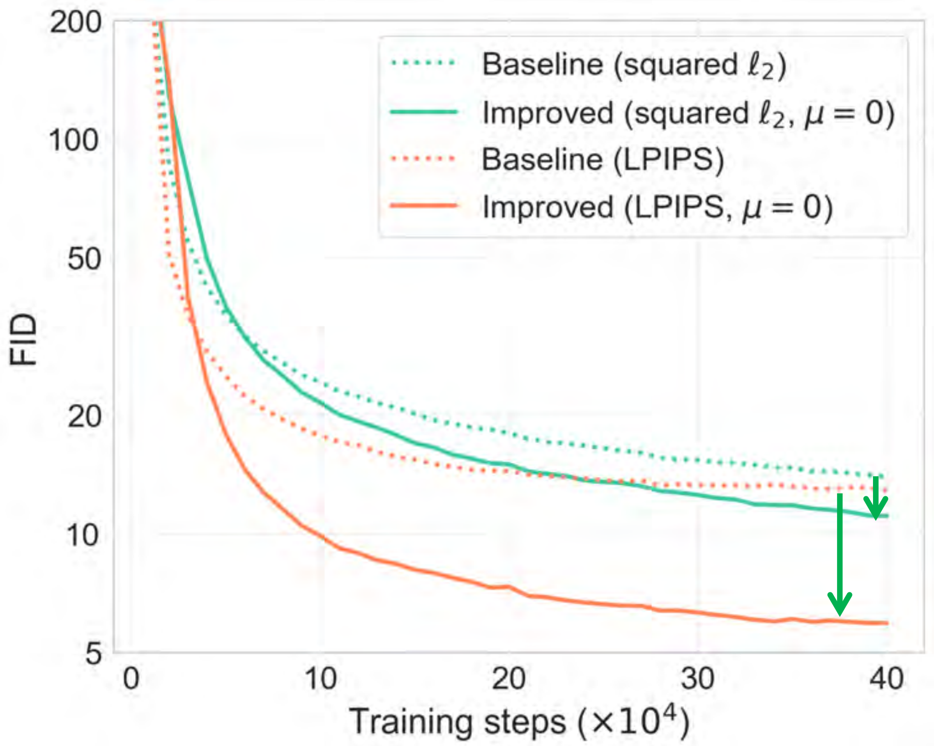
Using zero EMA decay rate for theoretical soundness

- **Previous belief:** the consistency training loss converges to the loss of consistency distillation in the limit of small noise gaps.
- **Improved analysis:** their gradients must match as well, which **only happens when the EMA is zero.**

$$\boldsymbol{\theta}^- = \mathrm{EMA}_\mu(\boldsymbol{\theta})$$

$$\Big\downarrow \; \mu = 0$$

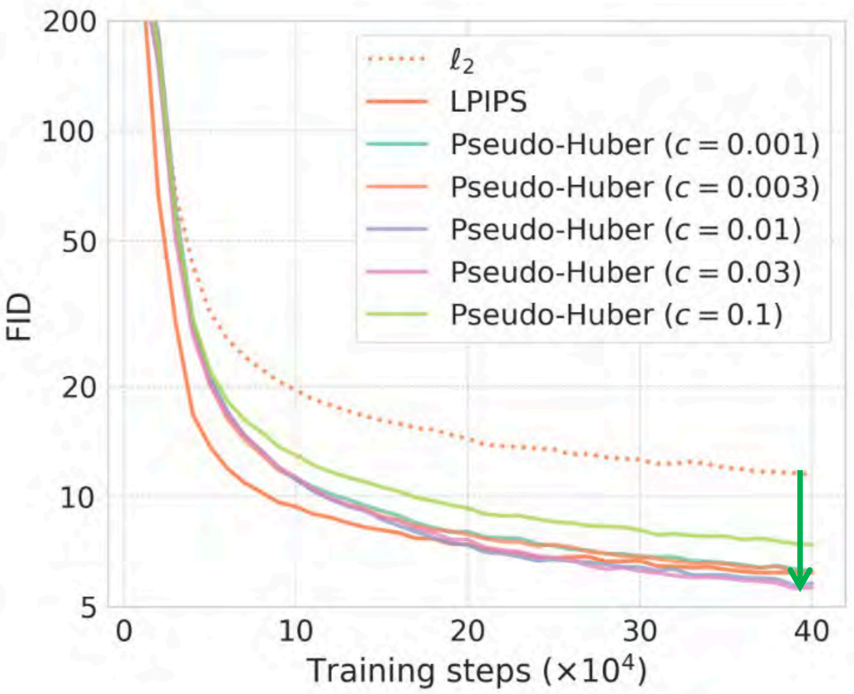
$$\boldsymbol{\theta}^- = \mathrm{stopgrad}(\boldsymbol{\theta})$$



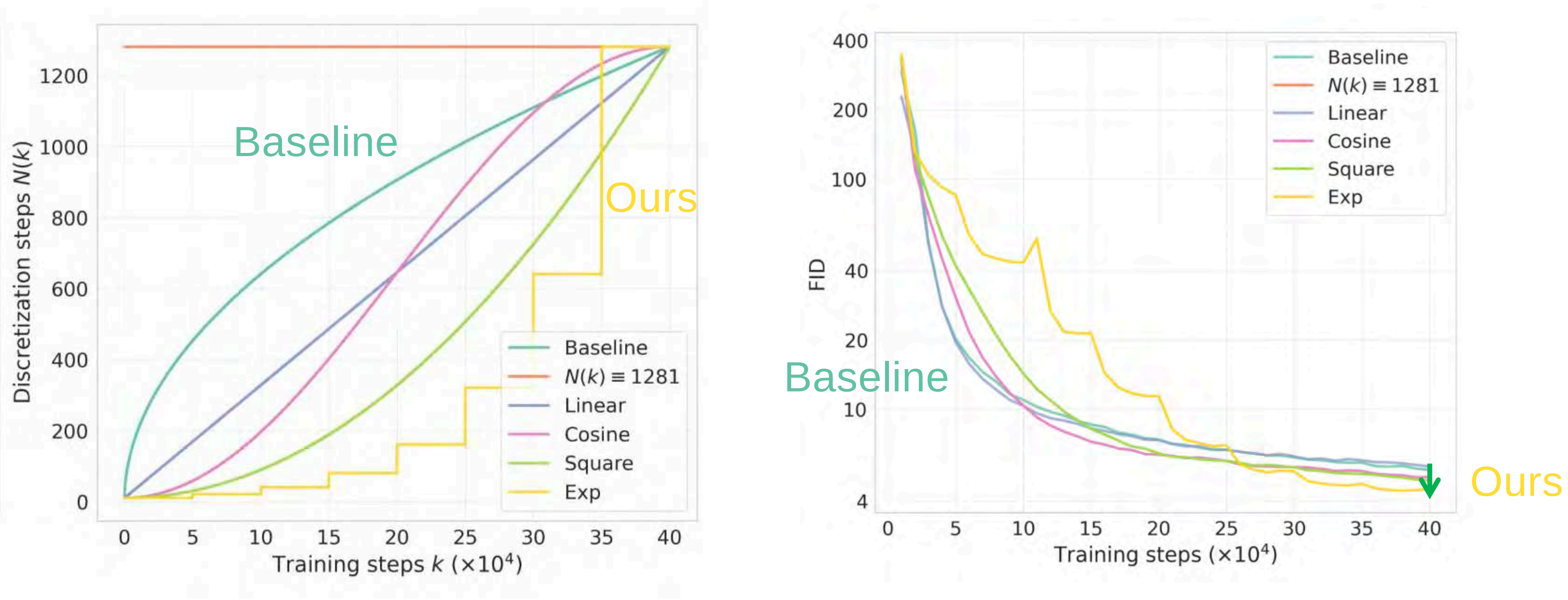
Measuring self-consistency with pseudo-Huber loss

