Consistency Models

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever

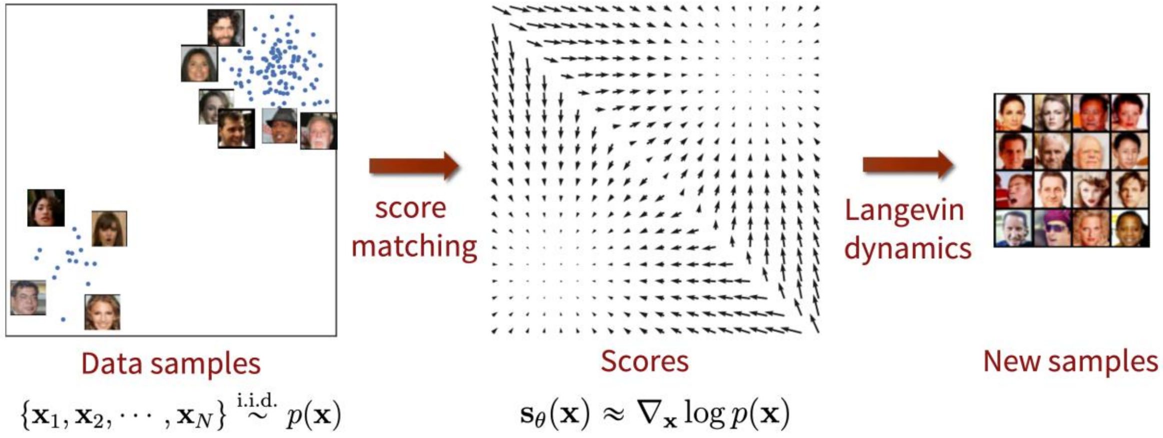
OpenAI

ICML 2023

Presented by Minho Park

Diffusion Models and Score-based Models

- Score function: the gradient of the logarithm of the probability density function, i.e., $\nabla_x \log p(x)$.
- Diffusion models: Latent variable models where latent variables are combined with a Markov chain.
  - The transitions of latent variables are defined as a conditional Gaussian distribution and starting at isotropic Gaussian distribution.



Score-based Models (NCSN)

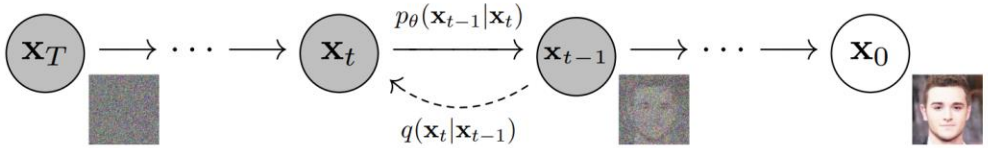


Figure 2: The directed graphical model considered in this work.

" Diffusion models [53] are latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (1)$$ "

Diffusion Models (DDPM)

Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." *NeurIPS.* 2019.
Sohl-Dickstein, Jascha, et al. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML.* 2015,

Reparameterize Diffusion Models

- Objective of diffusion models:

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

- We model $p_\theta(x_{t-1}|x_t)$ in $q(x_{t-1}|x_t,x_0)$ form (Gaussian distribution).
  - Predict x_0 with x_t. I.e., $\hat{x}_0(x_t,t;\theta)\Rightarrow q(x_{t-1}|x_t,\hat{x}_0)$.

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_{t-1};\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t\mathbf{I}),$$

$$\text{where}\quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0):=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0+\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t\quad\text{and}\quad\tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

- Reparameterize $\hat{x}_0$
  - $x_t=\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon$ where $\epsilon\sim\mathcal{N}(0,I)$.
  - $\hat{x}_0=\frac{x_t-\sqrt{1-\bar{\alpha}_t}\boxed{\epsilon_\theta(x_t,t)}}{\sqrt{\bar{\alpha}_t}}$ **Our new target!** **Is this similar to score function?**

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *NeurIPS.* 2020.

DDPM (SDE) vs. DDIM (ODE)

- Diffusion models also can be seem as stochastic differential equation.

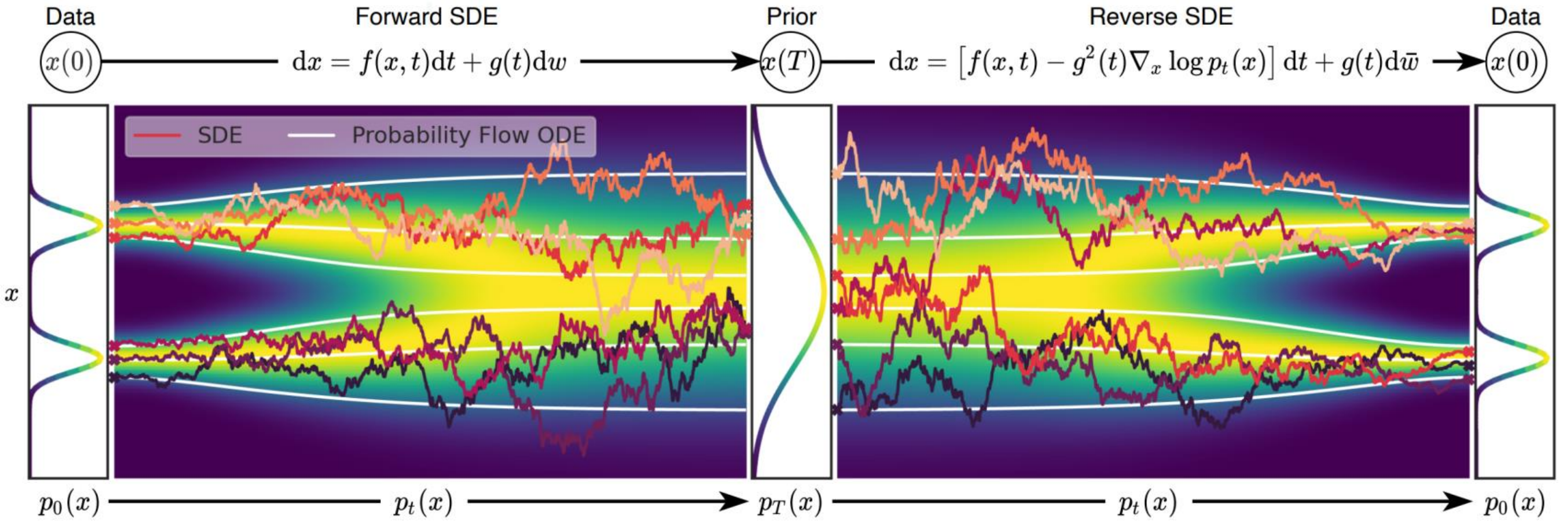


Figure 2: **Overview of score-based generative modeling through SDEs**. We can map data to a noise distribution (the prior) with an SDE (Section 3.1), and reverse this SDE for generative modeling (Section 3.2). We can also reverse the associated probability flow ODE (Section 4.3), which yields a deterministic process that samples from the same distribution as the SDE. Both the reverse-time SDE and probability flow ODE can be obtained by estimating the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Section 3.3).

Consistency Models

- Given a solution trajectory $\{x_t\}_{t\in[\epsilon,T]}$ of PF ODE (DDIM)
- Self-consistency: $f(x_t,t) = f(x_{t'},t')$ for all $t,t' \in [\epsilon,T]$.
  - Boundary condition: $f(x_\epsilon,\epsilon) = x_\epsilon$, i.e., $f(\cdot,\epsilon)$ is an identity function.

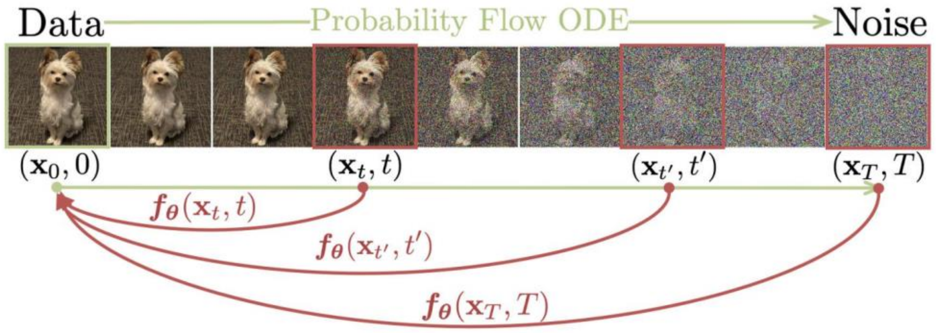


Figure 1: Given a Probability Flow (PF) ODE that smoothly converts data to noise, we learn to map any point (*e.g.*, $\mathbf{x}_t$, $\mathbf{x}_{t'}$, and $\mathbf{x}_T$) on the ODE trajectory to its origin (*e.g.*, $\mathbf{x}_0$) for generative modeling. Models of these mappings are called consistency models, as their outputs are trained to be consistent for points on the same trajectory.

