Diffusion Reading Group #18

Tanishq Abraham

3/11/2023

Diffusion models - refresher

Stochastic Differential Equation (SDE):

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

Probability Flow Ordinary Differential Equation (PF ODE):

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{\mu}(\mathbf{x}_t, t) - \frac{1}{2}\sigma(t)^2 \nabla \log p_t(\mathbf{x}_t)\right]\mathrm{d}t.$$

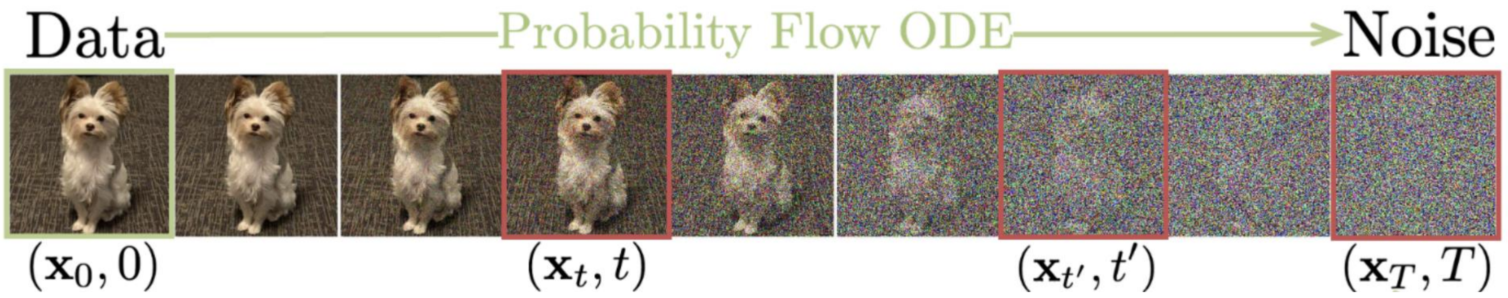
Diffusion models - refresher

Karras et al.:

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

Score function $\boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ obtained via denoising score matching

Use ODE solver to obtain solution trajectory, usually stop at $t = \epsilon$

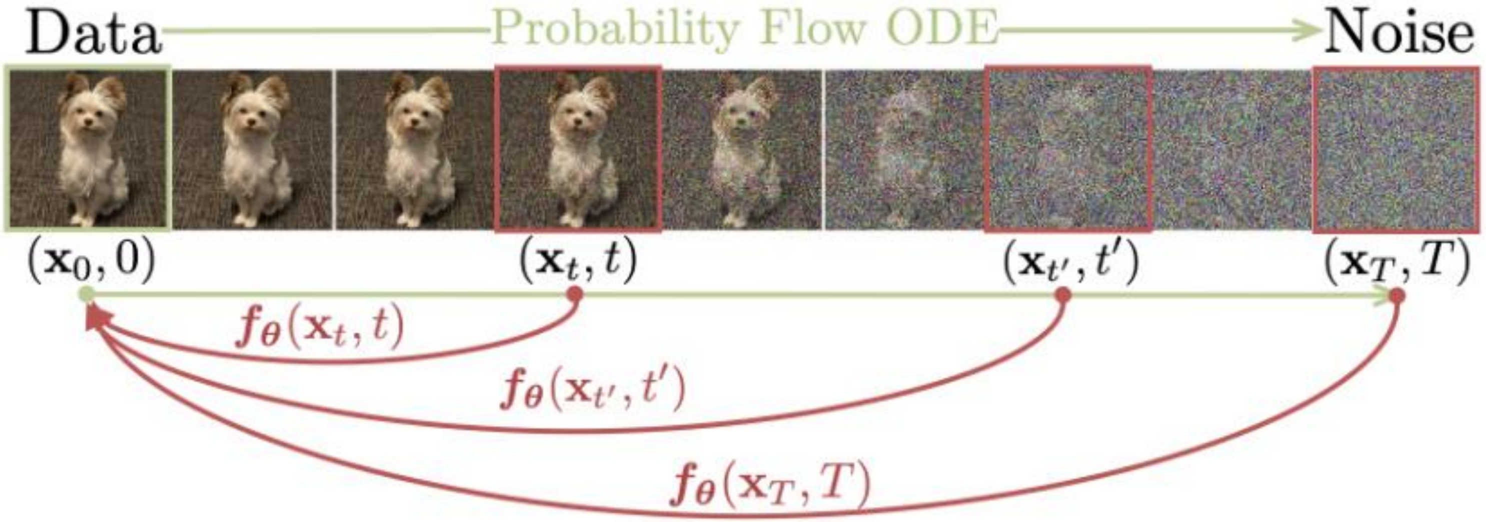


Consistency models

Define a consistency function $\boldsymbol{f} : (\mathbf{x}_t, t) \mapsto \mathbf{x}_\epsilon$ for any t