- Squared L2 distance has poor performance.
- LPIPS performs well but biases evaluation.
- **Our solution:** pseudo-Huber losses

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + c^2} - c$$



Improving the noise schedule

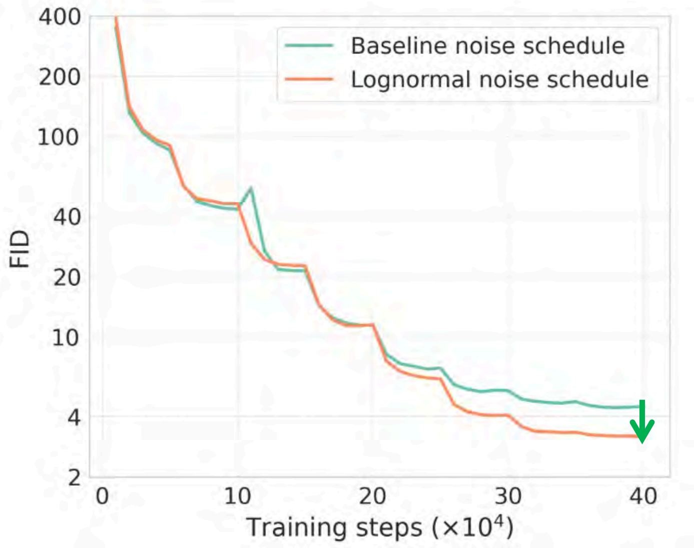


Double the total number of noise levels
during training per fixed number of iterations.

Improving the noise schedule

Sampling noise levels according to discretized lognormal

$$p(\sigma_i) \propto \operatorname{erf}\left(\frac{\log(\sigma_{i+1}) - P_{\text{mean}}}{\sqrt{2}P_{\text{std}}}\right) - \operatorname{erf}\left(\frac{\log(\sigma_i) - P_{\text{mean}}}{\sqrt{2}P_{\text{std}}}\right)$$



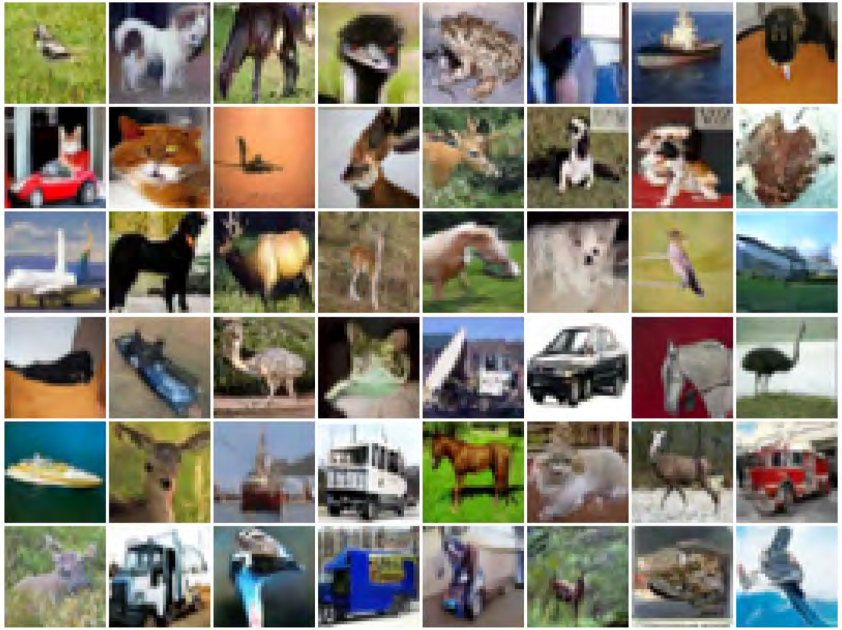
Improved Techniques for Consistency Training

Results on CIFAR-10

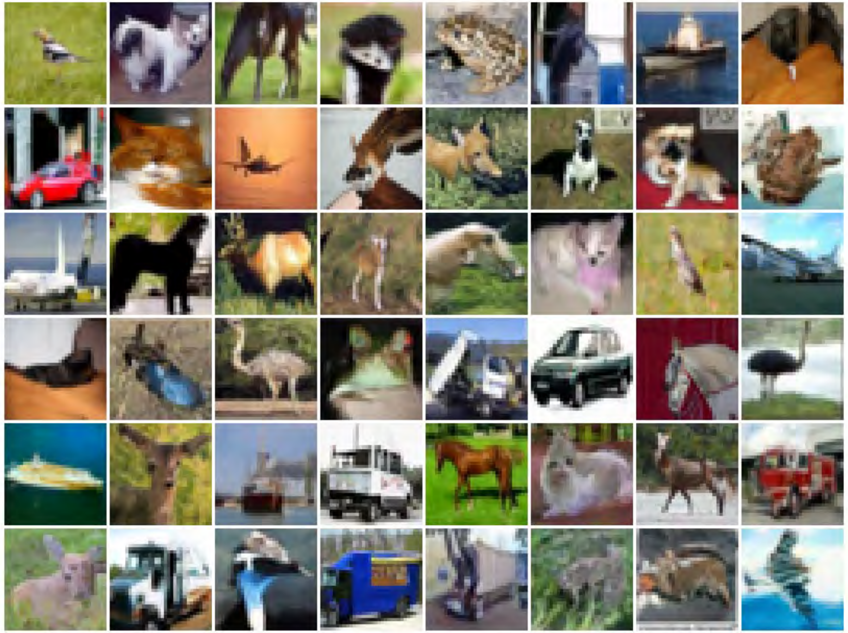
METHOD	NFE (↓)	FID (↓)	IS (↑)
Fast samplers & distillation for diffusion models			
DDIM (Song et al., 2020)	10	13.36	
DPM-solver-fast (Lu et al., 2022)	10	4.70	
3-DEIS (Zhang & Chen, 2022)	10	4.17	
UniPC (Zhao et al., 2023)	10	3.87	
Knowledge Distillation (Luhman & Luhman, 2021)	1	9.36	
DFNO (LPIPS) (Zheng et al., 2022)	1	3.78	
2-Rectified Flow (+distill) (Liu et al., 2022)	1	4.85	9.01
TRACT (Berthelot et al., 2023)	1	3.78	
	2	3.32	
Diff-Instruct (Luo et al., 2023)	1	4.53	9.89
PD* (Salimans & Ho, 2022)	1	8.34	8.69
	2	5.58	9.05
CD (LPIPS) (Song et al., 2023)	1	3.55	9.48
	2	2.93	9.75