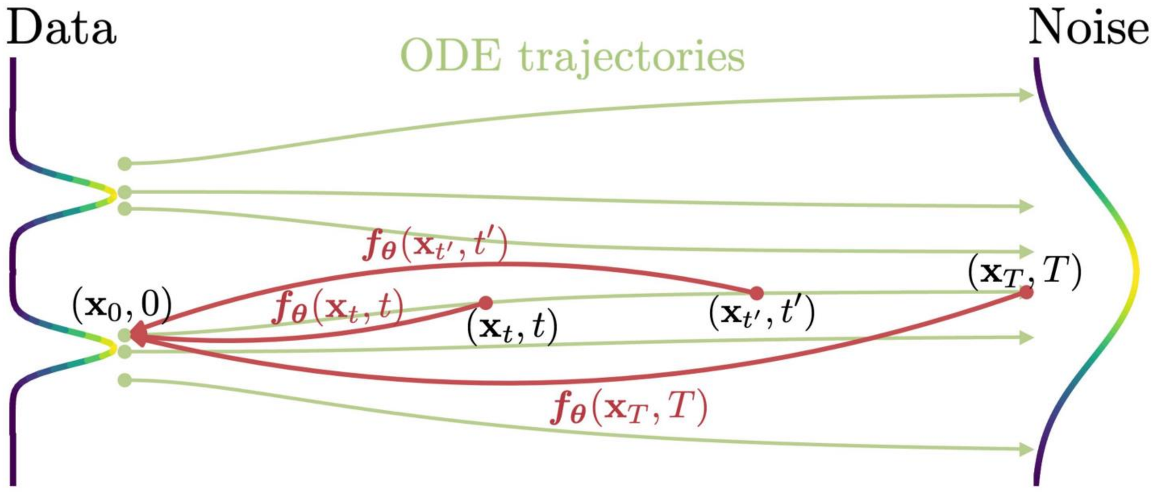


Figure 2: Consistency models are trained to map points on any trajectory of the PF ODE to the trajectory's origin.

Implement Boundary Condition Almost For Free

- Two simple parameterization tricks:

$F_\theta(x, t)$ is a free-form deep neural network.

1. Naïve parameterization

$$f_\theta(x,t) = \begin{cases} x & t = \epsilon \\ F_\theta(x,t) & t \in (\epsilon, T] \end{cases}$$

2. **Using skip connection**

$$f_\theta(x,t) = c_{skip}(t)x + c_{out}(t)F_\theta(x,t)$$

- $c_{skip}(t), c_{out}(t)$ are differentiable functions such that $c_{skip}(\epsilon) = 1, c_{out}(\epsilon) = 0$.

Inference Algorithm

- Single-step consistency sampling: $\hat{x}_\epsilon = f_\theta(x_T, T)$

- Multistep consistency sampling: Similar to DDIM sampling

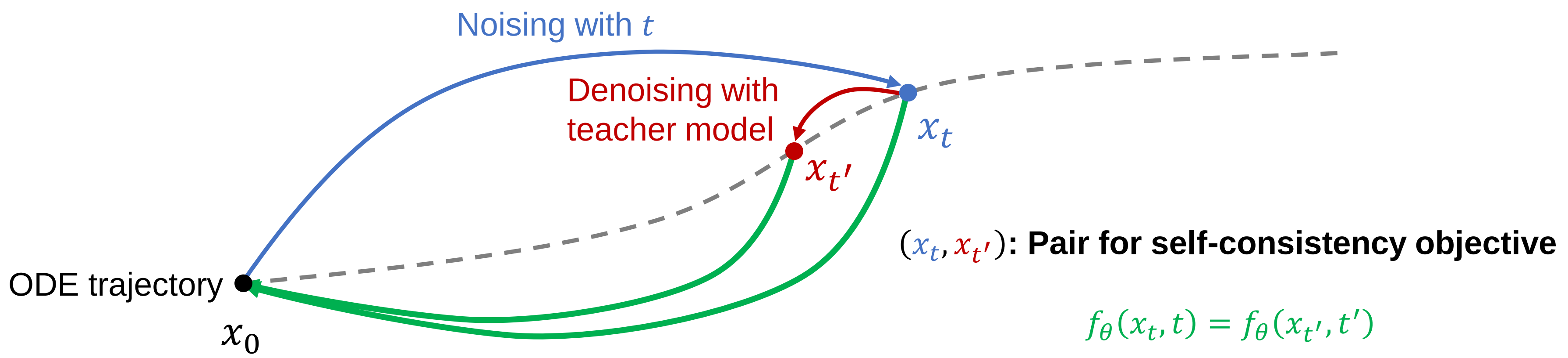
Algorithm 1 Multistep Consistency Sampling

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: $\mathbf{x}$

- In practice, they find time points with a **ternary search to optimize the FID of samples** obtained from Algorithm 1.
  - Ternary search needs assumption that the objective is a unimodal function.
  - FID is a unimodal function empirically in our experiments.

Training Consistency Models via Distillation

- Self-consistency objective: $f_\theta(x_t, t) = f_\theta(x_{t'}, t')$ for all $t, t' \in [\epsilon, T]$
- How can we obtain ODE trajectories from ODE solver?
- Sampling all trajectory $\{x_t\}_{t\in[\epsilon,T]}$ of PF ODE is inefficient.
  - To construct dataset, we sample x_T in pure Gaussian noise, and denoising it until $t = \epsilon$.
- We will use datasets for obtaining ODE trajectories.



Consistency Distillation

- Noising a sample x_0 with timestep t_{n+1}: $x_{t_{n+1}}$
- Find the sample on the same ODE trajectory utilizing pre-trained ODE solver ϕ.
- $\hat{x}_{t_n}^{\phi} \coloneqq x_{t_{n+1}} - (t_n - t_{n+1}) t_{n+1} s_{\phi}(x_{t_{n+1}}, t_{n+1})$ **Denoising with pre-trained diffusion models**
- Now, $x_{t_{n+1}}, \hat{x}_{t_n}^{\phi}$ are on the same ODE trajectory.

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

$$\mathcal{L}_{CD}^{N}(\theta, \theta^{-}; \phi) \coloneqq \mathbb{E}\left[\lambda(t_n) d\left(f_{\theta}(x_{t_{n+1}}, t_{n+1}), f_{\theta^{-}}\left(\hat{x}_{t_n}^{\phi}, t_n\right)\right)\right]$$

- θ^{-}: running average of the past values of θ, i.e., EMA of θ
- d: metric function such as L1, L2, and LPIPS

Consistency Distillation

- We find that compared to simply setting $\theta^{-} = \theta$, the EMA update and "stopgrad" operator can **greatly stabilize** the training process and improve the final performance of the consistency model.
- In alignment with the convention in deep reinforcement learning and momentum based contrastive learning we refer to f_{θ^-} as the "target network", and f_θ as the "online network".

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 $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 \boldsymbol{I})$
  $\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}} \leftarrow \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}; \boldsymbol{\phi})$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \boldsymbol{\phi}) \leftarrow$
    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \boldsymbol{\phi})$
  $\boldsymbol{\theta}^{-} \leftarrow \text{stopgrad}(\mu \boldsymbol{\theta}^{-} + (1-\mu)\boldsymbol{\theta})$
until convergence

What is the difference between

$$\mathcal{L}_{CD}^{N}(\theta, \theta^{-}; \phi) \coloneqq \mathbb{E}\left[\lambda(t_n) d\left(f_\theta(x_{t_{n+1}}, t_{n+1}), f_{\theta^-}\left(\hat{x}_{t_n}^{\phi}, t_n\right)\right)\right]$$

$$\mathcal{L}_{CD}^{N}(\theta, \theta^{-}; \phi) \coloneqq \mathbb{E}\left[\lambda(t_n) d\left(f_{\theta^-}(x_{t_{n+1}}, t_{n+1}), f_\theta\left(\hat{x}_{t_n}^{\phi}, t_n\right)\right)\right]$$

Theorem 1.

- If we reduce consistency distillation loss successfully, the difference between the original diffusion models and the consistency models will be reduced.

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*

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

Proof. The proof is based on induction and parallels the classic proof of global error bounds for numerical ODE solvers (Süli & Mayers, 2003). We provide the full proof in Appendix A.2. □

Remark:

- Since θ^- is a running average of the history of θ, we have $\theta^- = \theta$ when the optimization of Algorithm 2 converges.
- Importantly, our boundary condition $f_\theta(x, \epsilon) = x$ precludes the trivial solution $f_\theta(x, \epsilon) = 0$ from arising in consistency model training.

Training Consistency Models in Isolation

- Distilling from unbiased PF ODE estimator:

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

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

Definition 2. The consistency training loss is defined as

$$\mathcal{L}_{CD}^N(\theta, \theta^-; \phi) \coloneqq \mathbb{E}\big[\lambda(t_n) d\big(f_\theta(x + t_{n+1}z, t_{n+1}), f_{\theta^-}(x + t_n z, t_n)\big)\big]$$

Theorem 2.

- The consistency training loss is closely related to consistency distillation loss with the ground truth score model ϕ.

