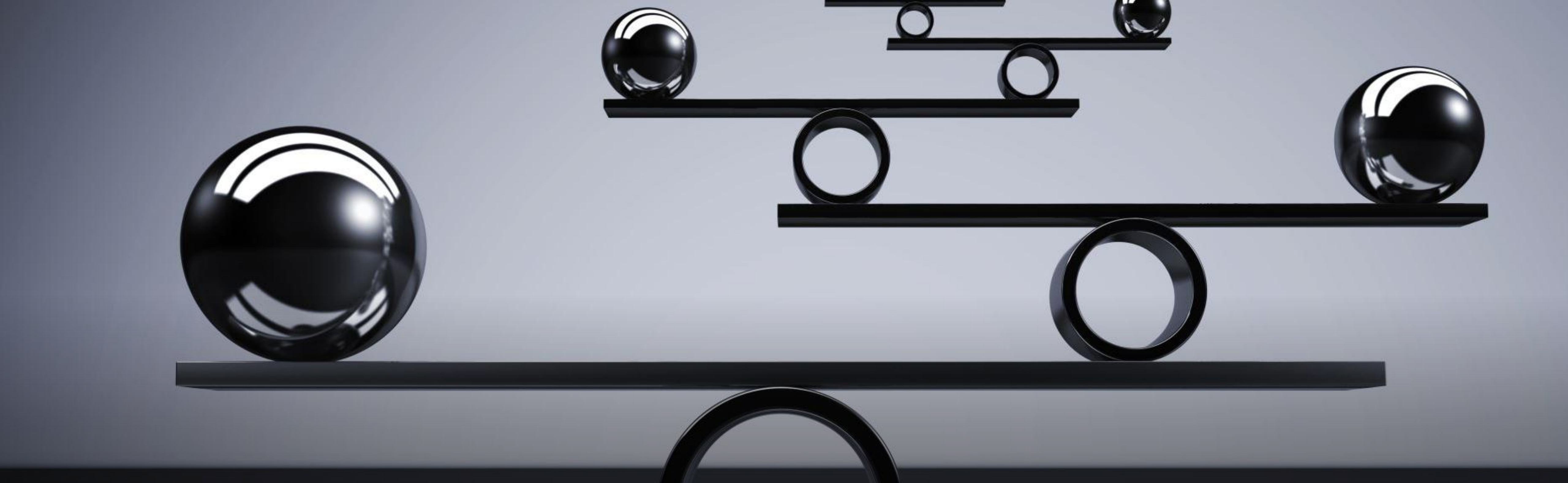


Consistency Models:
One-Step Image Generation

ICML 2023 Paper by OpenAI | Presented by Mary-Brenda Akoda