METHOD	NFE (↓)	FID (↓)	IS (↑)
Direct Generation			
Score SDE (Song et al., 2021)	2000	2.38	9.83
Score SDE (deep) (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
EDM-G++ (Kim et al., 2023)	35	1.77	
IGEBM (Du & Mordatch, 2019)	60	40.6	6.02
NVAE (Vahdat & Kautz, 2020)	1	23.5	7.18
Glow (Kingma & Dhariwal, 2018)	1	48.9	3.92
Residual Flow (Chen et al., 2019)	1	46.4	
BigGAN (Brock et al., 2019)	1	14.7	9.22
StyleGAN2 (Karras et al., 2020b)	1	8.32	9.21
StyleGAN2-ADA (Karras et al., 2020a)	1	2.92	9.83
CT (LPIPS) (Song et al., 2023)	1	8.70	8.49
	2	5.83	8.85
iCT (ours)	1	2.83	9.54
	2	2.46	9.80
iCT-deep (ours)	1	2.51	9.76
	2	2.24	9.89

CIFAR-10 samples from improved consistency training



One step
FID = 2.51, IS = 9.76
NFE = 1



Two step
FID = 2.24, IS = 9.89
NFE = 2

Improved techniques for consistency training

	Design choice in Song et al. (2023)	Our modifications
EMA decay rate for the teacher network	$\mu(k) = \exp(\frac{s_0 \log \mu_0}{N(k)})$	$\mu(k) = 0$
Metric in consistency loss	$d(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{LPIPS}(\boldsymbol{x}, \boldsymbol{y})$	$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\lVert \boldsymbol{x} - \boldsymbol{y} \rVert_2^2 + c^2} - c$
Discretization curriculum	$N(k) = \left\lceil \sqrt{\frac{k}{K}((s_1+1)^2 - s_0^2) + s_0^2} - 1 \right\rceil + 1$	$N(k) = \min(s_0 2^{\lfloor \frac{k}{K'} \rfloor}, s_1) + 1$, where $K' = \left\lfloor \frac{K}{\log_2 \lfloor s_1/s_0 \rfloor + 1} \right\rfloor$
Noise schedule	t_i, where $i \sim \mathcal{U}[\![1, N(k) - 1]\!]$	t_i, where $i \sim p(i)$, and $p(i) \propto \mathrm{erf}\left(\frac{\log(t_{i+1}) - P_{\text{mean}}}{\sqrt{2} P_{\text{std}}}\right) - \mathrm{erf}\left(\frac{\log(t_i) - P_{\text{mean}}}{\sqrt{2} P_{\text{std}}}\right)$
Weighting function	$\lambda(t_i) = 1$	$\lambda(t_i) = \frac{1}{t_{i+1} - t_i}$
Parameters	$s_0 = 2, s_1 = 150, \mu_0 = 0.9$ on CIFAR-10 $s_0 = 2, s_1 = 200, \mu_0 = 0.95$ on ImageNet 64×64	$s_0 = 10, s_1 = 1280$ $c = 0.00054\sqrt{d}$, d is data dimensionality $P_{\text{mean}} = -1.1, P_{\text{std}} = 2.0$
	$k \in [\![0, K]\!]$, where K is the total training iterations $t_i = (t_{\min}^{1/\rho} + \frac{i-1}{N(k)-1}(t_{\max}^{1/\rho} - t_{\min}^{1/\rho}))^\rho$, where $i \in [\![1, N(k)]\!], \rho = 7, t_{\min} = 0.002, t_{\max} = 80$	

Song, Dhariwal. Improved Techniques for Consistency Training, 2023

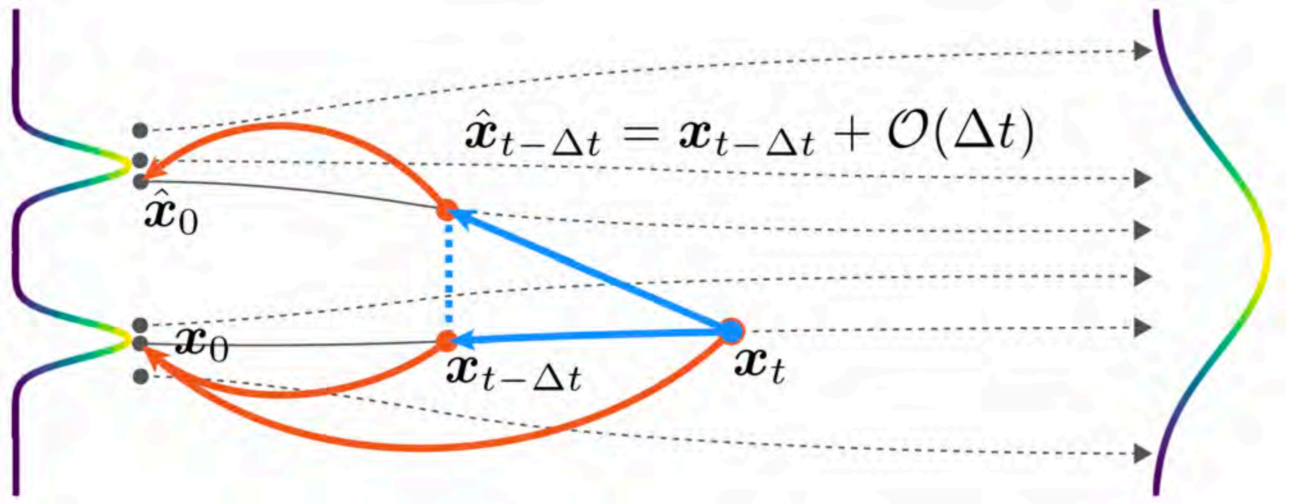
Summary

- Remove the dependency on LPIPS
  - Faster training
  - No metric gaming and data contamination in evals.
- Recipe for consistency training that outperforms consistency distillation
- Consistency models are among the state-of-the-art one-step generative models, on par with GANs, and better than VAEs, normalizing flows, etc.