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* $\inf_N \mathcal{L}_{CD}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) > 0$.

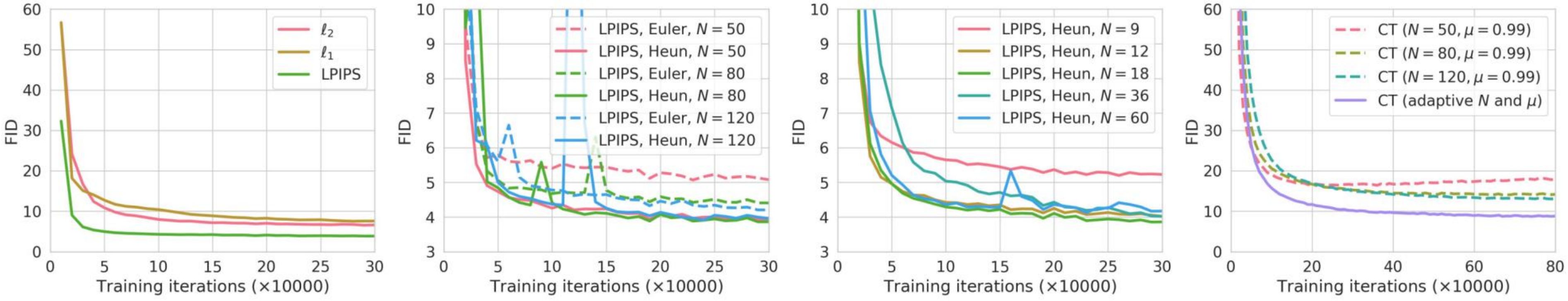
Proof. The proof is based on Taylor series expansion and properties of score functions (Lemma 1). A complete proof is provided in Appendix A.3. □

Remark:

- $\mathcal{L}_{CT}^{N}(\theta, \theta^-)$ only depends on the online network f_θ, and the target network f_{θ^-}, while being completely agnostic to diffusion model parameters ϕ.
- The loss function $\mathcal{L}_{CT}^{N}(\theta, \theta^-) \geq O(\Delta t)$ decreases at a slower rate than the remainder $o(\Delta t)$ and thus will dominate the loss in Eq. (9) as $N \to \infty$ and $\Delta t \to 0$.

Hyperparameter Search

- Metric functions, ODE solvers, and number of timesteps.
- Furthermore, they proposed "progressively increasing N".
  - When N is small (i.e., Δt is small), the consistency training loss has less "variance" but more "bias" underlying consistency distillation loss.
  - On the contrary, it has more "variance" but less "bias" when N is large.

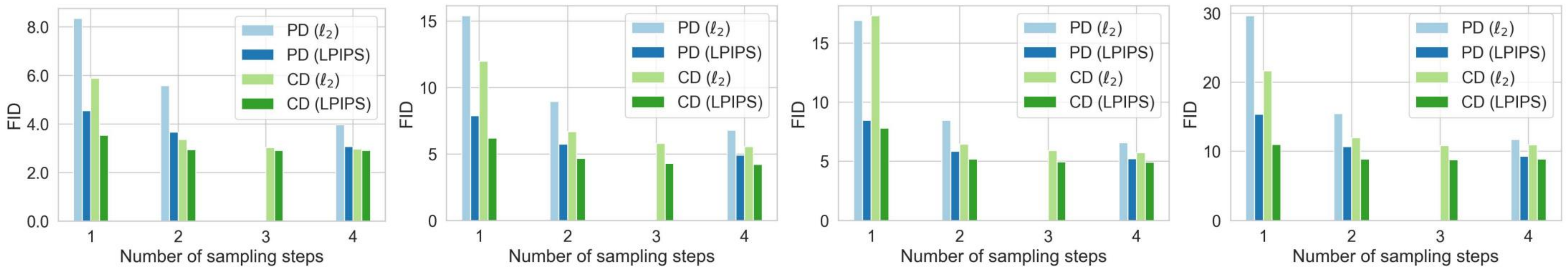


(a) Metric functions in CD. (b) Solvers and N in CD. (c) N with Heun solver in CD. (d) Adaptive N and μ in CT.

Figure 3: Various factors that affect consistency distillation (CD) and consistency training (CT) on CIFAR-10. The best configuration for CD is LPIPS, Heun ODE solver, and $N = 18$. Our adaptive schedule functions for N and μ make CT converge significantly faster than fixing them to be constants during the course of optimization.

Comparison With Progressive Distillation

- CD and PD are thus far the only distillation approaches that do not construct synthetic data before distillation.
- Using the LPIPS metric uniformly improves CD and PD compared to the squared L2 distance.
- CD uniformly outperforms PD across all datasets, sampling steps, and metric functions considered.



(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.

Comparison With State-of-the-art Generation Methods

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

- NFE: Neural Function Evaluations
- PD and CD distill the same EDM models.

Remark

- CT outperforms existing single-step, non-adversarial generative models
- CT achieves comparable quality to one-step samples from PD without relying on distillation.
- CD is better than CT though CT is trained with “unbiased” ground truth score model.

Karras, Tero, et al. "Elucidating the design space of diffusion-based generative models." *NeurIPS.* 2022.

Qualitative Results

- Comparison between EDM, CT (1 step), and CT (2 step)
- All samples obtained from the same initial noise vector share significant structural similarity, even though CT and EDM models are trained independently from one another.
- This indicates that CT is less likely to suffer from mode collapse, as EDMs do not.

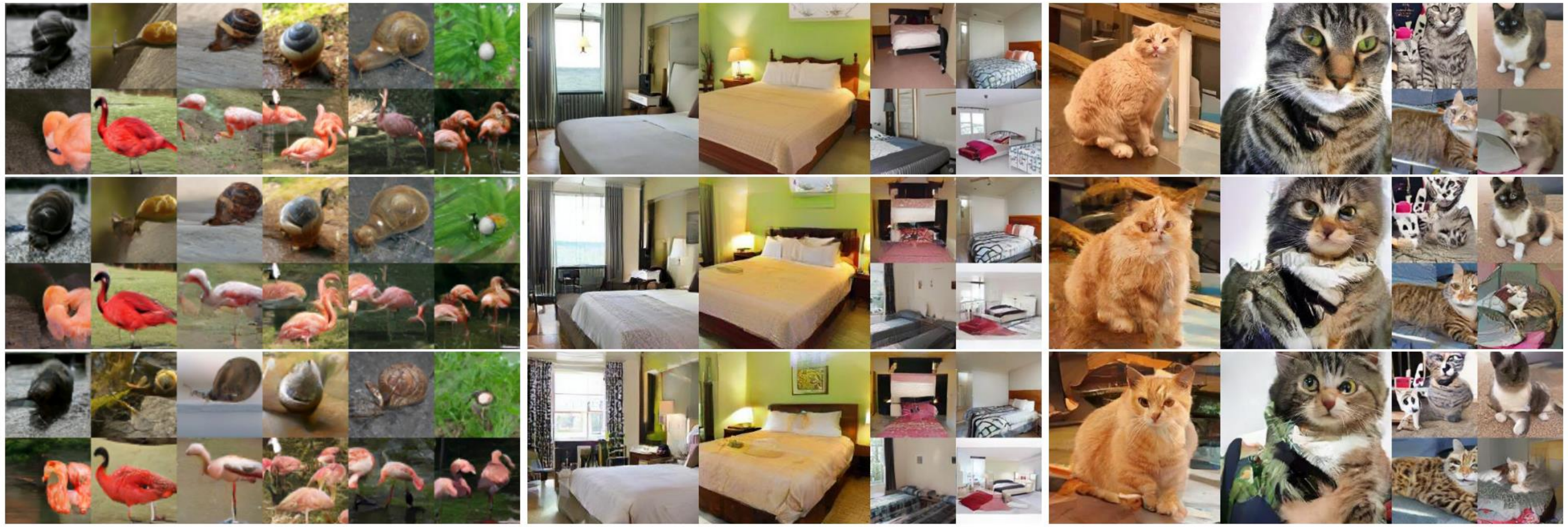


Figure 5: Samples generated by EDM (*top*), CT + single-step generation (*middle*), and CT + 2-step generation (*Bottom*). All corresponding images are generated from the same initial noise.

Zero-shot Image Editing

- Consistency models with multi-step sampling enable zero-shot image editing as diffusion models.

(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.