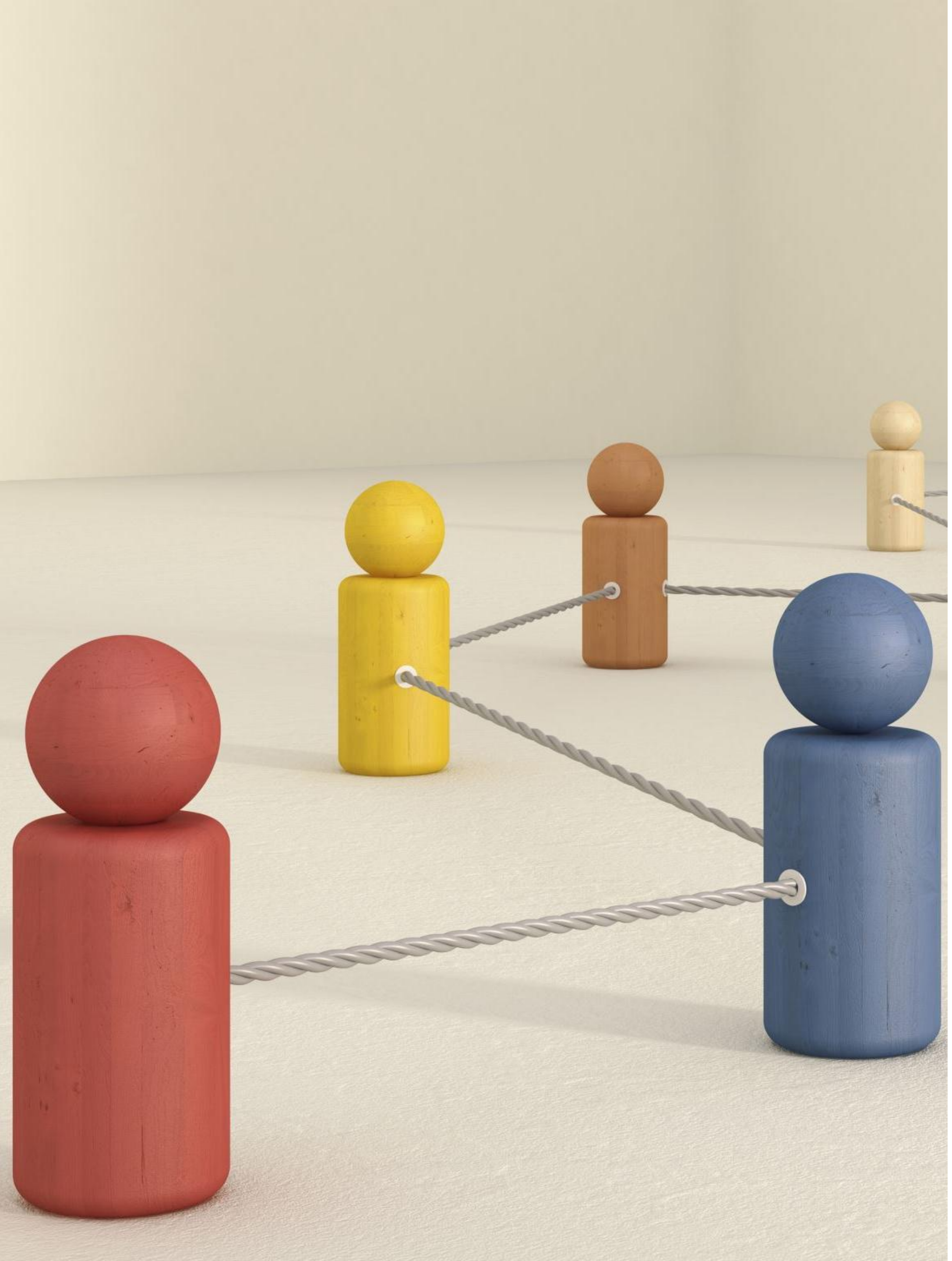


Table of Contents

Introduction

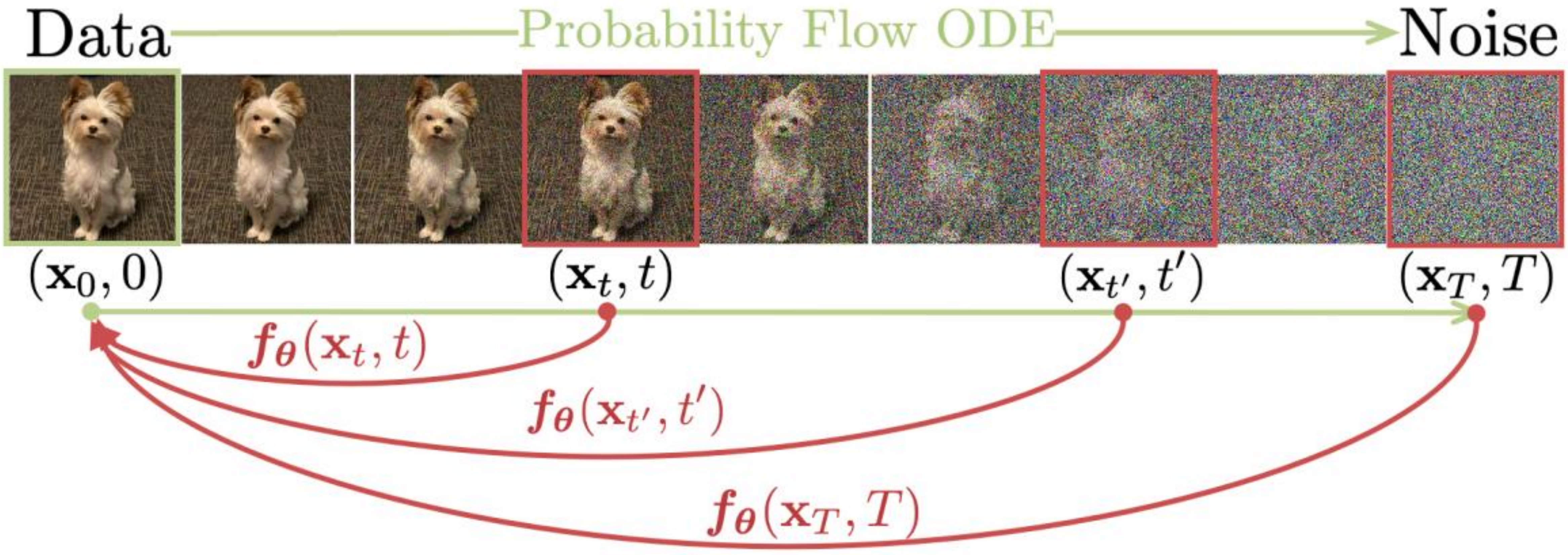
Generative models learn high-dimensional data distributions (e.g., images)