Therefore, $\boldsymbol{f}(\mathbf{x}_t, t) = \boldsymbol{f}(\mathbf{x}_{t'}, t')$ for all $t, t' \in [\epsilon, T]$

We train a neural network $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ by directly learning to enforce this self-consistency property



Consistency models - parameterization

Boundary condition: $\boldsymbol{f}(\mathbf{x}_\epsilon, \epsilon) = \mathbf{x}_\epsilon$ ($\boldsymbol{f}(\cdot, \epsilon)$ is the identity function)

The consistency model needs to be parameterized to respect this boundary condition

Option 1:

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) = \begin{cases} \mathbf{x} & t = \epsilon \\ F_{\boldsymbol{\theta}}(\mathbf{x}, t) & t \in (\epsilon, T] \end{cases}.$$

Option 2:

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

This is equivalent to Karras et al. preconditioning, this is used in the experiments.

Sampling

Single step sampling:

$$\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, T^2\boldsymbol{I})$$
$$\hat{\mathbf{x}}_\epsilon = \boldsymbol{f_\theta}(\mathbf{x}_T, T)$$

Multi-step sampling:

Algorithm 1 Multistep Consistency Sampling

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

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

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

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

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

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

end for

Output: $\mathbf{x}$

Consistency Distillation (CD)

Numerically solving ODE with Karras et al.-defined discretized timesteps:

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

where $\Phi(\cdots; \phi)$ is the update function of a one-step ODE solver applied to the PF ODE

Specifically using the Euler solver:

$$\hat{\mathbf{x}}_{t_n}^{\phi} = \mathbf{x}_{t_{n+1}} - (t_n - t_{n+1})t_{n+1}\boldsymbol{s}_{\phi}(\mathbf{x}_{t_{n+1}}, t_{n+1}).$$

Consistency Distillation (CD)

We can sample $\mathbf{x} \sim p_{\text{data}}$, then add Gaussian noise to $\mathbf{x}$

Given a datapoint $\mathbf{x}$, we sample from $\mathcal{N}(\mathbf{x}, t_{n+1}^2 \boldsymbol{I})$ to get $\mathbf{x}_{t_{n+1}}$, and take one discretization step of the numerical ODE solver (with the score function from our pretrained diffusion model) to get $\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}$

Our consistency model learns to enforce the consistency on the pair $\left(\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}, \mathbf{x}_{t_{n+1}}\right)$

Consistency Distillation (CD)

Loss function:

Definition 1. *The consistency distillation loss is defined as*

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

In practice $\lambda(t_n) = 1$ performs well. EMA + stopgrad improves training stability:

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

Consistency Distillation (CD)

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

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

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

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

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

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

until convergence

Consistency Training (CT)

A pretrained diffusion model was used to provide the score function in CD. We can instead directly estimate for training as the following is true:

$$\nabla \log p_t(\mathbf{x}_t) = -\mathbb{E}\left[\frac{\mathbf{x}_t - \mathbf{x}}{t^2} \,\middle|\, \mathbf{x}_t\right],$$

We can use this estimate of the score function in our loss function instead:

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

Consistency Training (CT)

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

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

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

Continuous-time objectives

Consistency Distillation (L2 loss, no stopgrad):

$$\mathcal{L}_{CD}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) = \mathbb{E}\left[\lambda(t)\left\|\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} - t\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}\boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)\right\|_2^2\right].$$