Continuous-Time Consistency Models

Lu, Song. Simplifying, Stabilizing & Scaling Continuous-Time Consistency Models. 2024

Accumulated discretization errors in discrete-time CMs

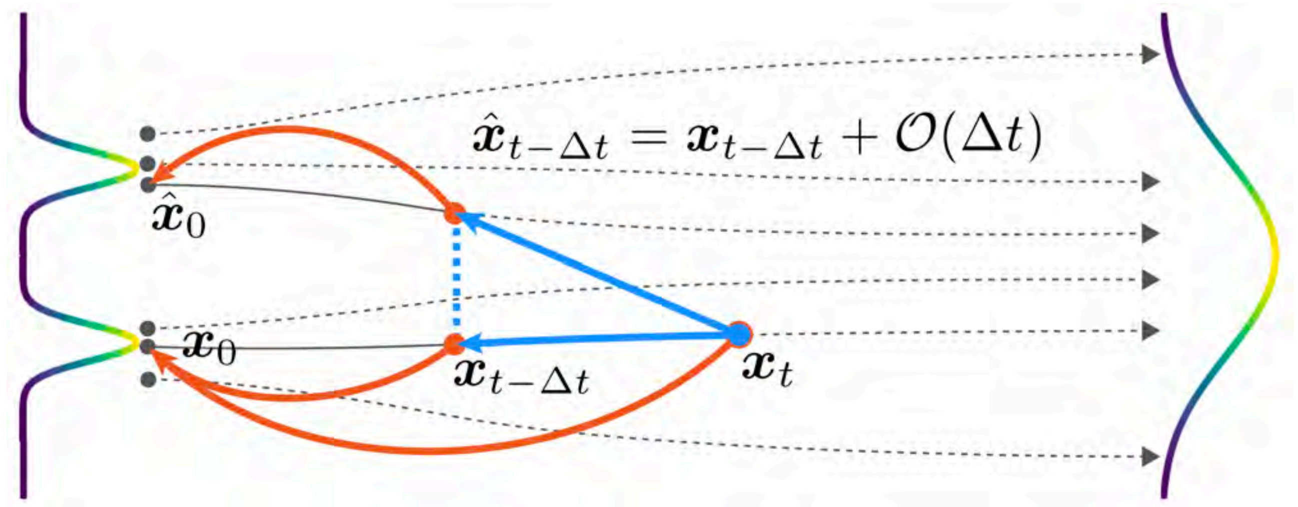


Small discretization errors at intermediate steps can compound

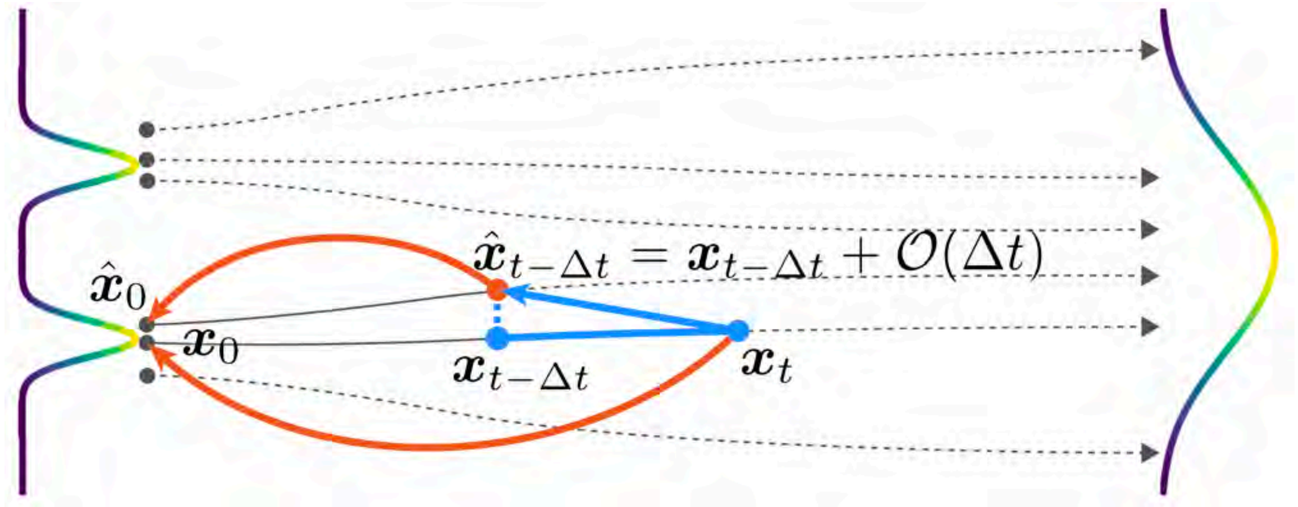
$$\min_{\boldsymbol{\theta}} \lambda(\sigma_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_{n+1}}, \sigma_{n+1}), \boxed{\boldsymbol{f}_{\boldsymbol{\theta}^-}(\boldsymbol{x}_{\sigma_n}, \sigma_n)})$$

May be on different trajectory

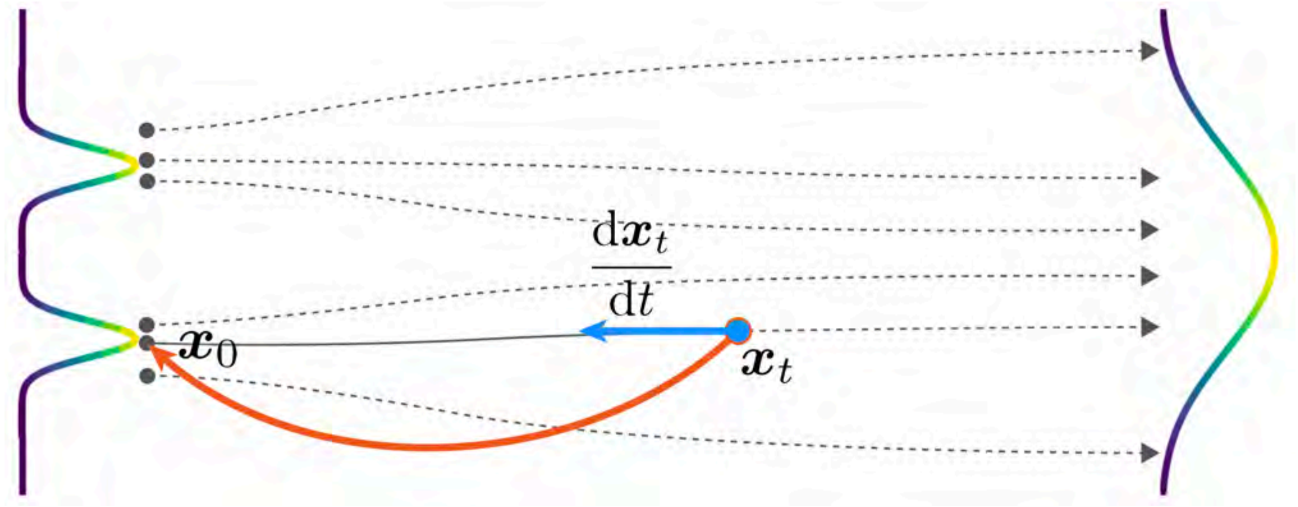
Accumulated discretization errors in discrete-time CMs



Accumulated discretization errors in discrete-time CMs



Continuous-time CMs: avoid discretization errors



Using the tangent direction as supervision signals

Continuous-time CMs: Two types of derivations

Continuous-time limit of the consistency distillation objective:

$$\lim_{\Delta t \to 0} \frac{1}{(\Delta t)^2} \| f_\theta(x_t, t) - f_{\theta^-}(x_t, t) \|_2^2 = \boxed{\left\| \frac{\mathrm{d} f_\theta(x_t, t)}{\mathrm{d}t} \right\|_2^2}$$

Continuous-time limit of the gradient:

$$\lim_{\Delta t \to 0} \nabla_\theta \frac{1}{\Delta t} \| f_\theta(x_t, t) - f_{\theta^-}(x_t, t) \|_2^2 = \nabla_\theta f_\theta^\top (x_t, t) \boxed{\frac{\mathrm{d} f_{\theta^-}(x_t, t)}{\mathrm{d}t}}$$

Stop gradients for θ

Continuous-time CMs: Taking limits for gradient!

