Consistency Models

Yang Song [1] **Prafulla Dhariwal** [1] **Mark Chen** [1] **Ilya Sutskever** [1]

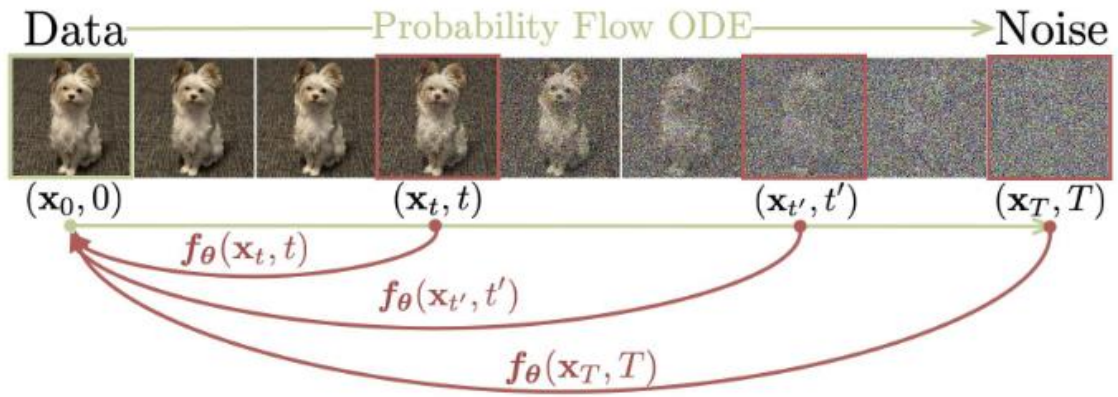
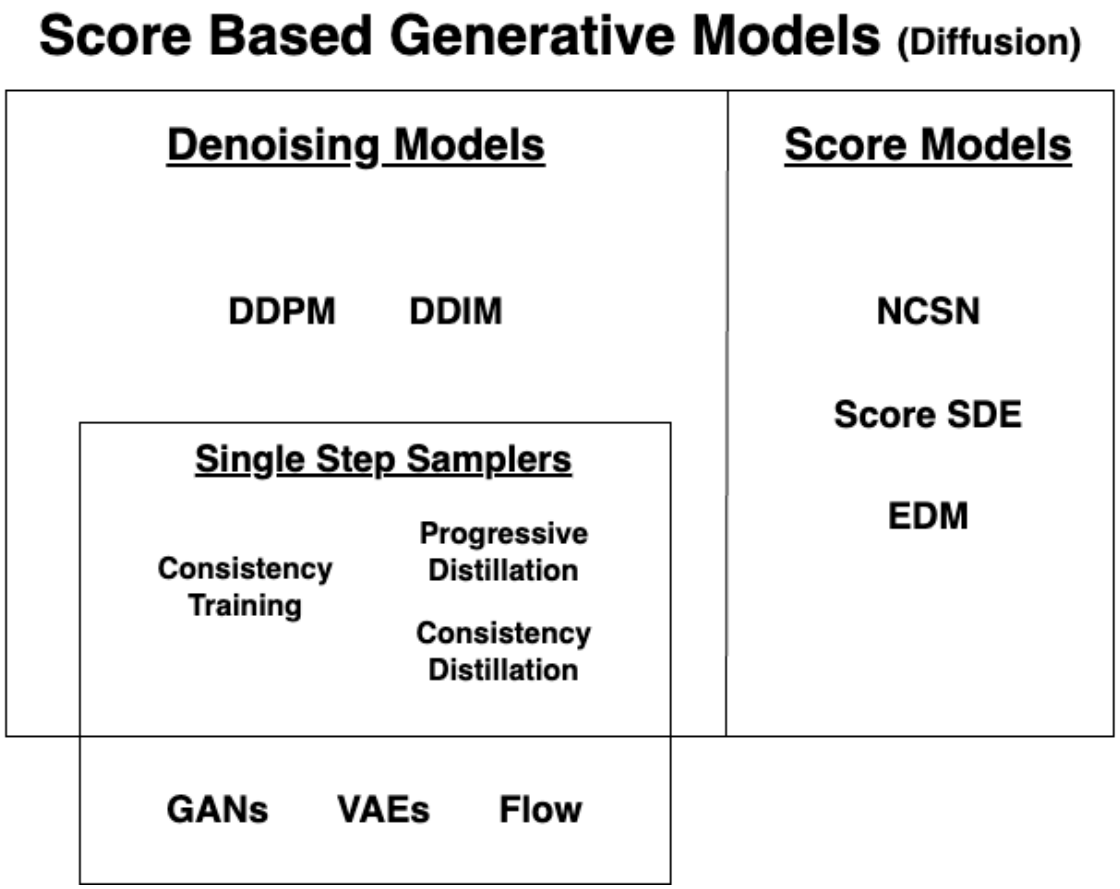
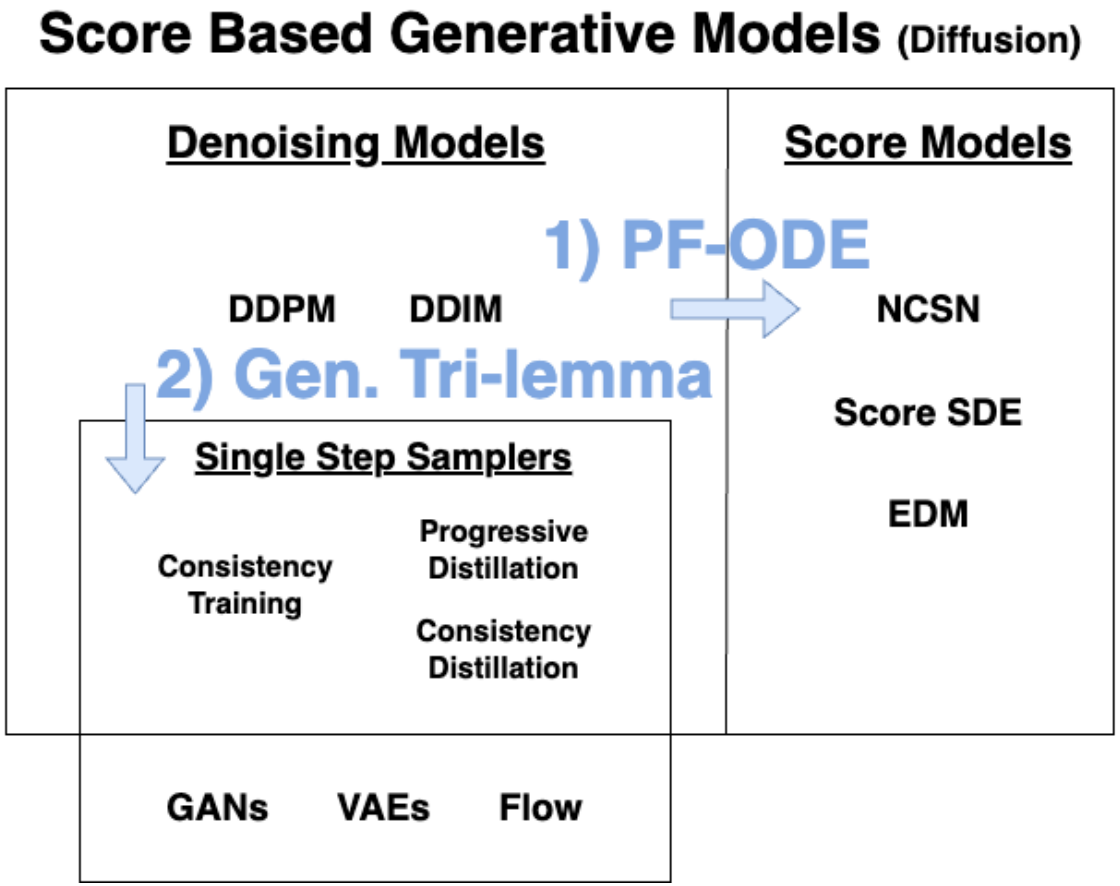


Table of Contents

Introduction



Introduction



Diffusion Models

Task

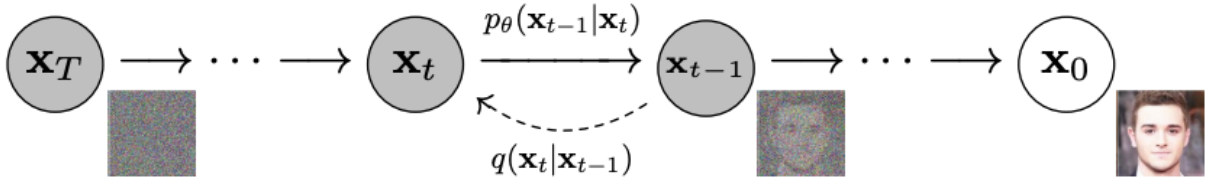


Figure 2: The directed graphical model considered in this work.