Subsequent Study: BOOT

- BOOT: Data-free Distillation of Denoising Diffusion Models with Bootstrapping
- BOOT can distill large-scale diffusion models DeepFloyd IF similar to Stable Diffusion
  - Project page, hugging face studio

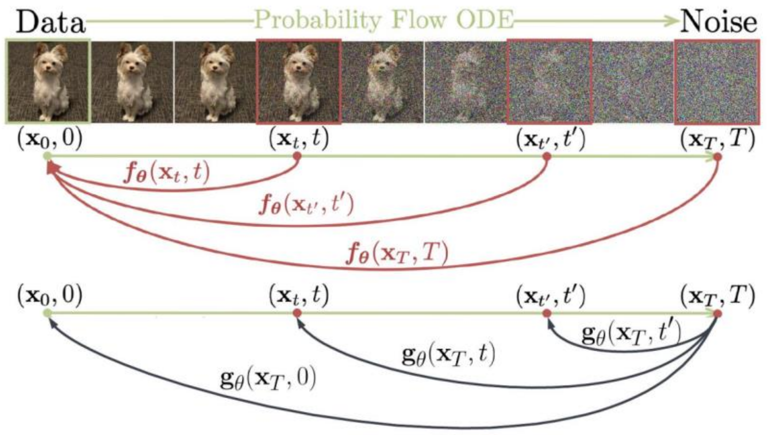


Figure 2: Comparison of Consistency Model (Song et al., 2023) (red ↑) and BOOT (black ↓) highlighting the opposing prediction pathways.

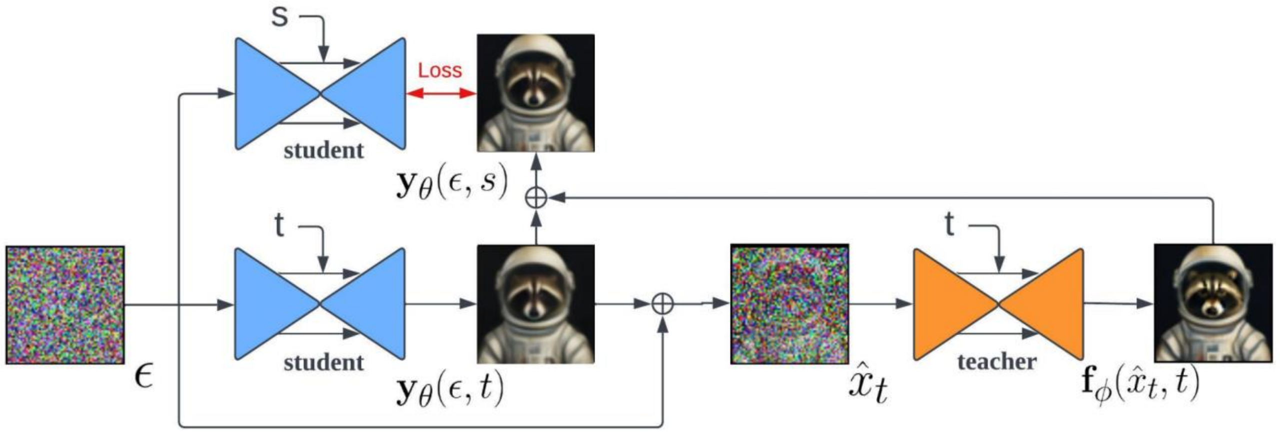


Figure 3: Training pipeline of BOOT. s and t are two consecutive timesteps where $s < t$. From a noise map $\boldsymbol{\epsilon}$, the objective of BOOT minimizes the difference between the output of a student model at timestep s, and the output of stacking the same student model and a teacher model at an earlier time t. **The whole process is data-free.**

Gu, Jiatao, et al. "BOOT: Data-free Distillation of Denoising Diffusion Models with Bootstrapping." *arXiv.* 2023.

BOOT: Data-free Distillation of Denoising Diffusion Models with Bootstrapping

Jiatao Gu et al.

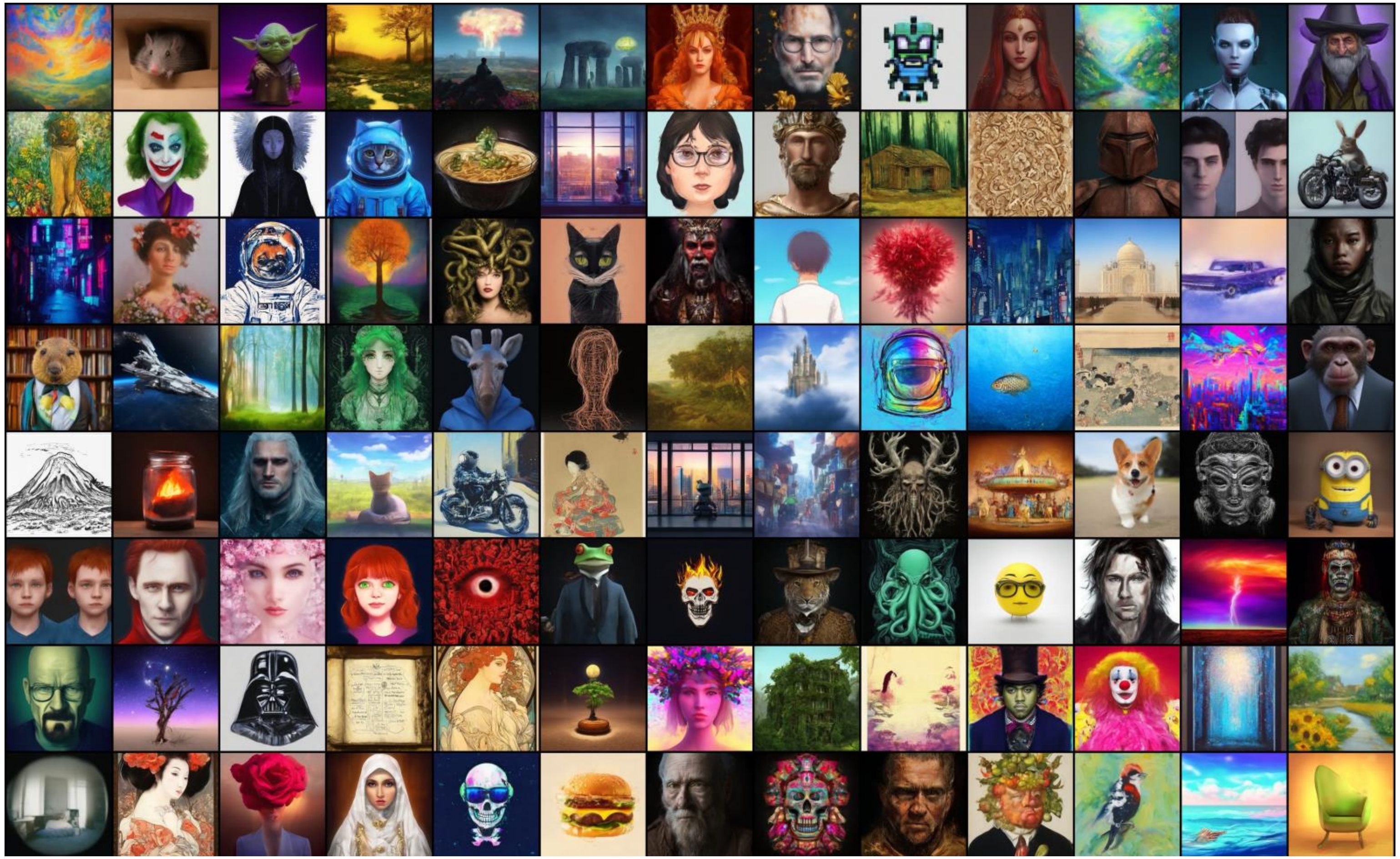
Apple, University of Pennsylvania

arXiv

Presented by Minho Park

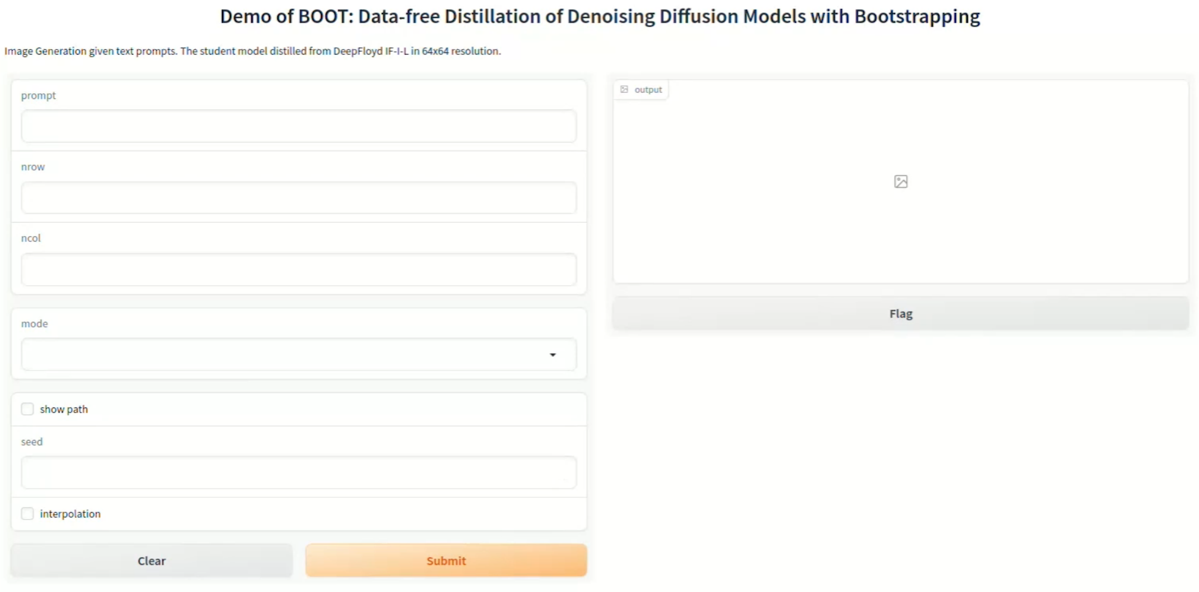
Motivation

- Random samples in 256x256 from our **single-step** student models distilled from DeepFloyd IF with prompts from diffusiondb.