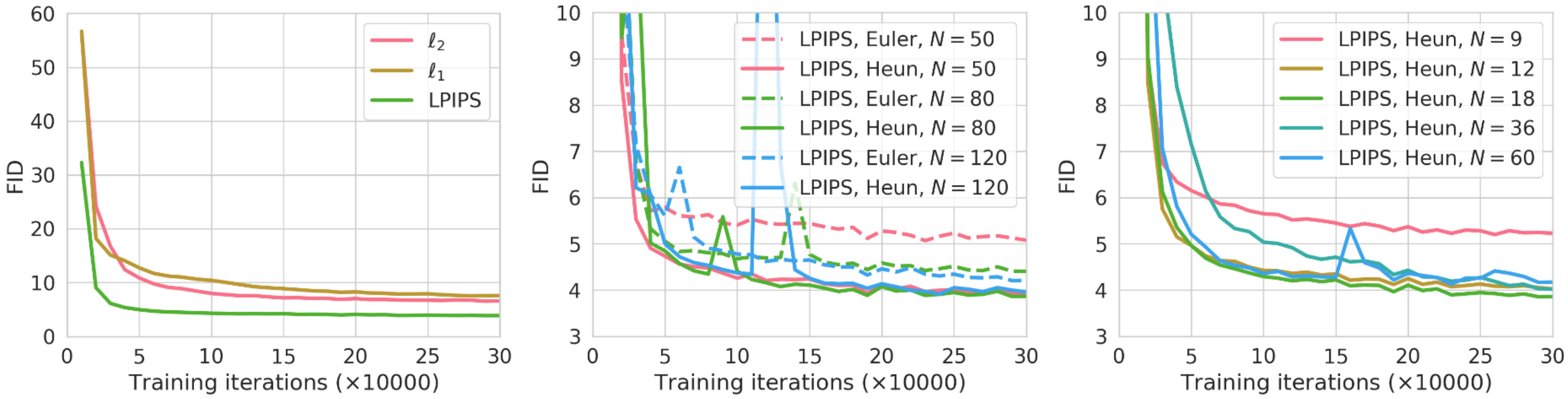
Consistency Training (L2 loss, stopgrad):

$$\mathcal{L}_{CT}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}) = 2\mathbb{E}\left[\lambda(t)\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)^{\mathsf{T}}\left(\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_t, t)}{\partial t} + \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \cdot \frac{\mathbf{x}_t - \mathbf{x}}{t}\right)\right].$$

Results – Consistency Distillation

LPIPS is best distance metric

Heun solver with N=18 performs best (for CIFAR10)

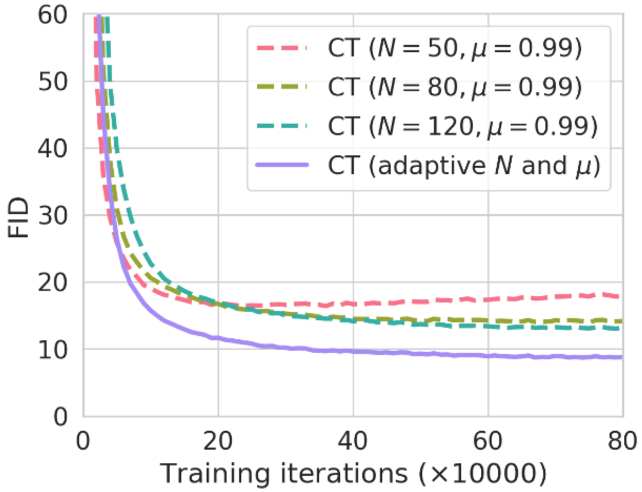


(a) Metric functions in CD. (b) Solvers and N in CD. (c) N with Heun solver in CD.

Results – Consistency Training

Based on CD results, LPIPS is used with Heun solver

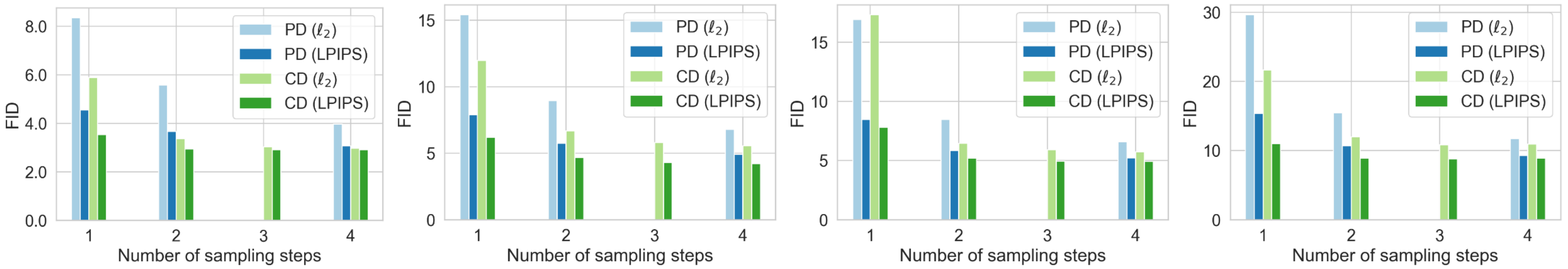
Low N has quick convergence but worse samples while high N has slow convergencebut better samples. Adaptive N schedule addresses this issue



(d) Adaptive N and μ in CT.