Model

$$f(x_{t_{n+1}}, t_{n+1}) \rightarrow x_{t_n}, x_T \rightarrow x_0$$

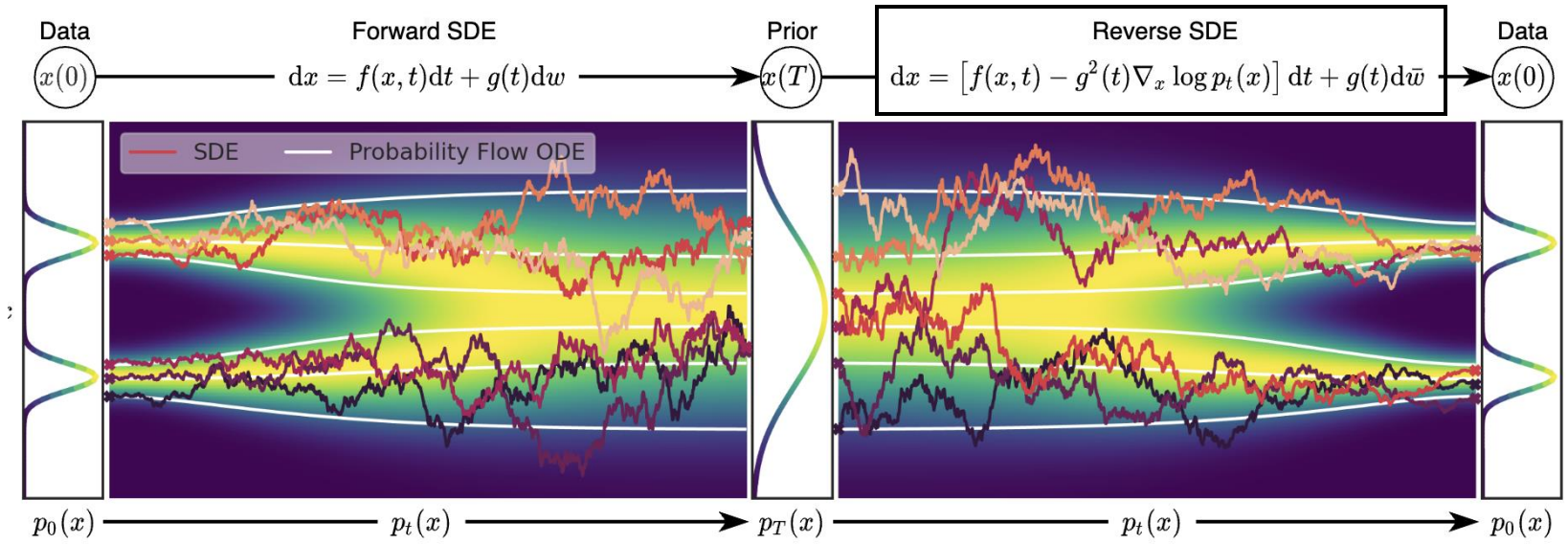
Parameterization

$$noise - predict, \qquad data - predict$$

Score Function and ODE

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

Denoising Diffusion



SDE

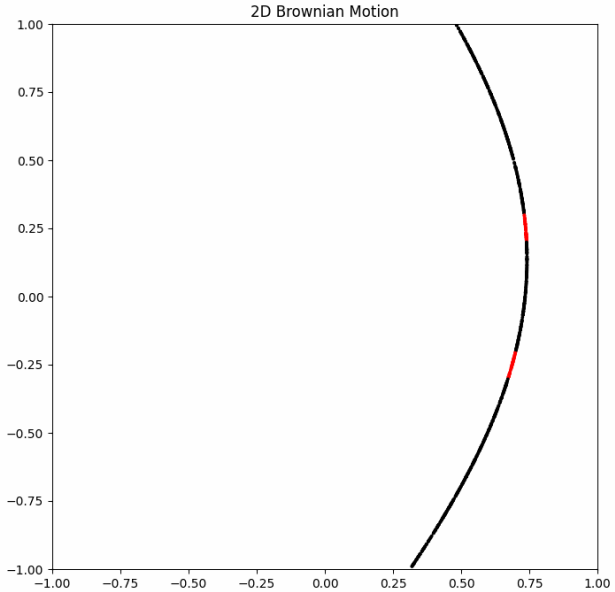
$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]\mathrm{d}t,$$

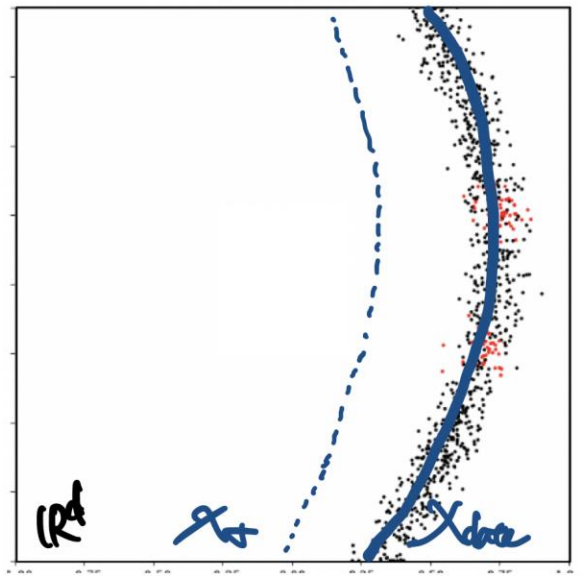
PF ODE

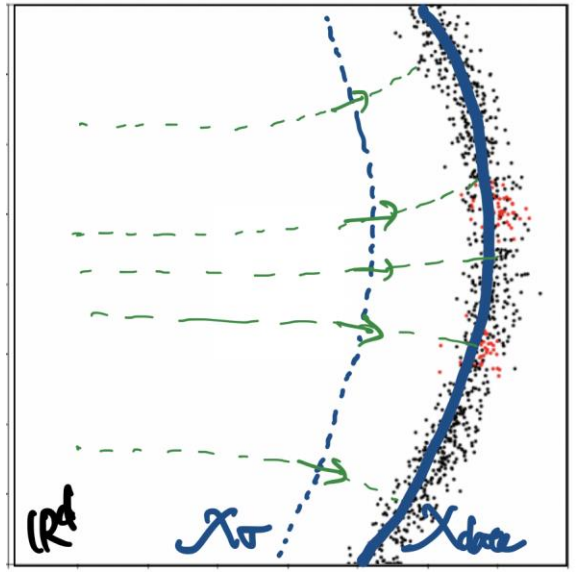
→ Score function

[1] Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

SDE vs. PF-ODE


2D Brownian Motion
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00





Score Models

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

$\rightarrow s_\theta\ (x, t)$

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\sigma} = -\sigma \nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) \quad \sigma \in [\sigma_{\min}, \sigma_{\max}],$$

$\rightarrow s_\theta\ (x, \sigma)$

Model

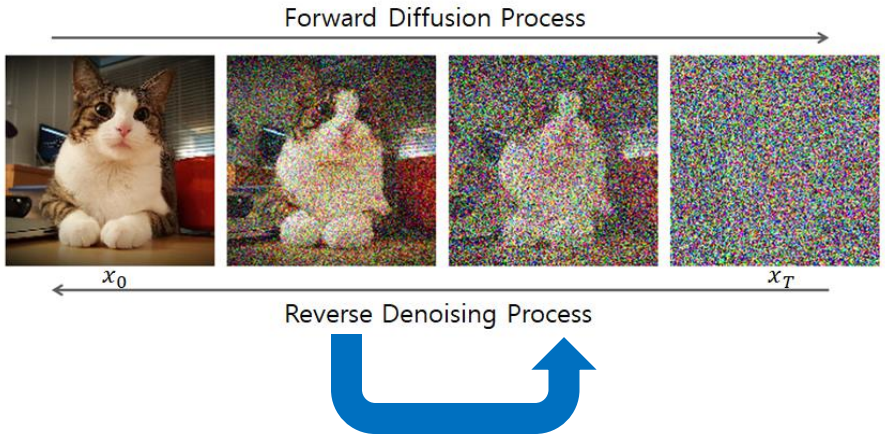
$f(x_{t_{n+1}}, t_{n+1}) \rightarrow\ x_{t_n}, x_T \rightarrow x_0$

Parameterization

$score\ function - predict$

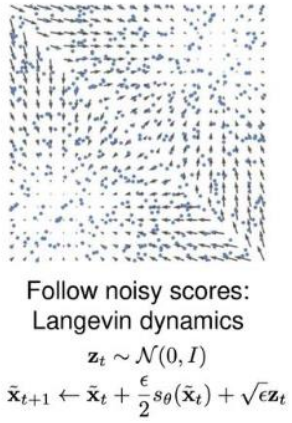
Denoising Process (Inference)