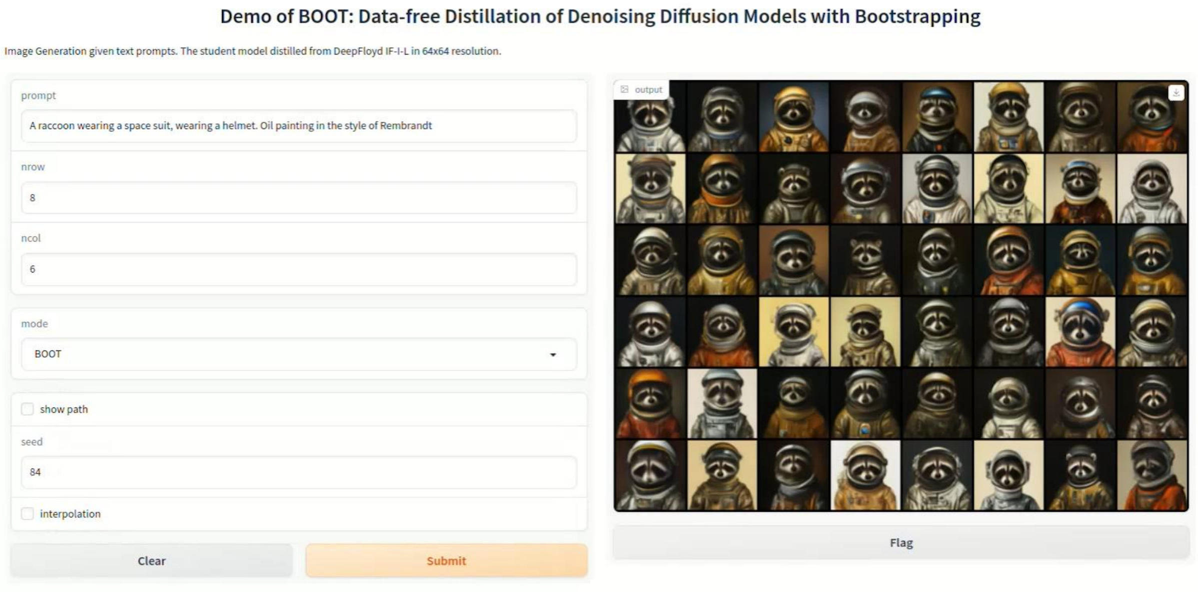
Motivation

First, prepare to run BOOT



prompt	nrow	ncol	mode	show path	seed
impasto, avatar, illustration, girl with red hair, slightly curly hair, European and American, freckles, jane's style, trends on artstation, crazy colors, light and shadow contrast, high detail.	8	6	BOOT	false	84
Papillon dog puppy in the style of pencil drawing, fantasy art, enigmatic, mysterious.	8	6	BOOT	false	84
A raccoon wearing a space suit, wearing a helmet. Oil painting in the style of Rembrandt	8	6	BOOT	false	84
A portrait of Einstein, style art, award winning quality, high detail	8	6	BOOT	false	84
An intricate forest painting, full of exotic plants and flowers, Arianna Caroli.	8	6	BOOT	false	84

Now, we show more results of BOOT



prompt	nrow	ncol	mode	show path	seed
impasto, avatar, illustration, girl with red hair, slightly curly hair, European and American, freckles, jane's style, trends on artstation, crazy colors, light and shadow contrast, high detail.	8	6	BOOT	false	84
Papillon dog puppy in the style of pencil drawing, fantasy art, enigmatic, mysterious.	8	6	BOOT	false	84
A raccoon wearing a space suit, wearing a helmet. Oil painting in the style of Rembrandt	8	6	BOOT	false	84
A portrait of Einstein, style art, award winning quality, high detail	8	6	BOOT	false	84
An intricate forest painting, full of exotic plants and flowers, Arianna Caroli.	8	6	BOOT	false	84

Knowledge Distillation of DDPM

- DDPM을 빠르게 만드는 두 가지 방법: 1) ODE-solver 연구, 2) distillation-based methods
  - ODE-solver: DDIM, PLMS, etc.
  - Distillation-based: Progressive distillation, **Consistency models**, and **BOOT**
- Comparison with Consistency Models

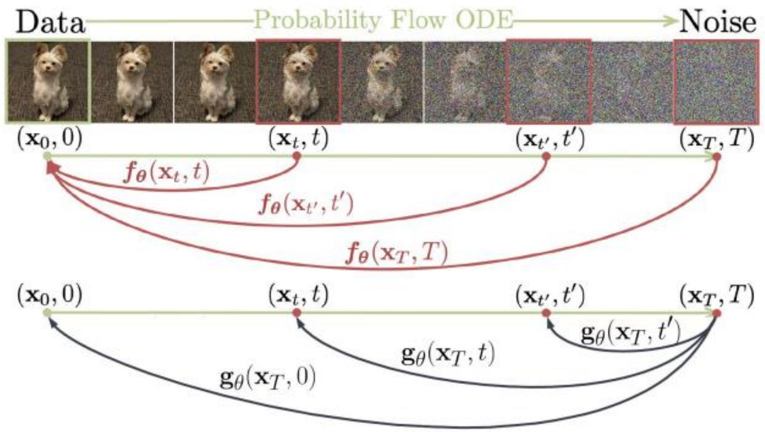


Figure 2: Comparison of Consistency Model (Song et al., 2023) (red ↑) and BOOT (black ↓) highlighting the opposing prediction pathways.

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

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

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

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

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

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

until convergence

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

Song, Yang, et al. "Consistency models." *ICML.* 2023.

Direct Distillation vs. Consistency Models

- **Direct distillation:** (Noise, Generated image pair)를 이용하여 student model을 학습하는 방법. ODE-solver를 끝까지 통과시켜 pair를 얻어야 하므로 한 step 학습하는데 50 steps을 통과시켜야 함.

$$\mathcal{L}_\theta^{\text{Direct}} = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)} \|\boldsymbol{g}_\theta(\boldsymbol{\epsilon}) - \texttt{ODE-Solver}(\boldsymbol{f}_\phi, \boldsymbol{\epsilon}, T \to 0)\|_2^2$$

- **Consistency models:** 50 steps의 ODE-solver를 지나기 힘들기 때문에, 한 스텝의 ODE-solver만 거치고 bootstrap fashion으로 self-consistent objective를 걸어주어서 student model을 학습시킴.

$$\mathcal{L}_\theta^{\text{CM}} = \mathbb{E}_{\boldsymbol{x}_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}),s,t\sim[0,T],s<t} \|\boldsymbol{g}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{g}_{\theta^-}(\boldsymbol{x}_s, s)\|_2^2$$

  - θ^-: self-consistency objective 에서 도움이 되었던 EMA teacher.

- **하지만, consistency models은 학습 때 사용했던 데이터를 필요로 하는데, text-to-image의 경우 billions의 (private 일 수도 있는) 데이터가 필요하므로 distillation 학습이 힘들다.**
  - 학습 시 쓰지 않았던 작은 데이터로 distillation을 진행하면 suboptimal distillation performance를 낸다고 함.

Luhman, Eric, and Troy Luhman. "Knowledge distillation in iterative generative models for improved sampling speed." *arXiv.* 2021.

Method: Single-ODE

- Direct distillation의 data-free, Consistency models의 짧은 training time의 장점을 결합한 것이 BOOT.
- 대신 BOOT는 consistency models와 다르게 항상 noise ϵ을 입력으로 받는 모델이다.
  - I.e., $g_\theta(\epsilon, t) \approx x_t = \mathrm{ODE} - \mathrm{Solver}\left(f_\phi, \epsilon, T \to t\right)$. The final sample can be obtained as $g_\theta(\epsilon, 0) \approx x_0$.
- 이 때, $g_\theta(\epsilon, t) \approx x_t$로 θ를 학습하는 것은 x_t는 noisy 한 image이므로 학습 수렴 어렵다.
  - 대신 $p(x_t|x)$의 평균인 $\mu_t = y_t = (x_t - \sigma_t \epsilon)/\alpha_t$를 예측하도록 하자.
  - Boundary condition: $y_0 = x_0$, $y_T =?$ ⇒ 이후 T 쪽 boundary condition loss를 추가해줌.
  - 저자들은 low frequency “signal” component of x_t라고 부른다.