Limit of objective: $$\left\| \frac{\mathrm{d} f_\theta(x_t, t)}{\mathrm{d}t} \right\|_2^2$$

- Gradients do not match discrete-time CMs
- Difficulty from “gradient of gradient”

Limit of gradient: $$f_\theta^\top(x_t, t) \frac{\mathrm{d} f_{\theta^-}(x_t, t)}{\mathrm{d}t}$$

- Smoothly transition the gradient landscape from discrete time to continuous time
- **Deep-learning-friendly** ☺

Continuous-time consistency training: unbiased gradient!

$$\frac{\mathrm{d}f_{\theta^-}(x_t,t)}{\mathrm{d}t} = \nabla_{x_t} f_{\theta^-} \cdot \frac{\mathrm{d}x_t}{dt} + \partial_t f_{\theta^-}$$

- Same estimator in Flow Matching!

$$\mathbb{E}_{x_0,z}\left[\frac{\mathrm{d}f_{\theta^-}(x_t,t)}{\mathrm{d}t}\right] = \mathbb{E}_{x_t}\left[\nabla_{x_t} f_{\theta^-} \cdot \mathbb{E}_{x_0|x_t}\left[\dot{\alpha}_t x_0 + \dot{\sigma}_t z\right] + \partial_t f_{\theta^-}\right]$$

$$z = \frac{x_t - \alpha_t x_0}{\sigma_t}$$

In continuous-time, the gradient of CT is an **unbiased estimator** of that of CD.

Key difficulty in training continuous-time CMs: tangent variance

$$\nabla_\theta \mathbb{E}_{\boldsymbol{x}_t, t}\left[w(t) \boldsymbol{f}_\theta^\top(\boldsymbol{x}_t, t) \frac{\mathrm{d}\boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t)}{\mathrm{d}t} \right]$$

$$= \nabla_{\boldsymbol{x}_t} \boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t) \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \partial_t \boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t)$$

The variance of the tangent causes training instability!

e.g., previous works find that when decreasing Δt, the training becomes extremely unstable, leading to worse performance.

Part 1: Simplified formulations of diffusion models and CMs

TrigFlow: unifying EDM and Flow Matching by trigonometric interpolations

Diffusion process: $\boldsymbol{x}_t = \cos(t)\boldsymbol{x}_0 + \sin(t)\boldsymbol{z}$ for $t \in [0, \frac{\pi}{2}]$

PF-ODE: $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$

Consistency model: $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$

Boundary condition: $f(x, 0) \equiv x$

Part 2: Stabilizing the training of continuous-time CMs

Consistency model:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent:

$$\frac{\mathrm{d}\boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t)}{\mathrm{d}t} = -\cos(t)\left(\sigma_d \boldsymbol{F}_{\theta^-}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, t\right) - \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}\right) - \sin(t)\left(\boldsymbol{x}_t + \sigma_d \frac{\mathrm{d}\boldsymbol{F}_{\theta^-}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, t\right)}{\mathrm{d}t}\right)$$

Stable
(Init from diffusion)

Stable
(constant variance)

Stable
(pretrained diffusion)

Part 2: Stabilizing the training of continuous-time CMs

Consistency model:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent:

$$\frac{\mathrm{d}\boldsymbol{F}_{\theta^-}}{\mathrm{d}t} = \sin(t)\nabla_{\boldsymbol{x}_t}\boldsymbol{F}_{\theta^-}\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \sin(t)\partial_t \boldsymbol{F}_{\theta^-}$$

Stable (Jacobian during BP)

Stable (pretrained diffusion)

Part 2: Stabilizing the training of continuous-time CMs

Consistency model: $$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent: $$\sin(t)\partial_t \boldsymbol{F}_{\theta^-} = \sin(t)\frac{\partial c_{\text{noise}}(t)}{\partial t} \cdot \frac{\partial \text{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}} \cdot \frac{\partial \boldsymbol{F}_{\theta^-}}{\partial \text{emb}(c_{\text{noise}})}$$

Key: design network architecture that is well-conditioned w.r.t. t.

Part 2: Stabilizing the training of continuous-time CMs

Consistency model:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent:

$$\sin(t)\partial_t \boldsymbol{F}_{\theta^-} = \sin(t)\boxed{\frac{\partial c_{\text{noise}}(t)}{\partial t}} \cdot \frac{\partial \text{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}} \cdot \frac{\partial \boldsymbol{F}_{\theta^-}}{\partial \text{emb}(c_{\text{noise}})}$$

- Previous methods: $c_{noise}(t) = \log(\tan(t))$
- Ours: $c_{noise}(t) = t$

Part 2: Stabilizing the training of continuous-time CMs

Consistency model:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent:

$$\sin(t)\partial_t \boldsymbol{F}_{\theta^-} = \sin(t)\frac{\partial c_{\text{noise}}(t)}{\partial t} \cdot \boxed{\frac{\partial \text{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}}} \cdot \frac{\partial \boldsymbol{F}_{\theta^-}}{\partial \text{emb}(c_{\text{noise}})}$$

- Use small Fourier scales
- Reason: *cos'(fx) = - f * sin(fx)*

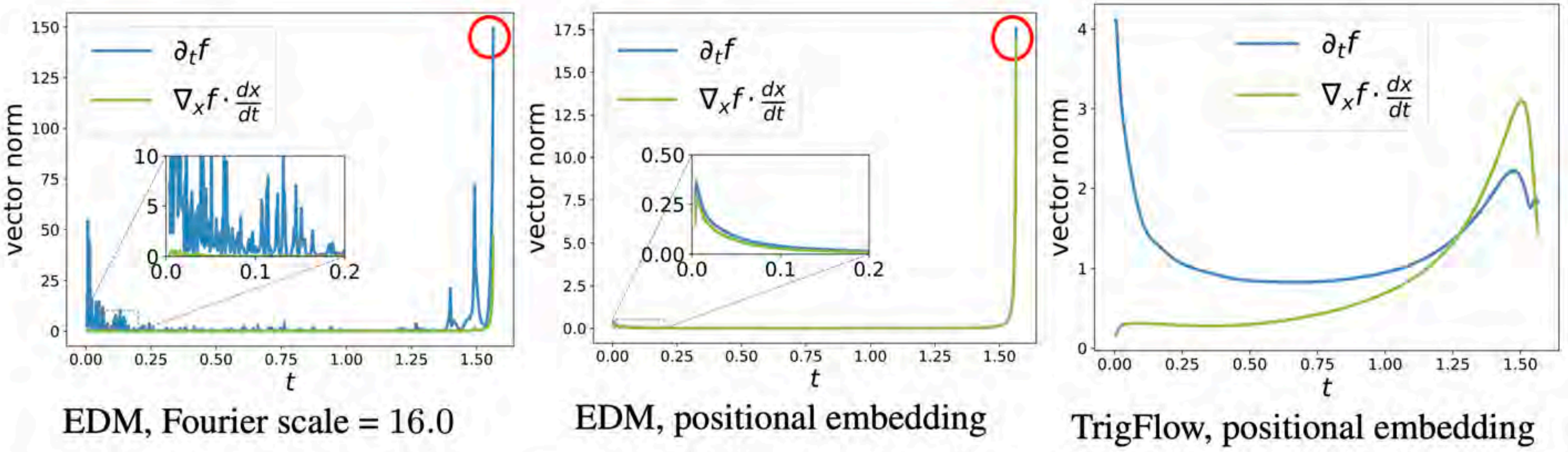
Part 2: Stabilizing the training of continuous-time CMs