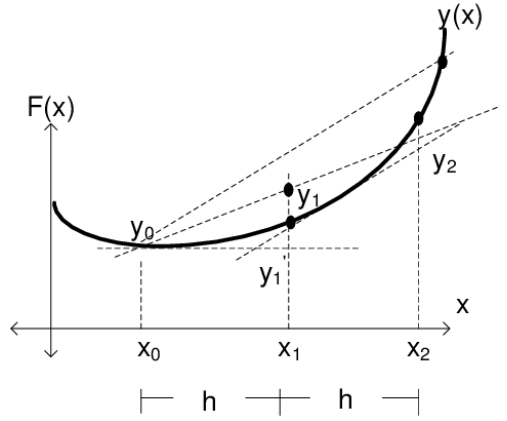
$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\sigma} = -\sigma \nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) \quad \sigma \in [\sigma_{\min}, \sigma_{\max}],$$



Direct Mapping

→ Empirical ODE solver

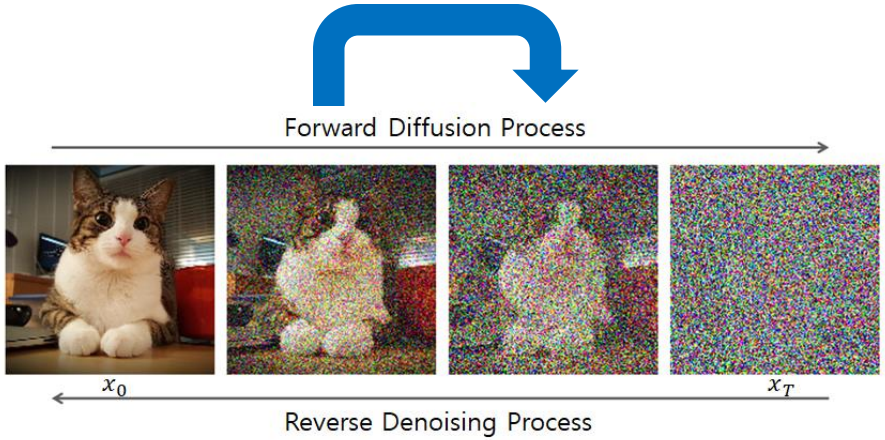


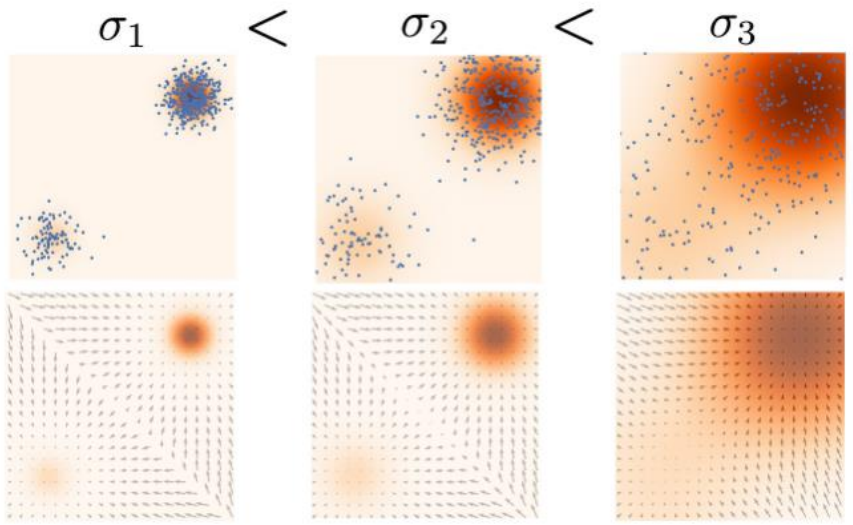


Langevin Dynamics
Euler / Heun Estimation

→ Numerical ODE solver

Noising Process



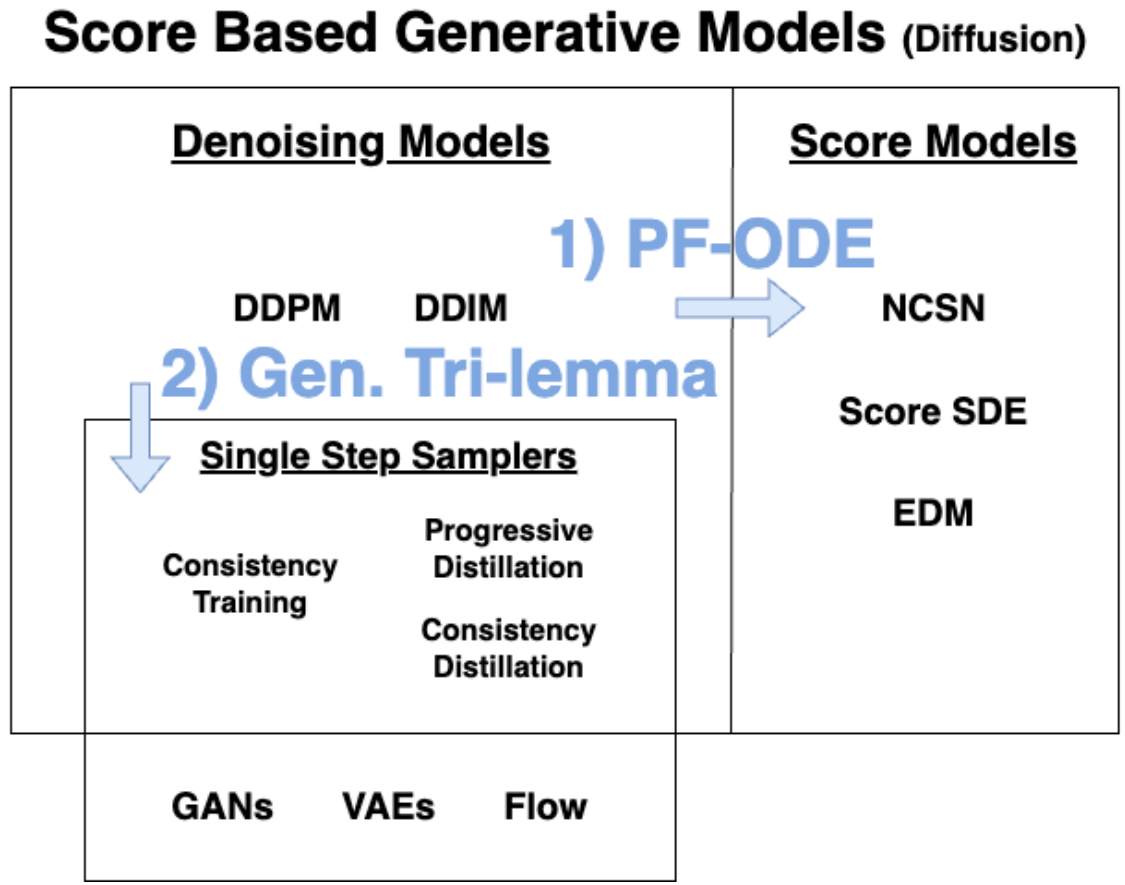


Data pairs for denoising training

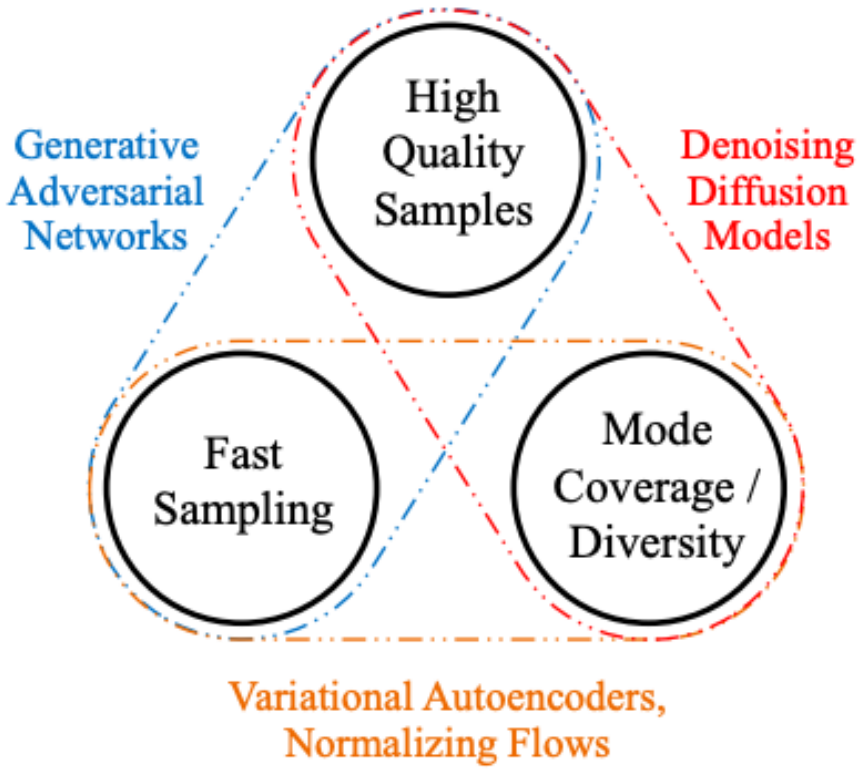
Data point sampling

$$X \sim f(x), x \in S, \boldsymbol{S} \subseteq \mathbb{R}^{\boldsymbol{d}}$$

[2] Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." *Advances in neural information processing systems* 32 (2019).