Method: Single-ODE

- DDIM sampling ($s < t$)

$$\boldsymbol{x}_s = (\sigma_s/\sigma_t)\,\boldsymbol{x}_t + (\alpha_s - \alpha_t\sigma_s/\sigma_t)\,\overset{\hat{x}_0}{\underline{\boldsymbol{f}_\phi(\boldsymbol{x}_t, t)}}$$

- Reparameterization with y and $\lambda_t = -\log(\alpha_t/\sigma_t)$. 이 때 λ_t는 “negative half log-SNR”라고도 불림.

$$\boldsymbol{y}_s = \left(1 - e^{\lambda_s - \lambda_t}\right) \boldsymbol{f}_\phi(\boldsymbol{x}_t, t) + e^{\lambda_s - \lambda_t}\boldsymbol{y}_t$$

- Continuous version for objective function: ($s - t$의 꼴로 묶어서 $\lim\limits_{s \to t}$ ... 하면 됨)

$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = -\lambda_t' \cdot (\boldsymbol{f}_\phi(\boldsymbol{x}_t, t) - \boldsymbol{y}_t)$$

Method: Learning with Bootstrapping

- Objective function:

Collapsing을 막기 위한 EMA가 필요 없었음.
Teacher ϕ가 어차피 non-zero output을 뱉기 때문.

y_T 쪽에 대한 boundary condition loss

$$\mathcal{L}_\theta^{\mathrm{BS}} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I),t\sim[\delta,T]}\left[\frac{\tilde{w}_t}{\delta^2}\left\|\boldsymbol{y}_\theta(\boldsymbol{\epsilon},s) - \mathrm{SG}\left[\boldsymbol{y}_\theta(\boldsymbol{\epsilon},t) + \underbrace{\delta\lambda_t'\cdot\left(\left(\boldsymbol{f}_\phi(\hat{\boldsymbol{x}}_t,t)\right) - \boxed{\boldsymbol{y}_\theta(\boldsymbol{\epsilon},t)}\right)}_{\text{incremental improvement}}\right]\right\|_2^2\right]$$

$$\mathcal{L}_\theta^{\mathrm{BC}} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\|\boldsymbol{f}_\phi(\boldsymbol{\epsilon},t_{\max}) - \boldsymbol{y}_\theta(\boldsymbol{\epsilon},t_{\max})\|_2^2\right]$$

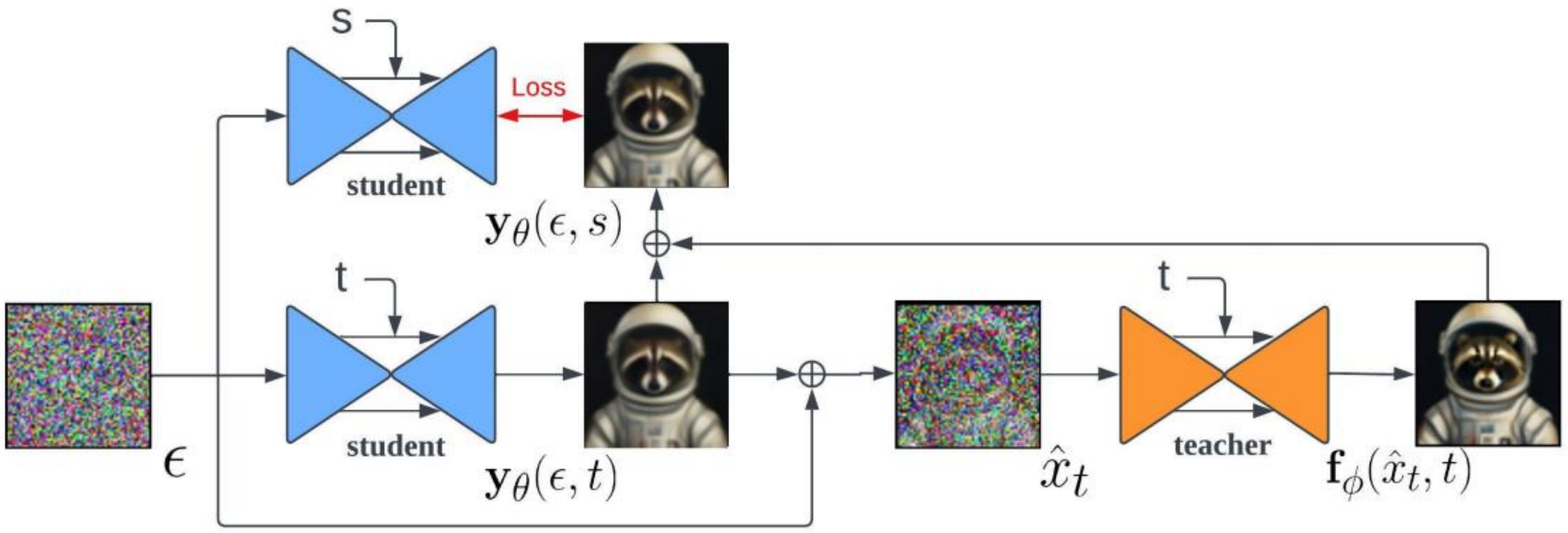


Figure 3: Training pipeline of BOOT. s and t are two consecutive timesteps where $s < t$. From a noise map ϵ, the objective of BOOT minimizes the difference between the output of a student model at timestep s, and the output of stacking the same student model and a teacher model at an earlier time t. **The whole process is data-free.**

Method: Error Accumulation

- BOOT에서는 y_θ가 large t에 대해서 부정확할 때, 이후 과정들도 다 같이 부정확해지는 문제가 학습에서 발생할 수 있다.
  - y_θ가 large t에 대해서 부정확하면, teacher model이 out-of-distribution input을 받게 되고 이후 학습이 불안정해지는 문제가 발생함.

- 이를 막기 위해서 두 가지 방법을 사용하였음.

1. 우리는 최종적으로 $t = 0$의 입력만 중요함에도 불구하고, DDPM의 학습 때 처럼 $t = 0, \dots, T$를 uniform 하게 sampling하여 학습을 진행하였음.
2. Bootstrapping하는 target의 결과로써 앞의 수식은 1st order method이지만, 실제로는 higher-order solver인 Heun's method 등을 사용하였음. (Heun's method: 2nd order method)

Method: Learning with Bootstrapping

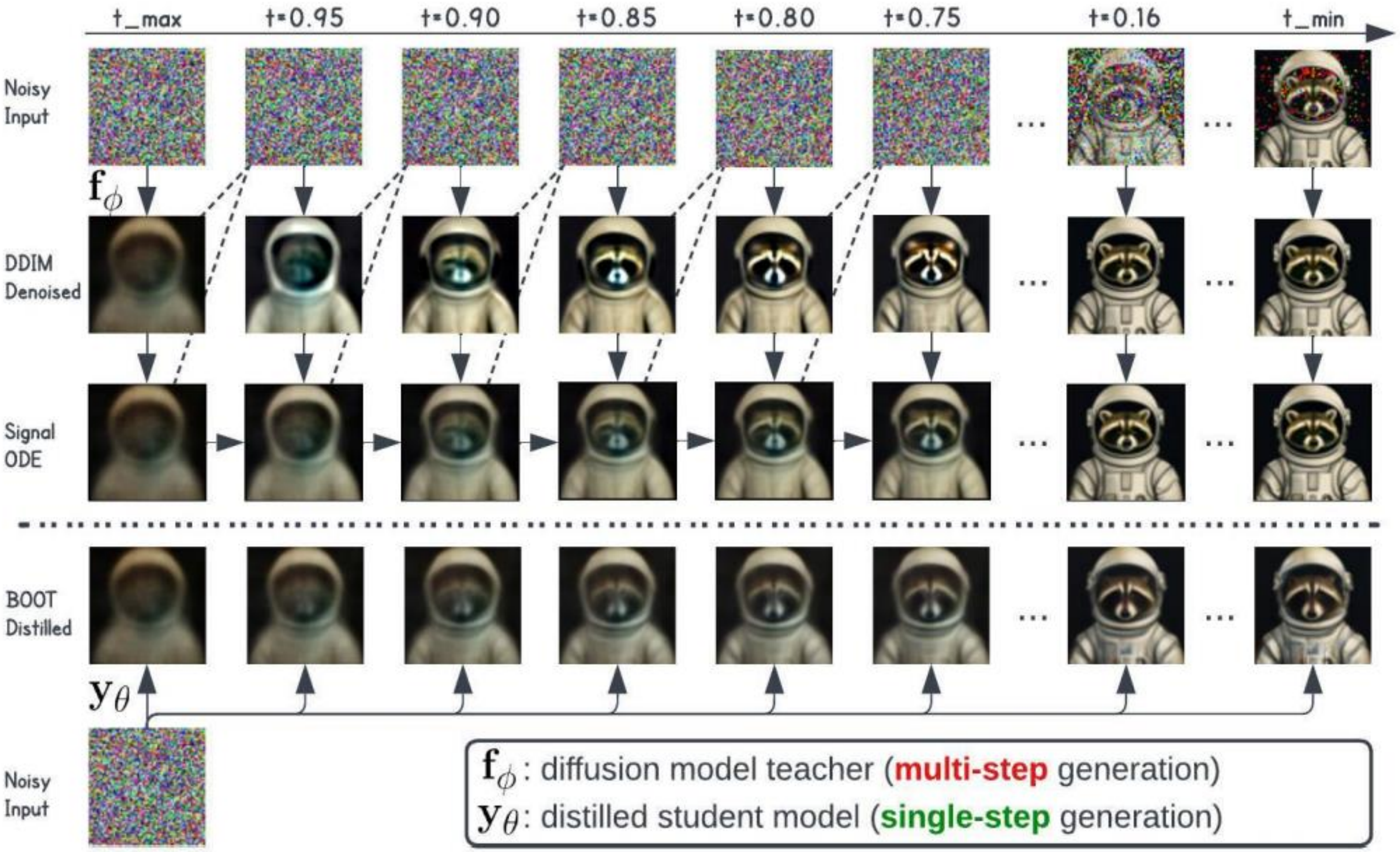


Figure 4: Comparison between the generated outputs of DDIM/Signal-ODE and our distilled model given the same prompt *A raccoon wearing a space suit, wearing a helmet. Oil painting in the style of Rembrandt* and initial noise input. By definition, signal-ODE converges to the same final sample as the original DDIM, while the distilled single-step model does not necessarily follow.