Applications: Image synthesis, inpainting, super-resolution, image denoising

Diffusion Models (DMs): Powerful but **slow sampling** (100s - 1000s steps) so not feasible for real-time application (e.g., MRI reconstruction).

Consistency Models (CMs): Directly map noisy data to clean data for **one-step generation**. Allows multistep sampling for **quality-compute trade-off**.

Consistency Models Concept



$f_\theta(x_t, t) = f_\theta(x_{t'}, t') \quad \text{for any } (x_t, x_{t'}) \text{ on the same PF ODE trajectory}$

$f_\theta(x_\varepsilon, \varepsilon) = x_\varepsilon$ *(boundary condition)*

Core Mathematical Foundation

Forward Diffusion as SDE[1]:

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

where $t \in [0, T]$; $\mu(x_t, t)$, $\sigma(t)$ = drift and diffusion coefficients, w_t = standard Brownian motion.

Probability Flow ODE (Reverse Diffusion):

- Same marginal distributions as original SDE, enabling deterministic transformations.

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

where $\nabla \log p_t(x_t)$ is called the score function of $p_t(x_t)$

Simplification to empirical PF ODE[2]:

- Applied $\mu(x,t) = 0$ and $\sigma(t) = \sqrt{2t}$

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

1. Y. Song, et al, “Score-based generative modelling through stochastic differential equations,” ICLR, 2020.
2. T. Karras, M. Aittala, T. Aila, and S. Laine. “Elucidating the design space of diffusion-based generative models,” NeurIPS, 2022.

Training Method 1: Consistency Distillation (CD)

1. Start with pre-trained model
2. Sample noisy x at time, t_{n+1}
3. Get teacher's estimate at t_n
4. Minimise output differences between adjacent points
5. Update online network and target networks.

Algorithm 2 Consistency Distillation (CD)

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

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

repeat

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

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

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

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

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

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

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

until convergence

Training Method 2: Consistency Training (CT)

- **No teacher**: Uses score matching technique (unbiased estimator)
- **Progressive schedules, (N) and μ**:
  - Small N(k) (bigger Δt) → Faster initial learning; higher bias.
  - Large N(k) (smaller Δt) → Higher precision in later training. Higher variance, better results.
  - Slow updates of θ^{-} in later training → stabilises learning & reduces sensitivity to small fluctuations in θ.

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

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

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

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

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

  $k \leftarrow k + 1$

until convergence

Sampling with Consistency Models

One-Step Sampling:

- Directly map noise to clean data, $\hat{x}_\varepsilon$, across all time steps using $f_\theta(x_T,T)$

Multi-Step Sampling:

- Balances **speed** and **quality**.
- Alternates between denoising and noise injection at each step: to maintain smooth transitions and avoid instability.
- Controlled Noise Injection: Scaling factor ensures noise matches time step's level.