Score Based Generative Models (Diffusion)
Denoising Models
Score Models
1) PF-ODE
DDPM
DDIM
NCSN
2) Gen. Tri-lemma
Score SDE
EDM
Single Step Samplers
Progressive Distillation
Consistency Training
Consistency Distillation
GANs
VAEs
Flow

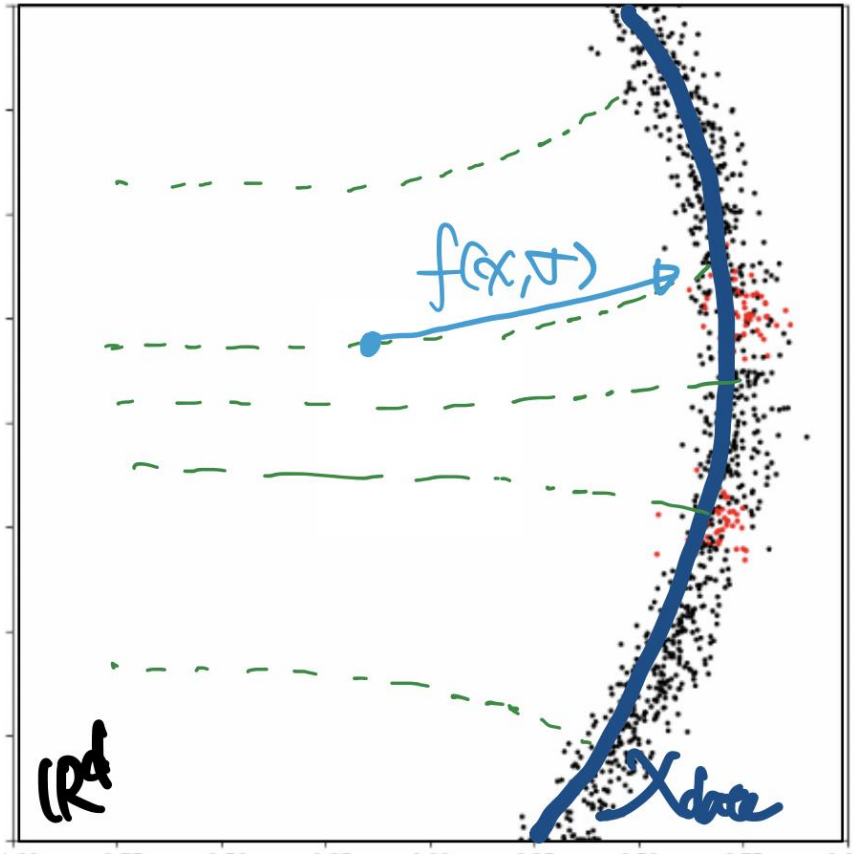
Generative Learning Trilemma



[3] Xiao, Zhisheng, Karsten Kreis, and Arash Vahdat. "Tackling the generative learning trilemma with denoising diffusion gans." *arXiv preprint arXiv:2112.07804* (2021).

Problem Definition

- Sustain Quality of diffusion models
  - PF–ODE solver
- Single-step sampling
  - Direct mapping, $z \rightarrow x$ $(\mathbb{R}^d \rightarrow S, S \subseteq \mathbb{R}^d)$



Consistency Function

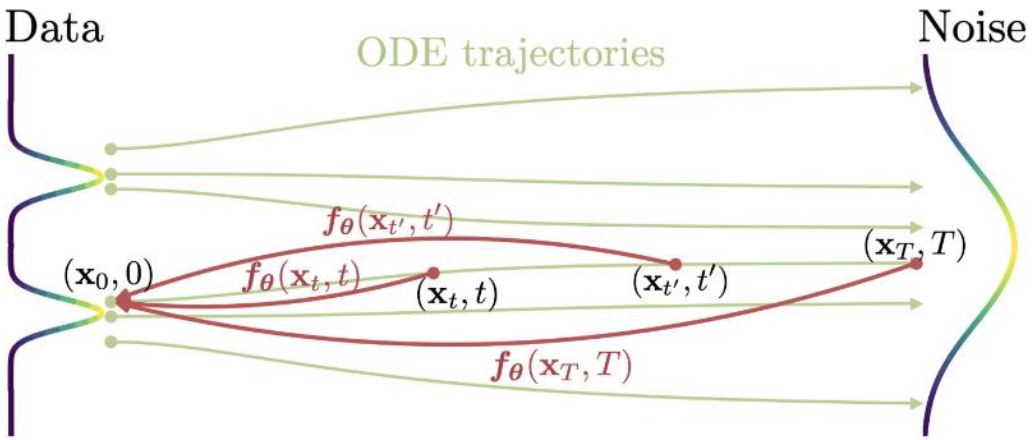


Figure 2: Consistency models are trained to map points on any trajectory of the PF ODE to the trajectory's origin.

Consistency function,

$$f : (x_t, t) \rightarrow x_\epsilon$$

must be self-consistent,

for any solution trajectory $\{x_t\}_{t\in[\epsilon,T]}$ on true PF ODE,

$$f(x_t, t) = f(x_{t'}, t'), \forall t, t' \in [\epsilon, T]$$

→ Approximation models can be trained via

1) CD (Consistency Distillation), 2) CT (Consistency Training)

Consistency Distillation

Algorithm 2 Consistency Distillation (CD)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Phi(\cdot,\cdot;\boldsymbol{\phi})$, $d(\cdot,\cdot)$, $\lambda(\cdot)$, and μ

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$

repeat

Sample $\mathbf{x} \sim \mathcal{D}$ and $n \sim \mathcal{U}[\![1, N-1]\!]$

Sample $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 \boldsymbol{I})$

$\hat{\mathbf{x}}^{\boldsymbol{\phi}}_{t_n} \leftarrow \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}; \boldsymbol{\phi})$

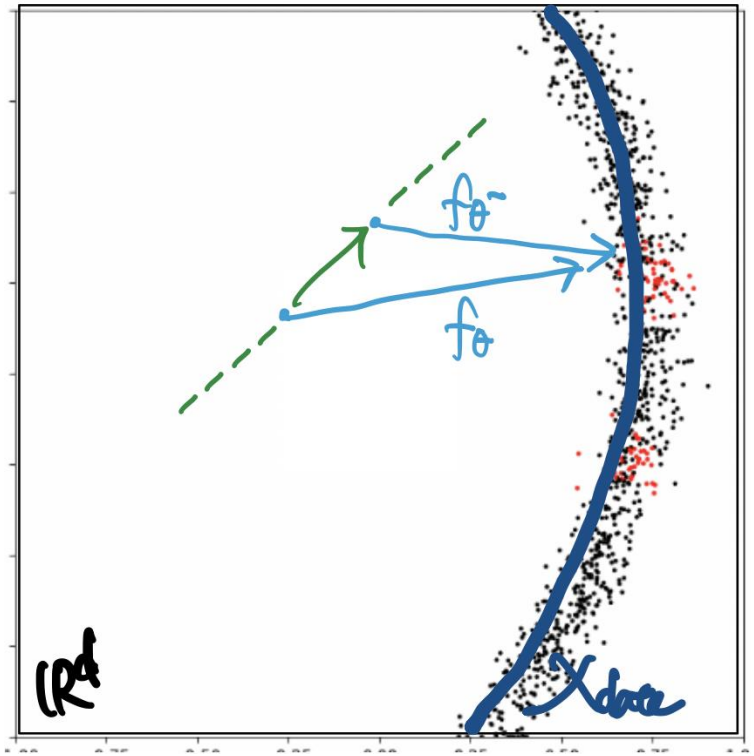
$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) \leftarrow$

$\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}^{\boldsymbol{\phi}}_{t_n}, t_n))$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi})$

$\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu \boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$

until convergence



Analogy to Contrastive Learning

- *Distillation* from Φ^{φ}
- *Data pair*: $\left(\widehat{x_{t_n}}^{\varphi},\ t_n\right) - \left(x_{t_{n+1}}, t_{n+1}\right)$
- Student: $\theta, x_{t_{n+1}}$, Teacher: θ^{-}, x_{t_n}
- Boundary condition $f(x_\epsilon, \varepsilon) = x_\epsilon$

$$\mathcal{L}_{CD}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \boldsymbol{\phi}) := \mathbb{E}[\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}, t_n))], \quad (7)$$

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t) F_{\boldsymbol{\theta}}(\mathbf{x}, t), \quad (5)$$

$$c_{\text{skip}}(t) = \frac{\sigma_{\text{data}}^2}{(t-\epsilon)^2 + \sigma_{\text{data}}^2}, \quad c_{\text{out}}(t) = \frac{\sigma_{\text{data}}(t-\epsilon)}{\sqrt{\sigma_{\text{data}}^2 + t^2}},$$