Method: Limitation

- Consistency models은 여전이 multi-step inference를 지원하는 반면에 BOOT는 더 이상 multi-step inference를 지원하지 않는다. 즉, multi-step inference 덕에 GAN에서 할 수 없고 Diffusion에서는 할 수 있던 방법론들이 불가능해졌다.
  - Zero-shot inpainting 등이 불가능함.
  - Classifier-free guidance도 불가능함.
- Distillation with guidance: Test-time에 guidance를 주지 못하기 때문에 guidance를 미리 고정해두고 distillation을 진행해야 한다.
  - Negative prompt도 고정해둬야 함.

$$\tilde{\boldsymbol{f}}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c}) = \boldsymbol{f}_\phi(\boldsymbol{x}_t, t, \boldsymbol{n}) + w \cdot (\boldsymbol{f}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c}) - \boldsymbol{f}_\phi(\boldsymbol{x}_t, t, \boldsymbol{n}))$$

Quantitative Results

	Steps	FFHQ 64 × 64		LSUN 256 × 256		ImageNet 64 × 64	
		FID / Prec. / Rec.	**fps**	**FID / Prec. / Rec.**	**fps**	**FID / Prec. / Rec.**	**fps**
DDPM	250	5.4 / 0.80 / 0.54	0.2	8.2 / 0.55 / 0.43	0.1	11.0 / 0.67 / 0.58	0.1
DDIM	50	7.6 / 0.79 / 0.48	1.2	13.5 / 0.47 / 0.40	0.6	13.7 / 0.65 / 0.56	0.6
	10	18.3 / 0.78 / 0.27	5.3	31.0 / 0.27 / 0.32	3.1	18.3 / 0.60 / 0.49	3.3
	1	225 / 0.10 / 0.00	54	308 / 0.00 / 0.00	31	237 / 0.05 / 0.00	34
Ours	1	9.0 / 0.79 / 0.38	54	23.4 / 0.38 / 0.29	32	16.3 / 0.68 / 0.36	34

Table 1: Comparison for image generation benchmarks on FFHQ, LSUN and class-conditioned ImageNet. For ImageNet, numbers are reported without using CFG ($w = 1$).

Qualitative Results

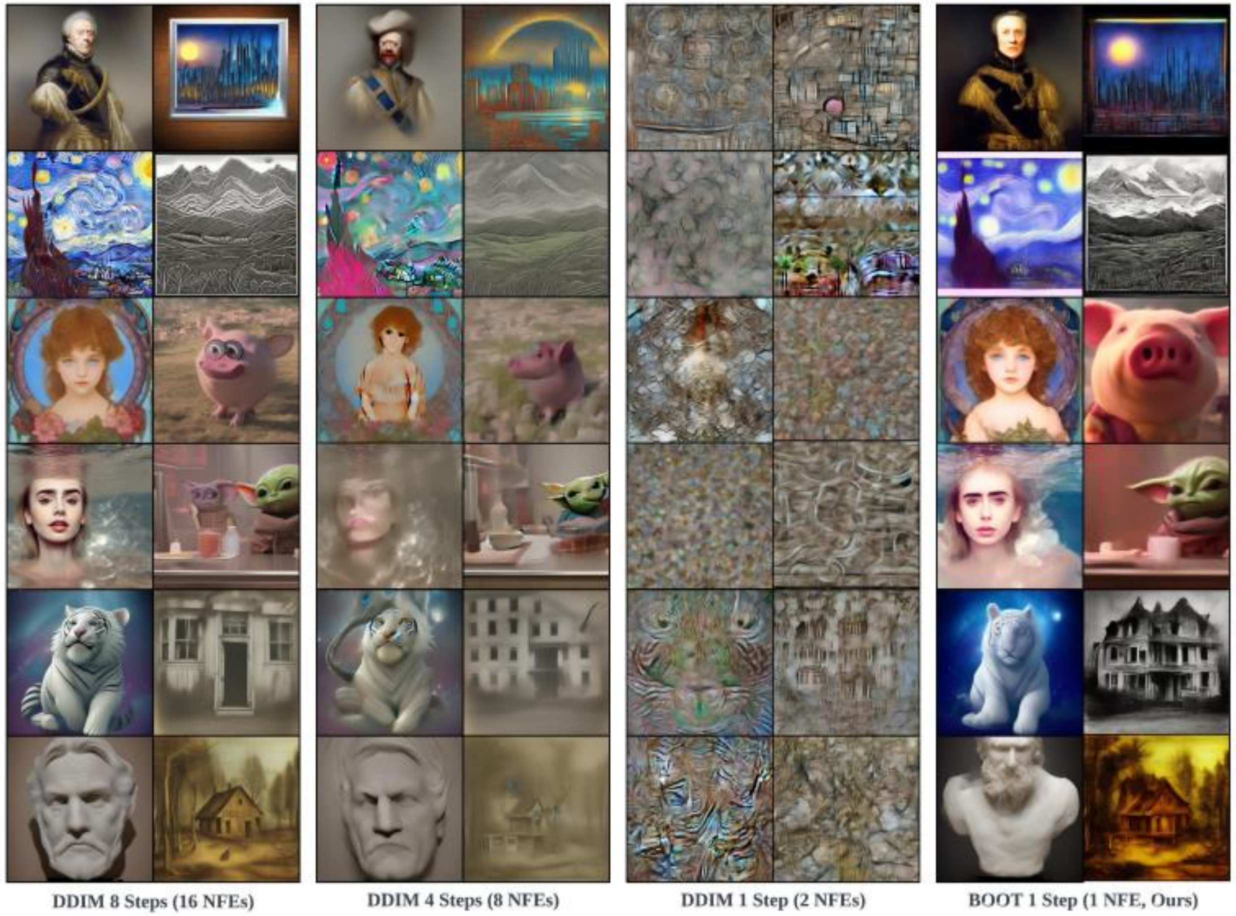


Figure 6: Uncurated samples of {50, 10, 1} DDIM sampling steps and the proposed BOOT from SD2.1-base, given the same set of initial noise input and prompts sampled from *diffusiondb*.

Qualitative Results

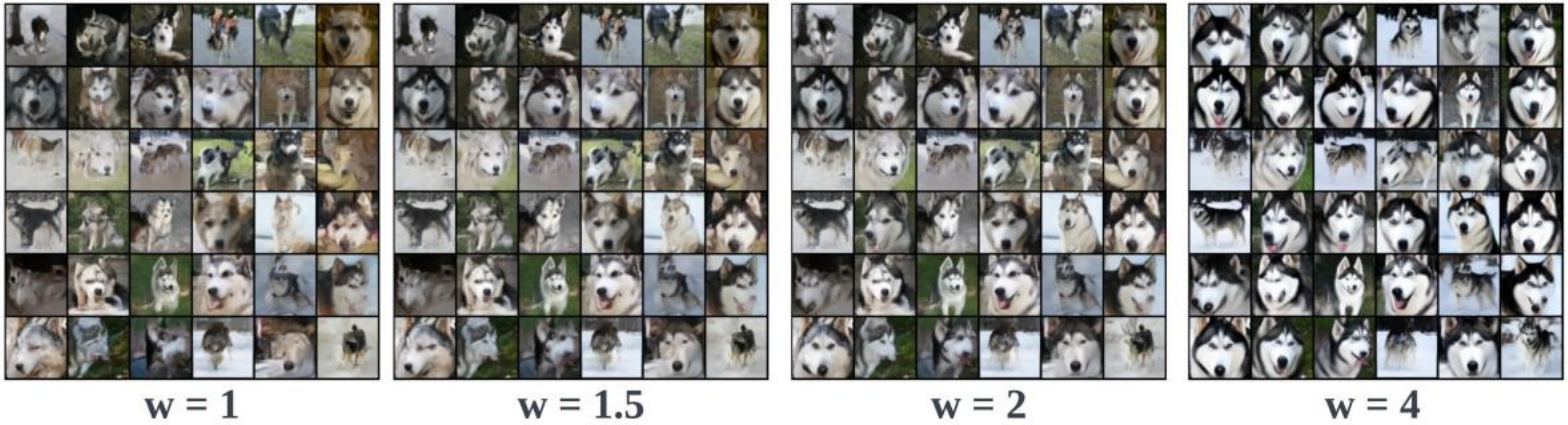


Figure 7: The distilled student is able to trade generation quality with diversity based on CFG weights.