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

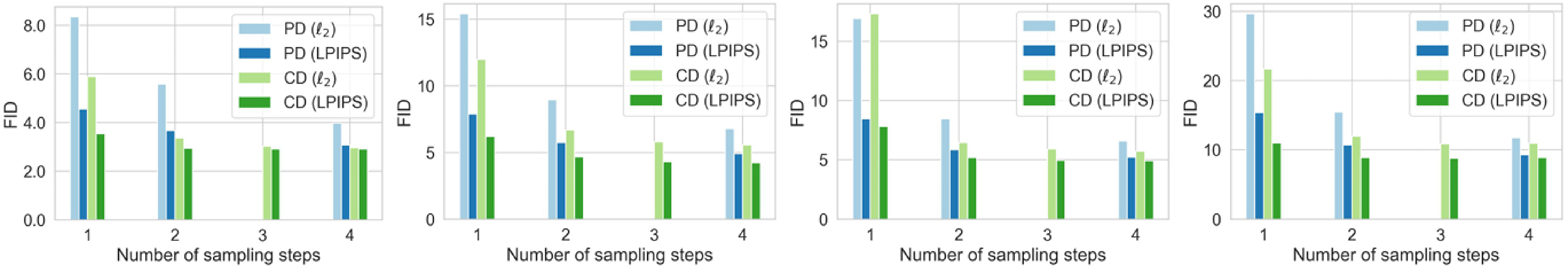
  $\hat{\mathbf{x}}_{\tau_n} \leftarrow \mathbf{x} + \sqrt{\tau_n^2 - \epsilon^2}\mathbf{z}$

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

end for

Output: $\mathbf{x}$

Results



(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.

1-Step FID: 3.55, 2-step FID: 2.93 (CIFAR-10) | 1-Step FID: 6.20, 2-step FID: 4.70 (ImageNet)

Results

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**

Results

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

Results

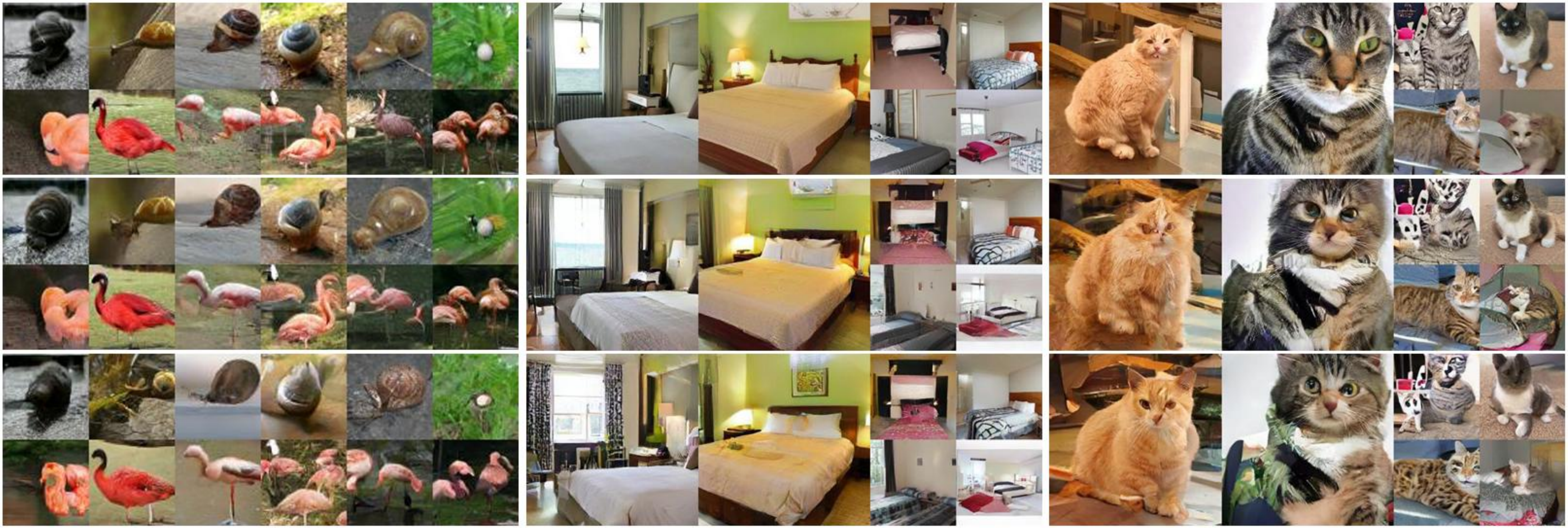


Figure 5: Samples generated by EDM (*top*), CT + single-step generation (*middle*), and CT + 2-step generation (*Bottom*). All corresponding images are generated from the same initial noise.