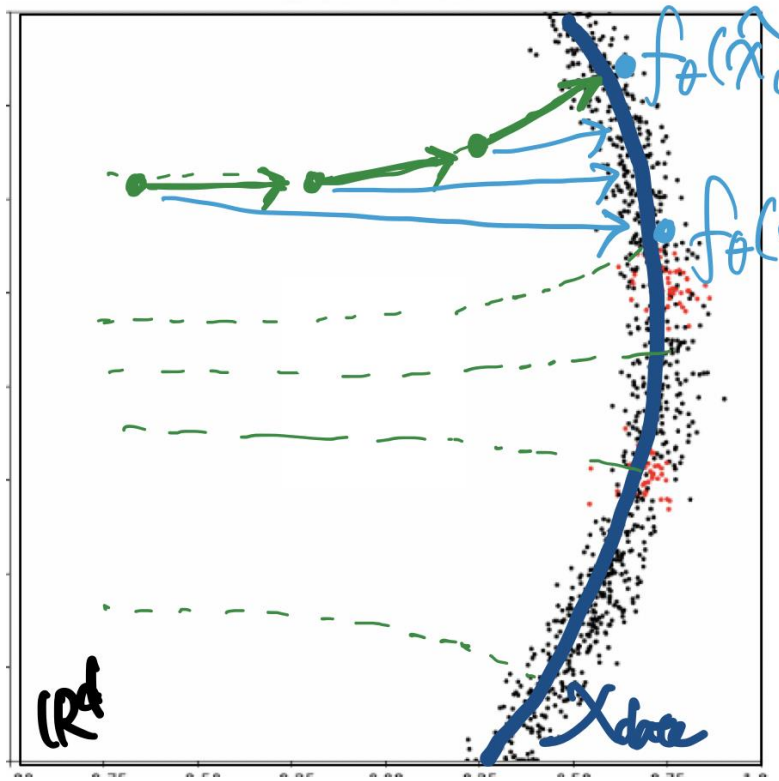
Why “Distillation”?

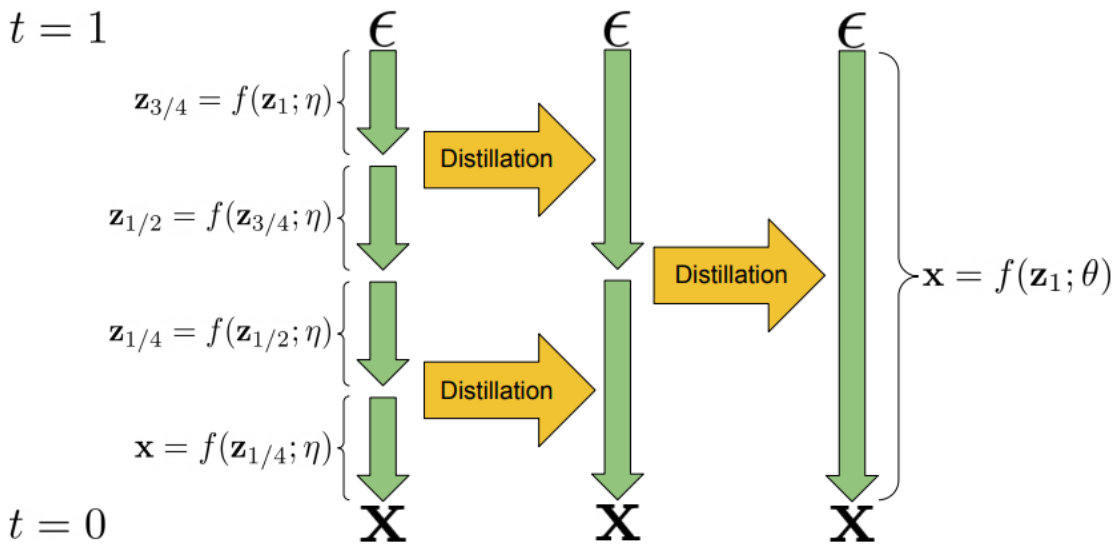
- Pretrained ODE solver for data pairs
- Loss function is bound by the ODE solver
- Model trains to be identical to ODE at $t = 1$

$$\lim_{iter \to \infty} f_\theta \neq f(x,t), rather\ f(x,t;\varphi)$$

Theorem 1. *Let* $\Delta t := \max_{n \in [\![1, N-1]\!]} \{|t_{n+1} - t_n|\}$, *and* $\boldsymbol{f}(\cdot, \cdot; \boldsymbol{\phi})$ *be the consistency function of the empirical PF ODE in Eq. (3). Assume* $\boldsymbol{f_\theta}$ *satisfies the Lipschitz condition: there exists* $L > 0$ *such that for all* $t \in [\epsilon, T]$, $\mathbf{x}$, *and* $\mathbf{y}$, *we have* $\|\boldsymbol{f_\theta}(\mathbf{x}, t) - \boldsymbol{f_\theta}(\mathbf{y}, t)\|_2 \leqslant L \|\mathbf{x} - \mathbf{y}\|_2$. *Assume further that for all* $n \in [\![1, N-1]\!]$, *the ODE solver called at* t_{n+1} *has local error uniformly bounded by* $O((t_{n+1} - t_n)^{p+1})$ *with* $p \geqslant 1$. *Then, if* $\mathcal{L}_{CD}^N(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) = 0$, *we have*

$$\sup_{n, \mathbf{x}} \|\boldsymbol{f_\theta}(\mathbf{x}, t_n) - \boldsymbol{f}(\mathbf{x}, t_n; \boldsymbol{\phi})\|_2 = O((\Delta t)^p).$$





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

Consistency Training

Algorithm 3 Consistency Training (CT)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, step schedule $N(\cdot)$, EMA decay rate schedule $\mu(\cdot)$, $d(\cdot,\cdot)$, and $\lambda(\cdot)$

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$ and $k \leftarrow 0$

repeat

  Sample $\mathbf{x} \sim \mathcal{D}$, and $n \sim \mathcal{U}[\![1, N(k)-1]\!]$

  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \leftarrow$

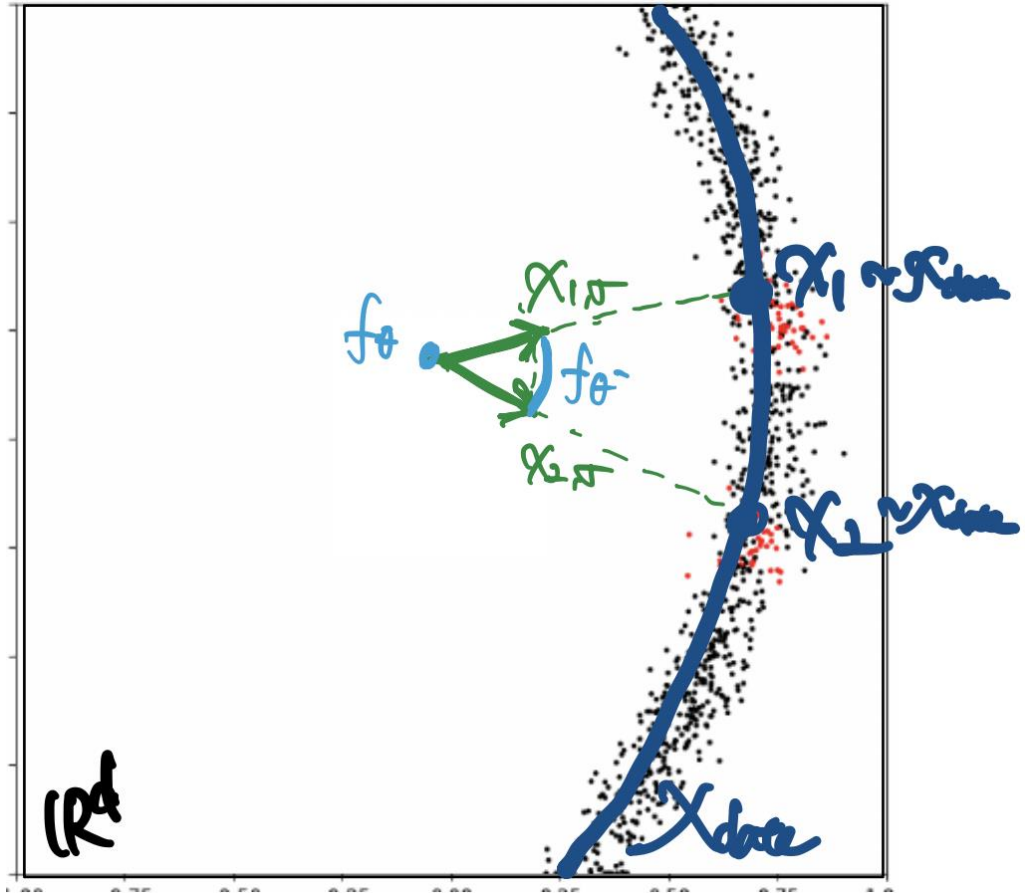
    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x} + t_{n+1}\mathbf{z}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x} + t_n \mathbf{z}, t_n))$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$

  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu(k)\boldsymbol{\theta}^- + (1-\mu(k))\boldsymbol{\theta})$