Results – Image Generation

CD beats Progressive Distillation (PD) at all timesteps for most datasets



(a) CIFAR-10 (b) ImageNet 64×64 (c) Bedroom 256×256 (d) Cat 256×256

Figure 4: Multistep image generation with consistency distillation (CD). CD outperforms progressive distillation (PD) across all datasets and sampling steps. The only exception is single-step generation on Bedroom 256×256.

Results – Image Generation

CD sets new SOTAs, while CT beats non-adversarial single-step methods (and is comparable to PD)

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)	12	5.28	
DPM-solver-3 (Lu et al., 2022)	12	6.03	
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
CR-GAN (Zhang et al., 2019)	1	14.6	8.40
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)	36	**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., 2019a)	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**

Results – Image Generation

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
SS-GAN (Chen et al., 2019b)	1	13.3		
PGGAN (Karras et al., 2018)	1	8.34		
PG-SWGAN (Wu et al., 2019)	1	8.0		
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

Results – Image Generation

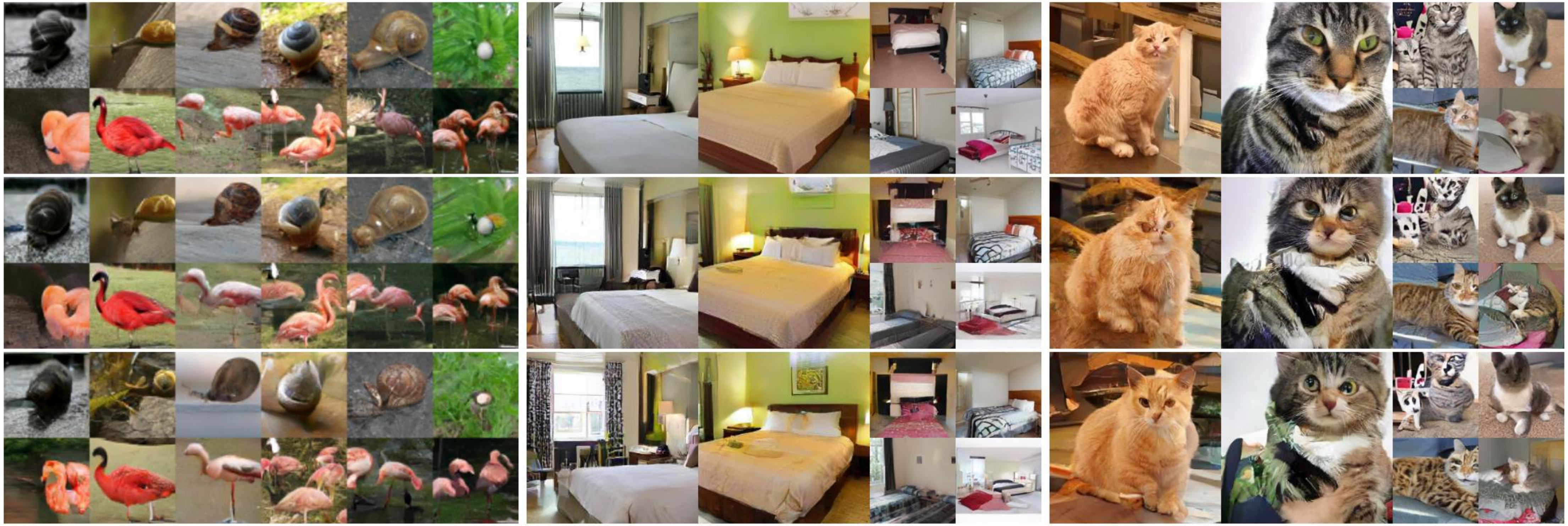


Figure 5: Samples generated by EDM (*top*), CT + single-step generation (*middle*), and CT + 2-step generation (*Bottom*). All corresponding images are generated from the same initial noise.

Results – Zero-shot Image Editing

(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.

Figure 6: Zero-shot image editing with a consistency model trained by consistency distillation on LSUN Bedroom 256×256.