Consistency model: $$\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_\theta\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

Tangent: $$\sin(t)\partial_t \boldsymbol{F}_{\theta^-} = \sin(t)\frac{\partial c_{\text{noise}}(t)}{\partial t} \cdot \frac{\partial \text{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}} \cdot \boxed{\frac{\partial \boldsymbol{F}_{\theta^-}}{\partial \text{emb}(c_{\text{noise}})}}$$

- Modify AdaGN layer to add normalization in time.

Part 2: Stabilizing the training of continuous-time CMs



Part 3: Reducing the variance across time steps

Original objective: $$\min_{\theta} \mathbb{E}_t \left[\boldsymbol{F}_\theta^\top \boldsymbol{y} \right] \Longleftrightarrow \nabla_\theta \mathbb{E}_t \left[\|\boldsymbol{F}_\theta - \boldsymbol{F}_{\theta^-} + \boldsymbol{y}\|_2^2 \right]$$

Adaptive weighting: $$\min_{\phi} \mathbb{E}_t \left[\boxed{\frac{e^{w_\phi(t)}}{D}} \|\boldsymbol{F}_\theta - \boldsymbol{F}_{\theta^-} + \boldsymbol{y}\|_2^2 - w_\phi(t) \right]$$

Optimal weighting will balance the variance across time steps: $$\boxed{\frac{e^{w^*(t)}}{D}} \mathbb{E} \left[\|\boldsymbol{F}_\theta - \boldsymbol{F}_{\theta^-} + \boldsymbol{y}\|_2^2 \right] \equiv 1$$

No need for manually-designed weighting!

Part 4: Reducing the variance across data points

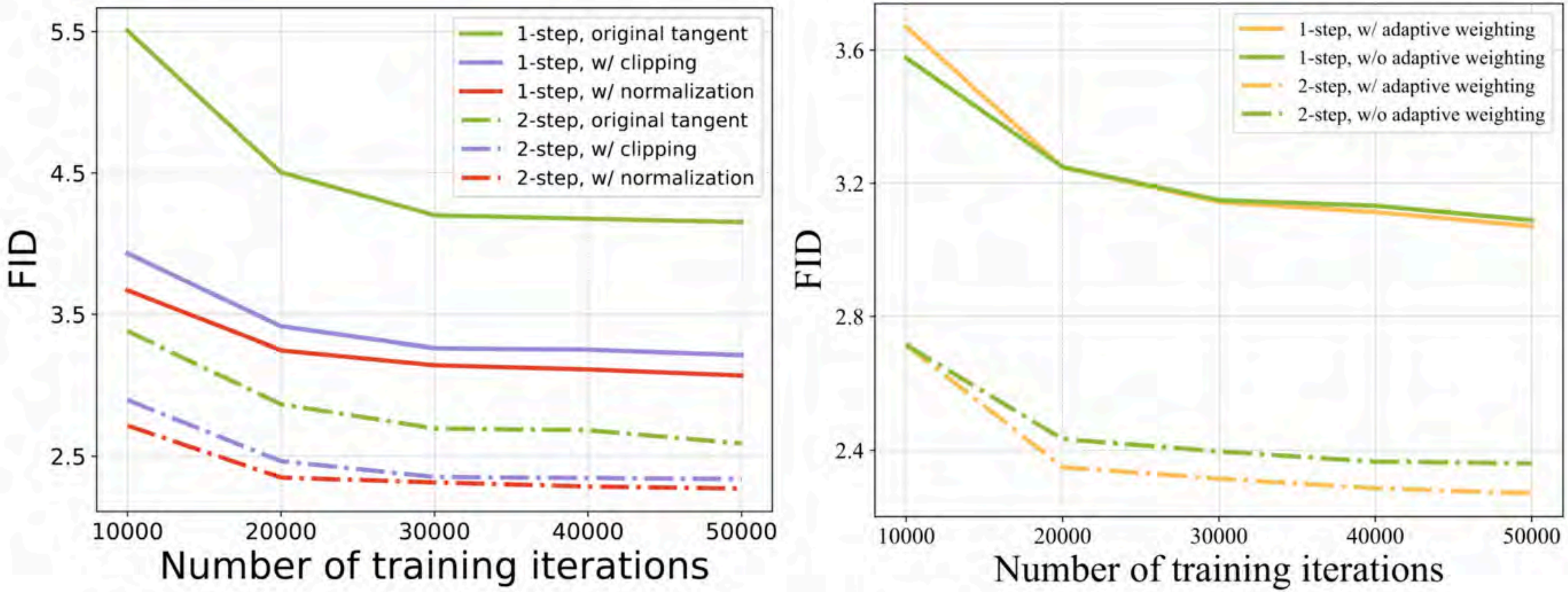
$\frac{\mathrm{d}\boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t)}{\mathrm{d}t}$ may have outliers at some dimensions of certain inputs

Intuition: mapping Gaussian to mixture-of-Gaussians will always have non-Lipschitzness at the boundary.