  $k \leftarrow k+1$

until convergence



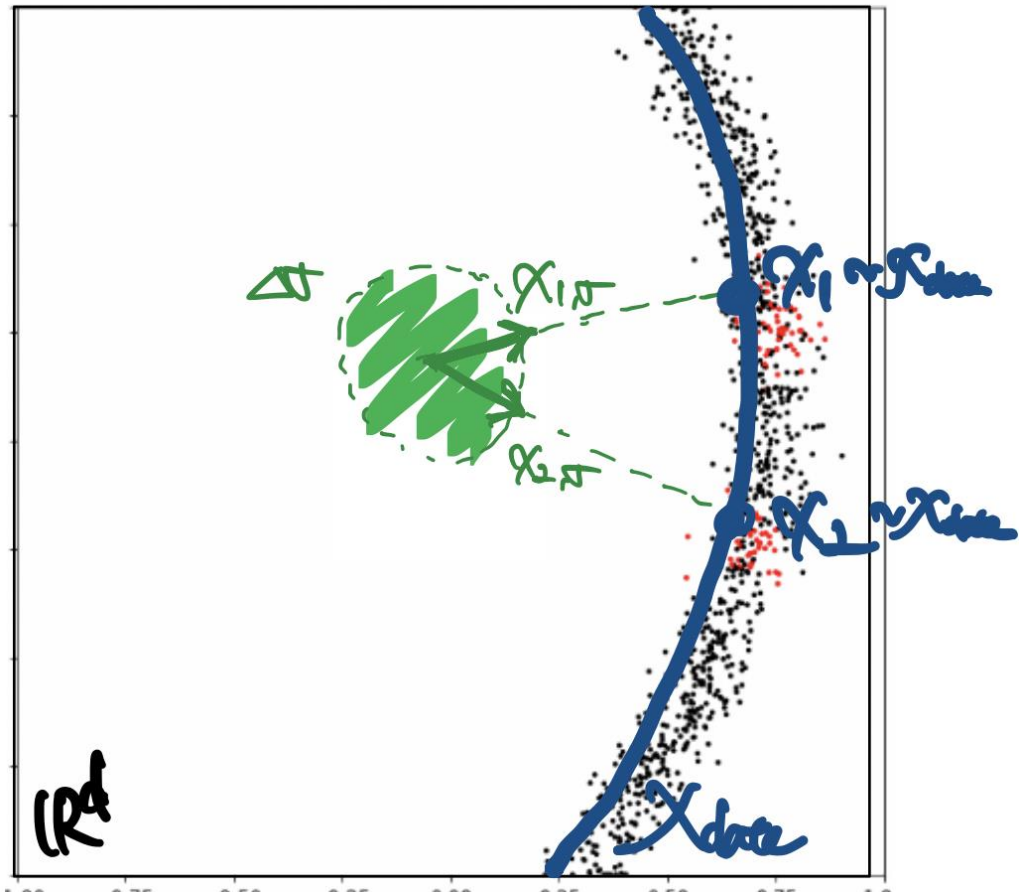
Theorem 2. *Let* $\Delta t := \max_{n \in [\![1, N-1]\!]} \{|t_{n+1} - t_n|\}$. *Assume* d *and* $\boldsymbol{f}_{\boldsymbol{\theta}^-}$ *are both twice continuously differentiable with bounded second derivatives, the weighting function* $\lambda(\cdot)$ *is bounded, and* $\mathbb{E}[\|\nabla \log p_{t_n}(\mathbf{x}_{t_n})\|_2^2] < \infty$. *Assume further that we use the Euler ODE solver, and the pre-trained score model matches the ground truth,* i.e., $\forall t \in [\epsilon, T] : \boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}, t) \equiv \nabla \log p_t(\mathbf{x})$. *Then,*

$$\mathcal{L}_{CD}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) = \mathcal{L}_{CT}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-) + o(\Delta t), \tag{9}$$

where the expectation is taken with respect to $\mathbf{x} \sim p_{data}$, $n \sim \mathcal{U}[\![1, N-1]\!]$, *and* $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 \boldsymbol{I})$. *The consistency training objective, denoted by* $\mathcal{L}_{CT}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$, *is defined as*

$$\mathbb{E}[\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x} + t_{n+1}\mathbf{z}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x} + t_n \mathbf{z}, t_n))], \tag{10}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. *Moreover,* $\mathcal{L}_{CT}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \geqslant O(\Delta t)$ *if* $\inf_N \mathcal{L}_{CD}^N(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) > 0$.



$$N(k) = \left\lceil \sqrt{\frac{k}{K}((s_1+1)^2 - s_0^2) + s_0^2} - 1 \right\rceil + 1$$

$$\mu(k) = \exp\left(\frac{s_0 \log \mu_0}{N(k)}\right),$$

Metrics

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**

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

METHOD	NFE (↓)	FID (↓)	IS (↑)
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**

Metrics

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**

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

METHOD	NFE (↓)	FID (↓)	IS (↑)
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**

Metrics

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**

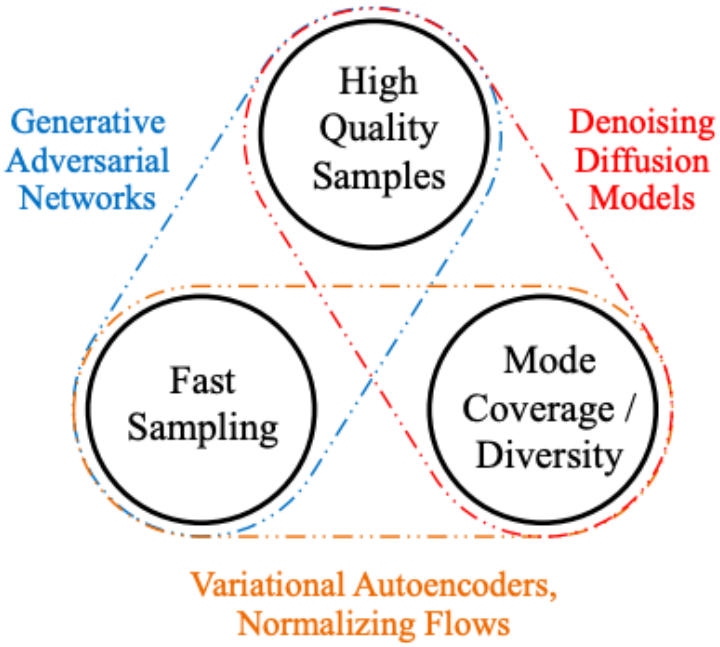
Table 1: Sample quality on CIFAR-10. *Methods that require synthetic data construction for distillation.

METHOD	NFE (↓)	FID (↓)	IS (↑)
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**

Implications and Applications

- Consistency Distillation
  - Valid distillation method for score-based models
    - Enable single-step sampling
    - Sustain sample quality
  - Dependency on pre-trained ODE solvers

- Consistency Training
  - Direct generation, without dependency
  - Relatively poor generation performance



Characteristics from Diffusion

- Semantic Continuity
  - Multistep sampling, Inpainting, Interpolation, Denoising, Upscaling…
- Controlled Data Generation
  - Classifier Free Guidance *(LCM)*

Algorithm 1 Multistep Consistency Sampling

Input: Consistency model $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot,\cdot)$, sequence of time points $\tau_1 > \tau_2 > \cdots > \tau_{N-1}$, initial noise $\hat{\mathbf{x}}_T$

$\mathbf{x} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_T, T)$

for $n = 1$ **to** $N-1$ **do**

  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

  $\hat{\mathbf{x}}_{\tau_n} \leftarrow \mathbf{x} + \sqrt{\tau_n^2 - \epsilon^2}\mathbf{z}$

  $\mathbf{x} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{\tau_n}, \tau_n)$

end for

Output: $\mathbf{x}$

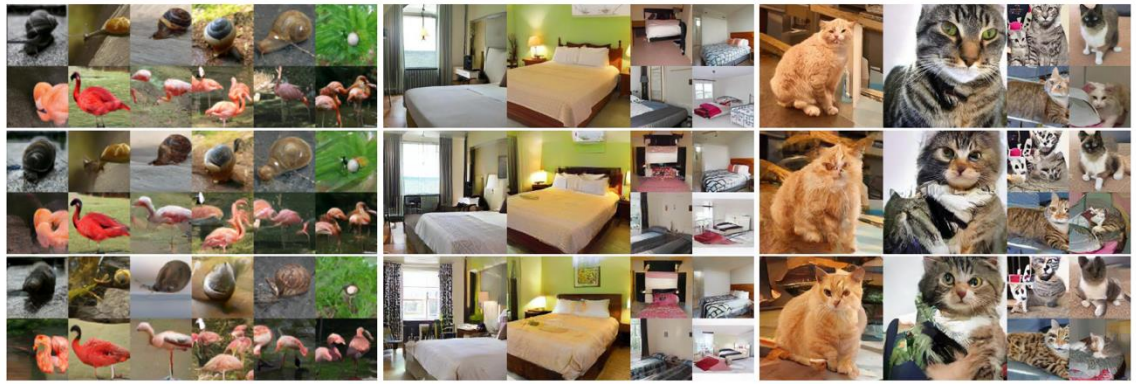
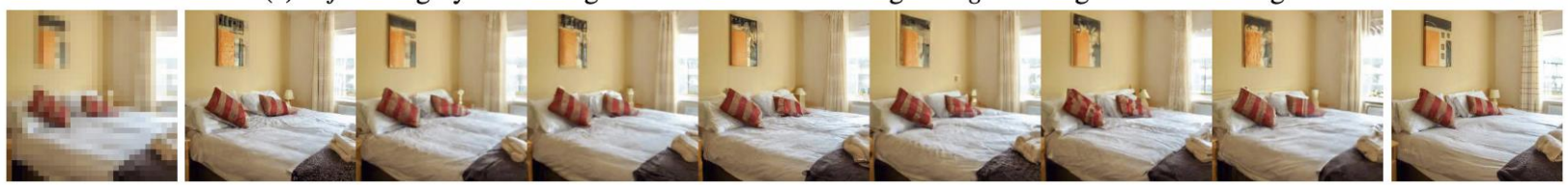


Figure 5: Samples generated by EDM (*top*), CT + single-step generation (*middle*), and CT + 2-step generation (*Bottom*). All corresponding images are generated from the same initial noise.