Ablation Studies

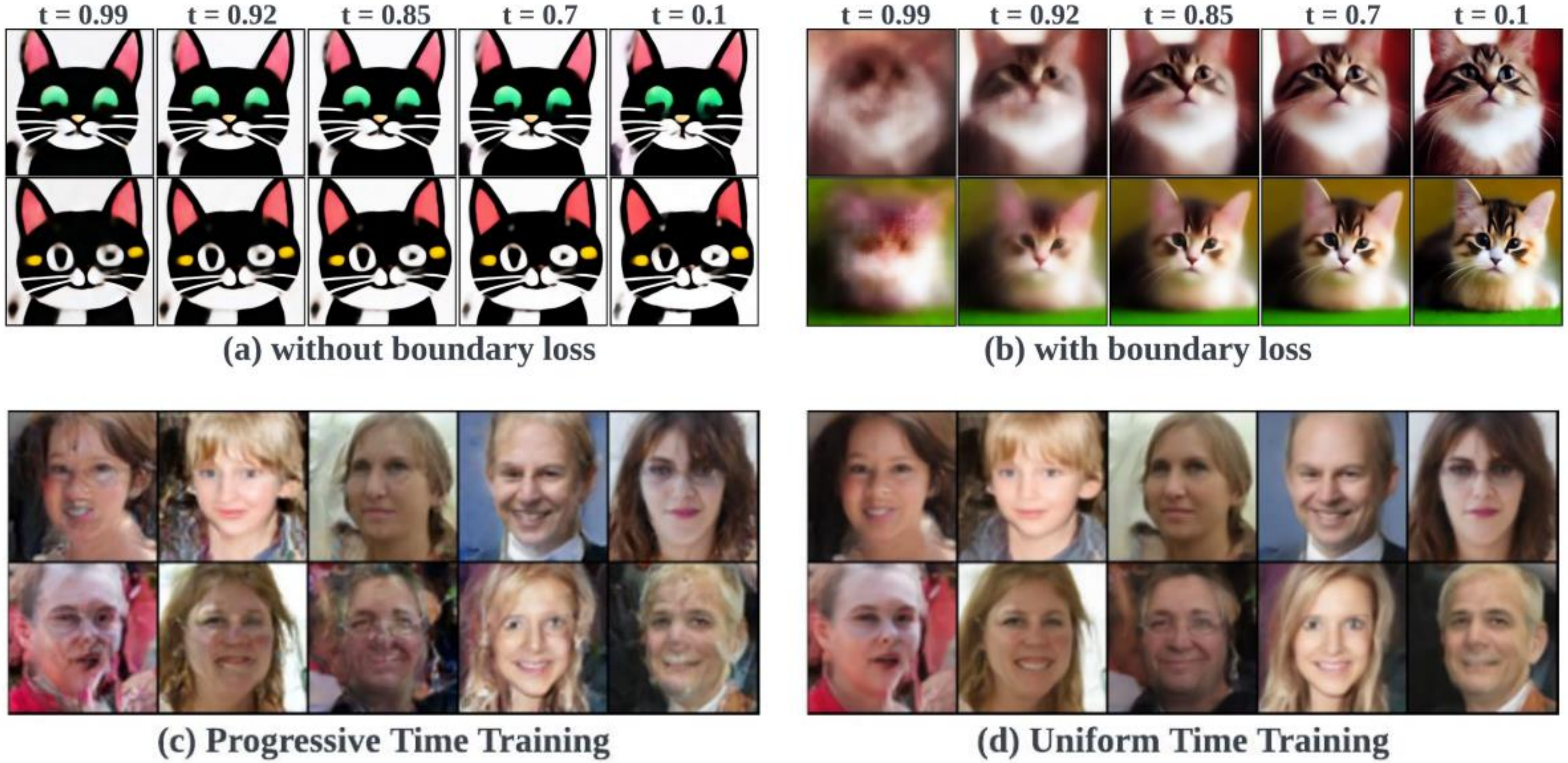
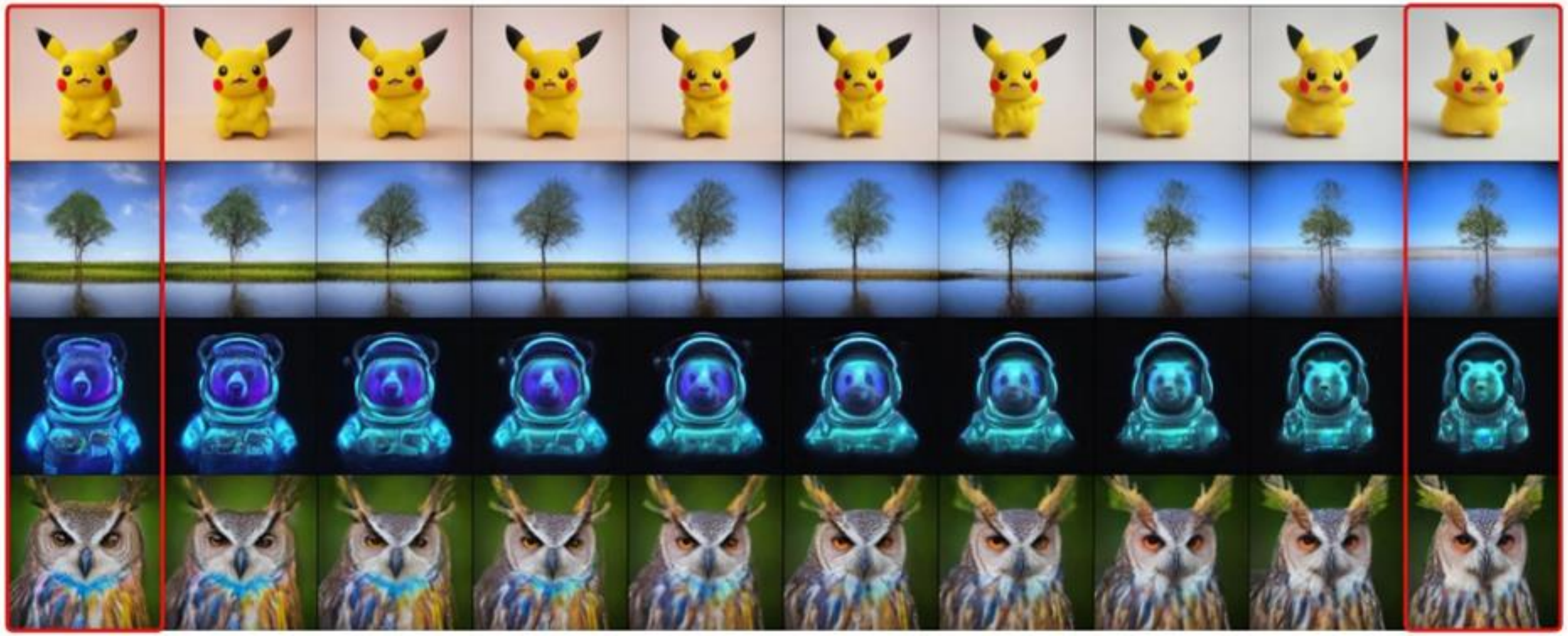


Figure 8: Ablation Study. (a) vs. (b): The additional boundary loss in § 3.2 alleviates the mode collapsing issue and prompts diversity in generation. (c) vs. (d): Uniform time training yields better generation compared with progressive time training.

Latent Space Interpolation

- GAN과 더 비슷해졌기 때문에 GAN에서 했던 분석들을 할 수 있을 것 같음.



GAN처럼 더 극단적인 latent space interpolation 을 보고 싶기는 함.

Figure 9: Latent space interpolation of the student model distilled from the IF teacher. We randomly sample two noises to generate images (shown in red boxes) given the same text prompts, and then linearly interpolate the noises to synthesize images shown in the middle.

Fixed Noise Input

- GAN과 더 비슷해졌기 때문에 GAN에서 했던 분석들을 할 수 있을 것 같음.

Figure 10: With fixed noise, we can perform controllable generation by swapping the keywords from the prompts. The prompts are chosen from the combination of *portrait of a {owl, raccoon, tiger, fox, llama, gorilla, panda} wearing { a t-shirt, a jacket, glasses, a crown} { drinking a latte, eating a pizza, reading a book, holding a cake} cinematic, hdr.* All images are generated from the student distilled from IF teacher given the same noise input.