Solution: ignore the outliers, focus on the training at high density regions

→ **Replace $\frac{df}{dt}$ with $\frac{df}{dt} / (\left\| \frac{df}{dt} \right\| + c)$, where c is a hyperparameter.**

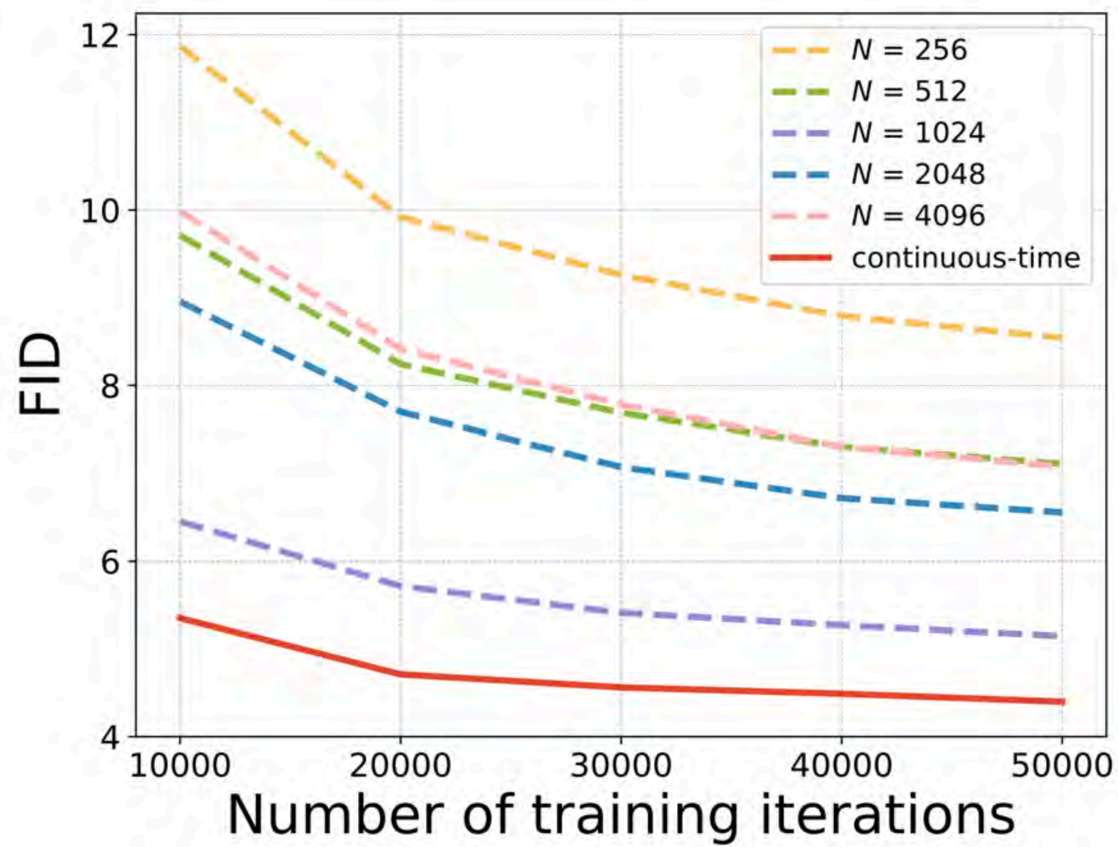
Effectiveness of variance reduction



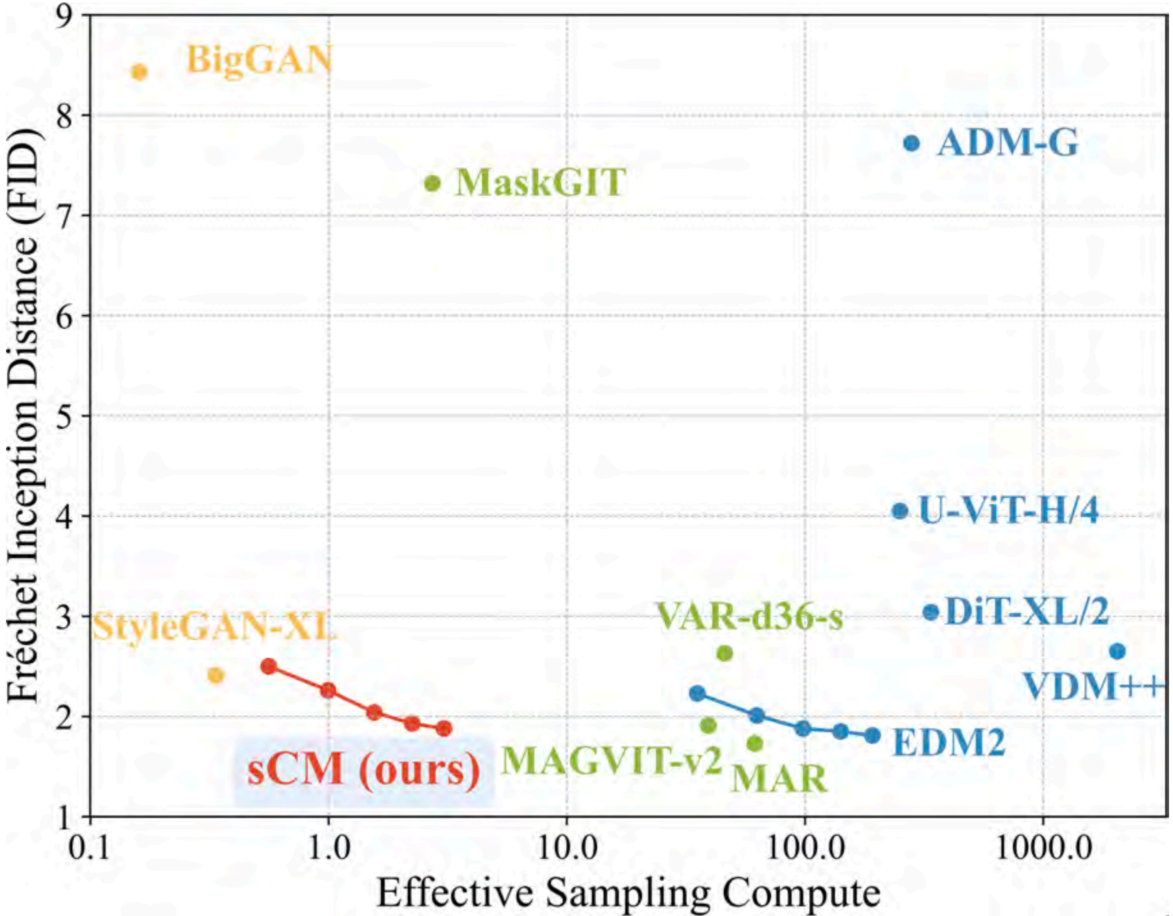
(a) Tangent Normalization (b) Adaptive Weighting

Continuous-time CMs outperform discrete-time CMs

For the first time!