Algorithm 4 Zero-Shot Image Editing

Input: Consistency model $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot,\cdot)$, sequence of time points $t_1 > t_2 > \cdots > t_N$, reference image $\mathbf{y}$, invertible linear transformation $\boldsymbol{A}$, and binary image mask $\boldsymbol{\Omega}$
$\mathbf{y} \leftarrow \boldsymbol{A}^{-1}[(\boldsymbol{A}\mathbf{y}) \odot (1 - \boldsymbol{\Omega}) + \mathbf{0} \odot \boldsymbol{\Omega}]$
Sample $\mathbf{x} \sim \mathcal{N}(\mathbf{y}, t_1^2 \boldsymbol{I})$
$\mathbf{x} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t_1)$
$\mathbf{x} \leftarrow \boldsymbol{A}^{-1}[(\boldsymbol{A}\mathbf{y}) \odot (1 - \boldsymbol{\Omega}) + (\boldsymbol{A}\mathbf{x}) \odot \boldsymbol{\Omega}]$
for $n = 2$ **to** N **do**
    Sample $\mathbf{x} \sim \mathcal{N}(\mathbf{x}, (t_n^2 - \epsilon^2)\boldsymbol{I})$
    $\mathbf{x} \leftarrow \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t_n)$
    $\mathbf{x} \leftarrow \boldsymbol{A}^{-1}[(\boldsymbol{A}\mathbf{y}) \odot (1 - \boldsymbol{\Omega}) + (\boldsymbol{A}\mathbf{x}) \odot \boldsymbol{\Omega}]$
end for
Output: $\mathbf{x}$

(a) *Left*: The gray-scale image. *Middle*: Colorized images. *Right*: The ground-truth image.



(b) *Left*: The downsampled image (32 × 32). *Middle*: Full resolution images (256 × 256). *Right*: The ground-truth image (256 × 256).

(c) *Left*: A stroke input provided by users. *Right*: Stroke-guided image generation.

IMPROVED TECHNIQUES FOR TRAINING CONSISTENCY MODELS

Yang Song & Prafulla Dhariwal
OpenAI
{songyang,prafulla}@openai.com

Table 2: Comparing the quality of unconditional samples on CIFAR-10.

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

Table 1: Comparing the design choices for CT in Song et al. (2023) versus our modifications.

	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	$d(\boldsymbol{x}, \boldsymbol{y}) = \text{LPIPS}(\boldsymbol{x}, \boldsymbol{y})$	$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + c^2} - c$
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	σ_i, where $i \sim \mathcal{U}[\![1, N(k)-1]\!]$	σ_i, where $i \sim p(i)$, and $p(i) \propto$ $\text{erf}\left(\frac{\log(\sigma_{i+1}) - P_{\text{mean}}}{\sqrt{2} P_{\text{std}}}\right) - \text{erf}\left(\frac{\log(\sigma_i) - P_{\text{mean}}}{\sqrt{2} P_{\text{std}}}\right)$
Weighting function	$\lambda(\sigma_i) = 1$	$\lambda(\sigma_i) = \frac{1}{\sigma_{i+1} - \sigma_i}$
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 $\sigma_i = (\sigma_{\min}^{1/\rho} + \frac{i-1}{N(k)-1}(\sigma_{\max}^{1/\rho} - \sigma_{\min}^{1/\rho}))^\rho$, where $i \in [\![1, N(k)]\!], \rho = 7, \sigma_{\min} = 0.002, \sigma_{\max} = 80$	

[5] Song, Yang, and Prafulla Dhariwal. "Improved techniques for training consistency models." *arXiv preprint arXiv:2310.14189* (2023).

- Smaller loss weighting on higher noise level
- Smaller noise level embedding
- Dropout

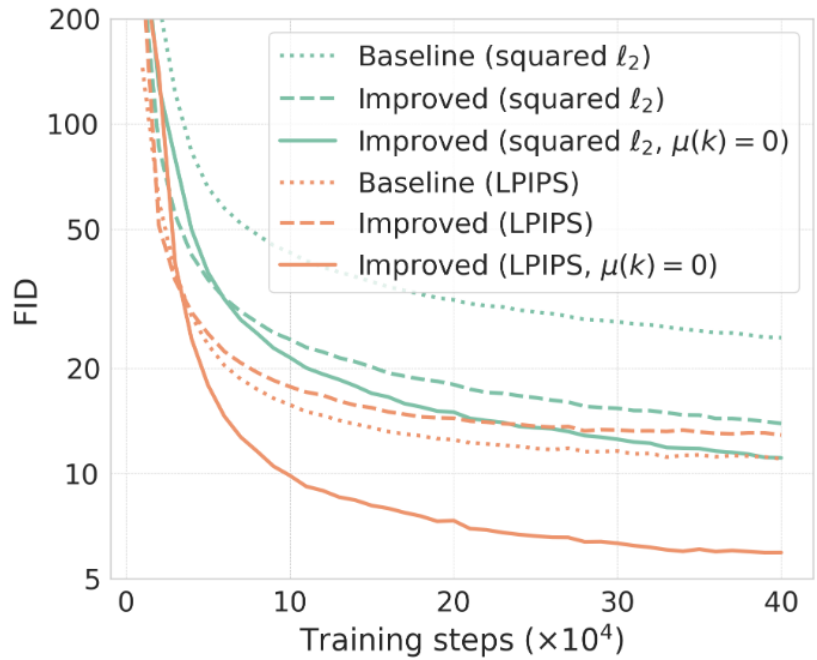
- Discard EMA
- Pseudo-Huber Loss

- Discretization step scheduling
- Noise sampling scheduling

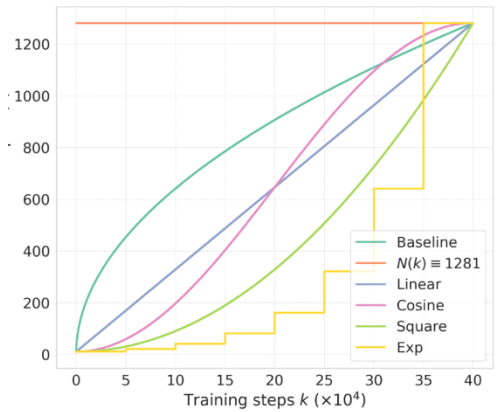
Proposition 1. *Given the notations introduced earlier, and using the uniform weighting function* $\lambda(\sigma) = 1$ *along with the squared* ℓ_2 *metric, we have*

$$\lim_{N\to\infty} \mathcal{L}^N(\theta,\theta^-) = \lim_{N\to\infty} \mathcal{L}^N_{CT}(\theta,\theta^-) = \mathbb{E}\Big[\big(1-\frac{\sigma_{min}}{\sigma_i}\big)^2(\theta-\theta^-)^2\Big] \quad \text{if } \theta^- \neq \theta \tag{6}$$

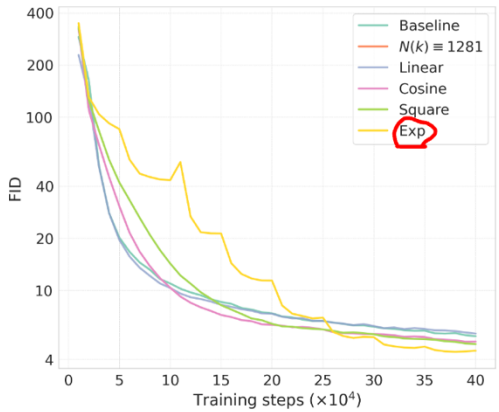
$$\lim_{N\to\infty} \frac{1}{\Delta\sigma}\frac{\mathrm{d}\mathcal{L}^N(\theta,\theta^-)}{\mathrm{d}\theta} = \begin{cases} \frac{\mathrm{d}}{\mathrm{d}\theta}\mathbb{E}\Big[\frac{\sigma_{min}}{\sigma_i^2}\Big(1-\frac{\sigma_{min}}{\sigma_i}\Big)(\theta-\xi)^2\Big], & \theta^- = \theta \\ +\infty, & \theta^- < \theta \\ -\infty, & \theta^- > \theta \end{cases} \tag{7}$$



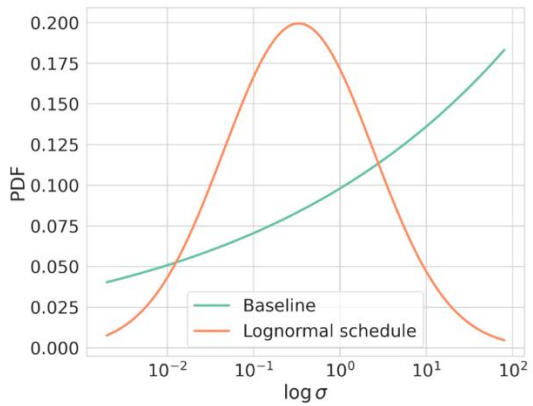
(a) LPIPS & squared ℓ_2 metrics.