Advantages

- Fast, efficient one-step generation
- Quality-compute trade-off for multistep generation
- Zero-shot editing capabilities
- Two Training Modes: distillation and as standalone
- Outperforms existing distillation techniques and doesn’t require synthetic datasets
- Better samples than existing single-step generation models (except for some GANs)

Limitations

Distillation limits quality to that of the pre-trained model.

LPIPS introduced undesirable biases in evaluation, affecting the perceived quality of generated samples.

High computational resources: required for training.

Not always state-of-the-art: Sample quality can lag behind fully iterative diffusion or very large GANs.

Future Work

Improved Techniques for Training Consistency Models (ICLR 2024 | Oral)

- **Improved Consistency Training (iCT)**: learns directly from data without distillation.
- **Removed EMA for teacher network**: led to significant improvement in FIDs.
- **Pseudo-Huber Losses**: replaces LPIPS, reducing bias in evaluation.
- **Lognormal Noise Schedule**: as CT objective, improving sample quality & efficiency.
- **Improved Sample Quality**: 4x over CT, better FID scores, and **surpassed CD**.

Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models (Preprint Oct. 2024)

Simplified, Unified Theoretical Formulation: to identify root causes of training instability.

Improved Network Architecture and Training Objectives: for stable and scalable training.

Large-Scale Model Training: trained largest CM with up to 1.5B parameters on ImageNet 512x512.

Efficient Sampling: Quality comparable to leading diffusion models using only 2 steps (~50x speedup; 0.11s for 1 sample).

Narrowed FID gap with teacher: to within 10% in 2 steps.

Beyond OpenAI

Consistency Models Made Easy (by CMU | ICLR 2025):

- Easy Consistency Tuning: makes training CMs cost-effective and more accessible (CIFAR-10: 1 hour on 1 A100 vs. 1 week on 8 A100s).

Consistency Trajectory Models (by Sony AI | ICLR 2024):

- Generalises CMs and DMs, for efficient traversal along PF ODE.
- Flexible Sampling: supports deterministic and stochastic.
- SoTA FID for 1-step sampling on CIFAR-10 (FID 1.73) and ImageNet (FID 1.92).
- Beats EDM (35 NFE) and StyleGAN-XL. **Achieves student-beats-teacher**.

Conclusion

Motivation

- Diffusion Models need many iterative steps → slow sampling.
- Consistency Models aim for **fast one-step generation** without zero-shot editing and sample quality.

Key Ideas

- **Self-Consistency**: Any noisy version of a data point (at different times) maps back to the same clean sample.
- **Consistency Distillation**: Uses a pretrained diffusion model; 1-step approx. a teacher's multi-step ODE path.
- **Consistency Training**: from scratch by enforcing consistency on multiple noise levels of same data, no teacher.
- Architecture: Enforces a boundary condition at near-zero noise.

Advantages

- One-step or Few step Generation (**potential for real-time applications**), Zero-Shot Editing, Comparable (or Better) Quality, No synthetic data needed.

Additional theoretical + practical refinements **under active development**.

References

- Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modelling through stochastic differential equations,” *ICLR*, 2020.
- T. Karras, M. Aittala, T. Aila, and S. Laine. “Elucidating the design space of diffusion-based generative models,” *NeurIPS*, 2022.
- Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” *ICML*. PMLR, 2023.
- Y. Song and P. Dhariwal, “Improved techniques for training consistency models,” *ICLR*, 2024.
- Z. Geng, A. Pokle, W. Luo, J. Lin, and J. Z. Kolter, “Consistency models made easy,” *ICLR* 2025.
- C. Lu and Y. Song, “Simplifying, stabilizing and scaling continuous-time consistency mod-els,” arXiv preprint arXiv:2410.11081, 2024
- D. Kim, C-H. Lai, et al. “Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion”, ICLR 2024.