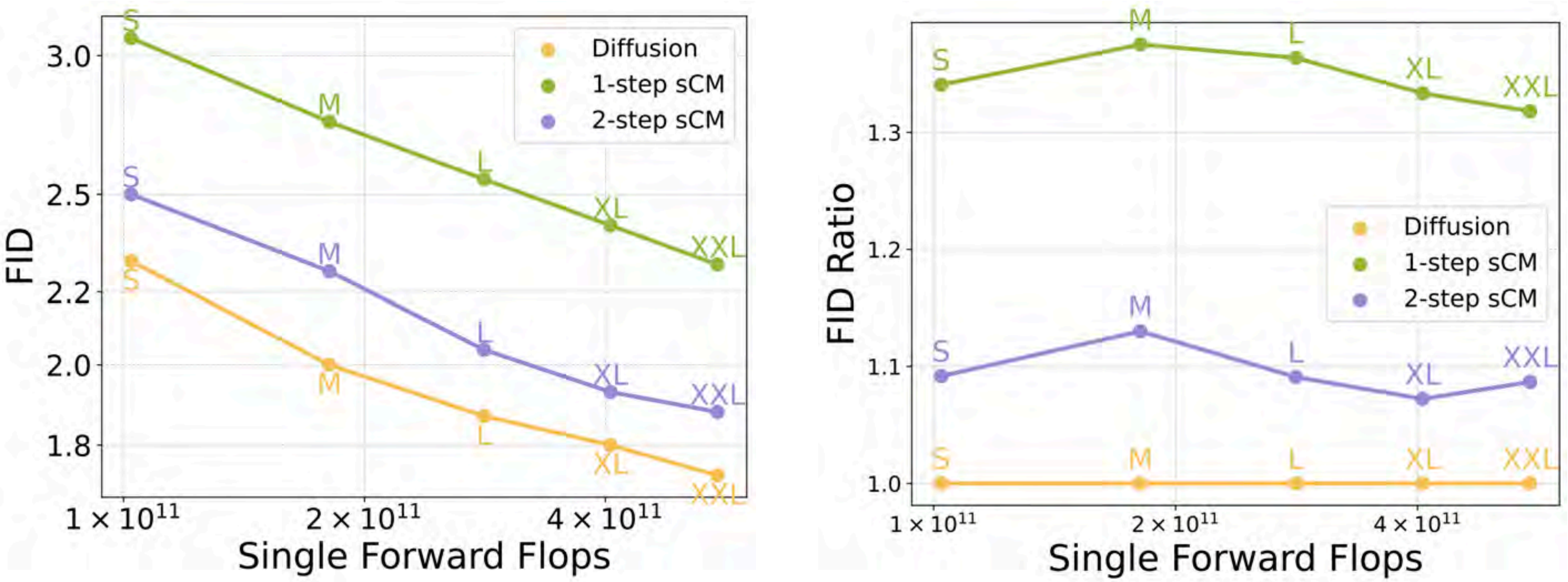
Comparable quality with 1/10 effective sampling compute



Selected 2-step samples on ImageNet 512x512

sCM scales commensurately with teacher diffusion models

Same scaling property!



Quantitative Results on ImageNet 64x64

Class-Conditional ImageNet 64×64

METHOD	NFE (↓)	FID (↓)
Diffusion Distillation		
DFNO (LPIPS) (Zheng et al., 2023b)	1	7.83
PID (LPIPS) (Tee et al., 2024)	1	9.49
TRACT (Berthelot et al., 2023)	1	7.43
	2	4.97
PD (Salimans & Ho, 2022)	1	10.70
(reimpl. from Heek et al. (2024))	2	4.70
CD (LPIPS) (Song et al., 2023)	1	6.20
	2	4.70
MultiStep-CD (Heek et al., 2024)	1	3.20
	2	1.90
sCD (ours)	1	**2.44**
	2	**1.66**
Consistency Training		
iCT (Song & Dhariwal, 2023)	1	4.02
	2	3.20
iCT-deep (Song & Dhariwal, 2023)	1	3.25
	2	2.77
ECT (Geng et al., 2024)	1	2.49
	2	1.67
sCT (ours)	1	**2.04**
	2	**1.48**

Quantitative Results on ImageNet 512x512

METHOD	NFE (↓)	FID (↓)	#Params
Diffusion models			
ADM-G (Dhariwal & Nichol, 2021)	250×2	7.72	559M
RIN (Jabri et al., 2022)	1000	3.95	320M
U-ViT-H/4 (Bao et al., 2023)	250×2	4.05	501M
DiT-XL/2 (Peebles & Xie, 2023)	250×2	3.04	675M
SimDiff (Hoogeboom et al., 2023)	512×2	3.02	2B
VDM++ (Kingma & Gao, 2024)	512×2	2.65	2B
DiffiT (Hatamizadeh et al., 2023)	250×2	2.67	561M
DiMR-XL/3R (Liu et al., 2024)	250×2	2.89	525M
DiffuSSM-XL (Yan et al., 2024)	250×2	3.41	673M
DiM-H (Teng et al., 2024)	250×2	3.78	860M
U-DiT (Tian et al., 2024b)	250	15.39	204M
SiT-XL (Ma et al., 2024)	250×2	2.62	675M
Large-DiT (Alpha-VLLM, 2024)	250×2	2.52	3B
MaskDiT (Zheng et al., 2023a)	79×2	2.50	736M
DiS-H/2 (Fei et al., 2024a)	250×2	2.88	900M
DRWKV-H/2 (Fei et al., 2024b)	250×2	2.95	779M
EDM2-S (Karras et al., 2024)	63×2	2.23	280M
EDM2-M (Karras et al., 2024)	63×2	2.01	498M
EDM2-L (Karras et al., 2024)	63×2	1.88	778M
EDM2-XL (Karras et al., 2024)	63×2	1.85	1.1B
EDM2-XXL (Karras et al., 2024)	63×2	**1.81**	1.5B
GANs & Masked Models			
BigGAN (Brock, 2018)	1	8.43	160M
StyleGAN-XL (Sauer et al., 2022)	1×2	2.41	168M
VQGAN (Esser et al., 2021)	1024	26.52	227M
MaskGIT (Chang et al., 2022)	12	7.32	227M
MAGVIT-v2 (Yu et al., 2023)	64×2	1.91	307M
MAR (Li et al., 2024)	64×2	**1.73**	481M
VAR-d36-s (Tian et al., 2024a)	10×2	2.63	2.3B

METHOD	NFE (↓)	FID (↓)	#Params
†Teacher Diffusion Model			
EDM2-S (Karras et al., 2024)	63×2	2.29	280M
EDM2-M (Karras et al., 2024)	63×2	2.00	498M
EDM2-L (Karras et al., 2024)	63×2	1.87	778M
EDM2-XL (Karras et al., 2024)	63×2	1.80	1.1B
EDM2-XXL (Karras et al., 2024)	63×2	1.73	1.5B
Consistency Training (sCT, ours)			
sCT-S (ours)	1	10.13	280M
	2	9.86	280M
sCT-M (ours)	1	5.84	498M
	2	5.53	498M
sCT-L (ours)	1	5.15	778M
	2	4.65	778M
sCT-XL (ours)	1	4.33	1.1B
	2	3.73	1.1B
sCT-XXL (ours)	1	4.29	1.5B
	2	3.76	1.5B
Consistency Distillation (sCD, ours)			
sCD-S	1	3.07	280M
	2	2.50	280M
sCD-M	1	2.75	498M
	2	2.26	498M
sCD-L	1	2.55	778M
	2	2.04	778M
sCD-XL	1	2.40	1.1B
	2	1.93	1.1B
sCD-XXL	1	**2.28**	1.5B
	2	**1.88**	1.5B

Summary

- Continuous-time CMs avoid the discretization error of PF-ODEs, producing better samples than discrete-time CMs.
- Always keep training stability in mind when designing scalable algorithms.
- Continuous-time CMs reduce the performance gap to SOTA diffusion models to within 10%, while achieving approximately a 50x speedup in sampling.