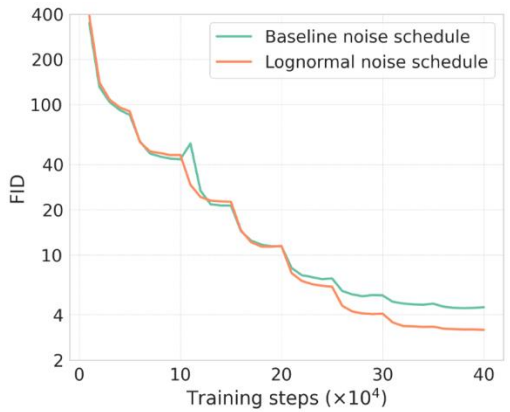
(b) Various curriculums for $N(k)$.



(c) FIDs vs. $N(k)$ curriculums.



(a) PDF of $\log \sigma$



(b) Lognormal vs. default schedules.

$$N(k) = \min(s_0 2^{\lfloor \frac{k}{K'} \rfloor}, s_1) + 1,$$
$$\text{where } K' = \left\lfloor \frac{K}{\log_2 \lfloor s_1/s_0 \rfloor + 1} \right\rfloor$$

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

Latent Consistency Models: Synthesizing High-Resolution Images with Few-step Inference

Simian Luo* **Yiqin Tan*** **Longbo Huang†** **Jian Li†** **Hang Zhao†**
Institute for Interdisciplinary Information Sciences, Tsinghua University
{luosm22, tyq22}@mails.tsinghua.edu.cn
{longbohuang, lijian83, hangzhao}@tsinghua.edu.cn

$$\tilde{\epsilon}_\theta(z_t, w, c, t) = (1+w)\epsilon_\theta(z_t, c, t) - w\epsilon_\theta(z_t, \emptyset, t)$$

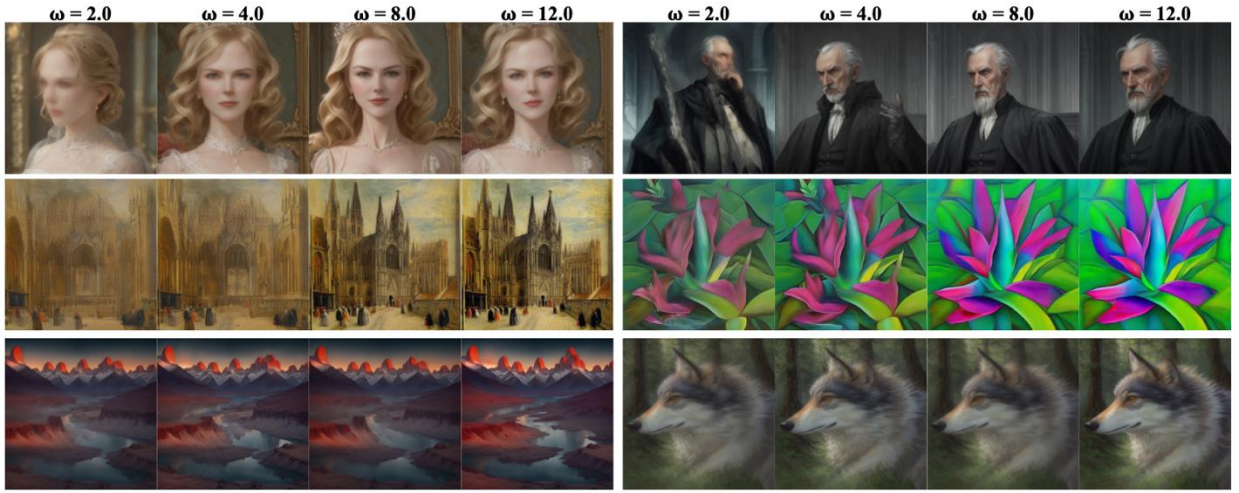
CFG — Conditional diffusion model — Unconditional diffusion model

PF ODE $$\frac{dx}{dt} = f(t)z_t + \frac{g^2(t)}{2\sigma_t}\epsilon_\theta(z_t, c, t)$$

Augmented PF ODE $$\frac{dx}{dt} = f(t)z_t + \frac{g^2(t)}{2\sigma_t}\tilde{\epsilon}_\theta(z_t, w, c, t)$$

CFG

$$\mathcal{L}_{\mathcal{CD}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Psi\right) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{c}, n}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+1}}, \boldsymbol{c}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{z}}^{\Psi}_{t_n}, \boldsymbol{c}, t_n)\right)\right]$$



→ Efficient sampling / High resolution / Text Conditioning

[6] Luo, Simian, et al. "Latent consistency models: Synthesizing high-resolution images with few-step inference." *arXiv preprint arXiv:2310.04378* (2023).

Algorithm 4 Latent Consistency Fine-tuning (LCF)

Input: customized dataset $\mathcal{D}^{(s)}$, pre-trained LCM parameter $\boldsymbol{\theta}$, learning rate η, distance metric $d(\cdot,\cdot)$, EMA rate μ, noise schedule $\alpha(t), \sigma(t)$, guidance scale $[w_{\min}, w_{\max}]$, skipping interval k, and encoder $E(\cdot)$

Encode training data into the latent space: $\mathcal{D}_z^{(s)} = \{(\boldsymbol{z}, \boldsymbol{c}) | \boldsymbol{z} = E(\boldsymbol{x}), (\boldsymbol{x}, \boldsymbol{c}) \in \mathcal{D}^{(s)}\}$

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$

repeat

Sample $(\boldsymbol{z}, \boldsymbol{c}) \sim \mathcal{D}_z^{(s)}$, $n \sim \mathcal{U}[1, N-k]$ and $w \sim [w_{\min}, w_{\max}]$

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

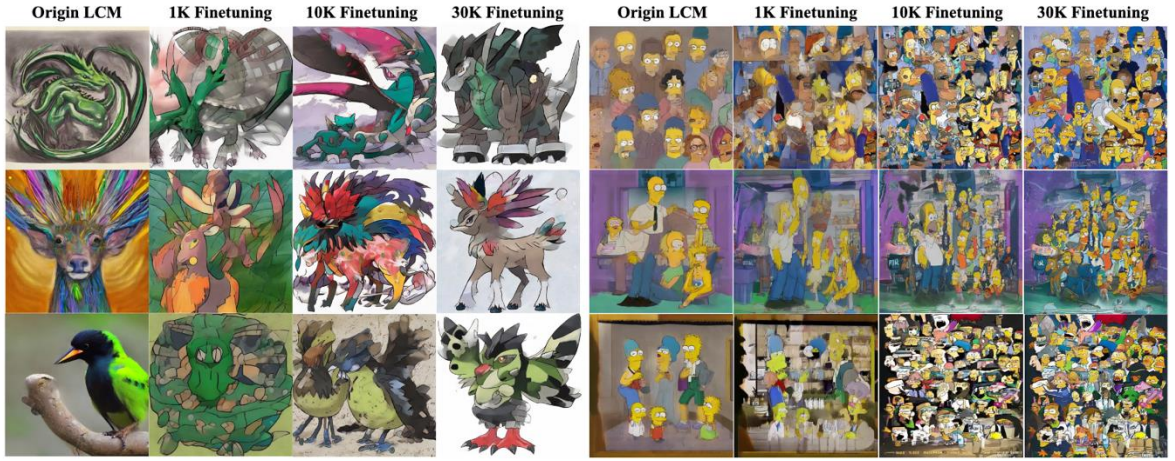
$\boldsymbol{z}_{t_{n+k}} \leftarrow \alpha(t_{n+k})\boldsymbol{z} + \sigma(t_{n+k})\boldsymbol{\epsilon} \quad , \quad \boldsymbol{z}_{t_n} \leftarrow \alpha(t_n)\boldsymbol{z} + \sigma(t_n)\boldsymbol{\epsilon}$

$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \leftarrow d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, \boldsymbol{c}, w), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\boldsymbol{z}_{t_n}, t_n, \boldsymbol{c}, w))$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$

$\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$

until convergence



Finetuning, CD → CT conversion