Improved Techniques for Training Consistency Models

Yang Song & Prafulla Dharwal
OpenAI

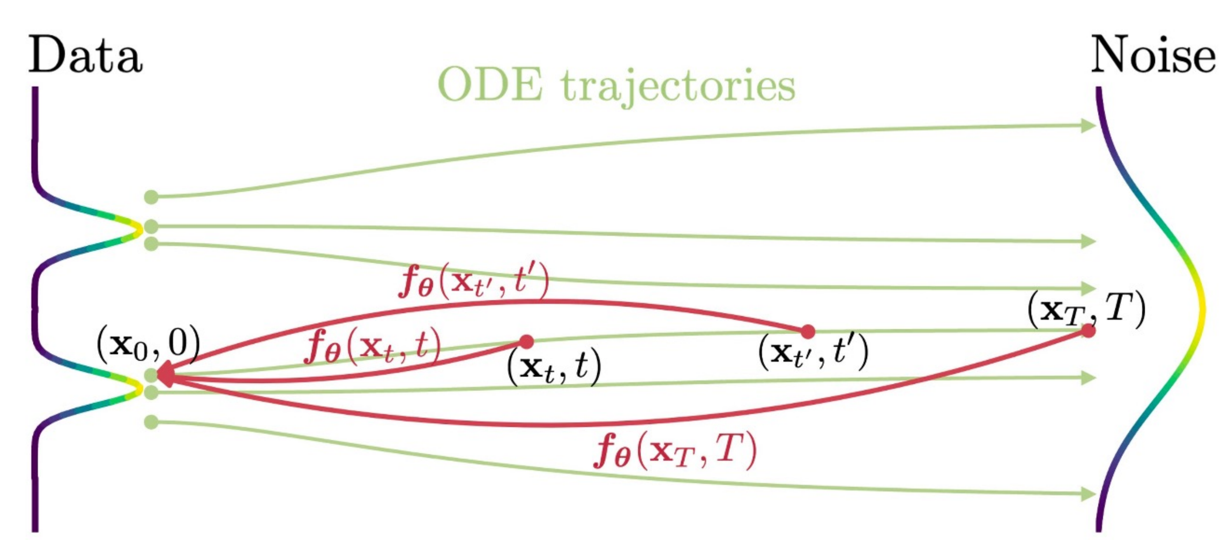
ABSTRACT

- Consistency models can produce high quality samples in one step. They are currently best trained by distillation from pre-trained diffusion models.
- We present several techniques to improve consistency training, a method that learns consistency models directly from data without relying on distillation, making them independent generative models.
- Our techniques include new weighting functions, noise schedules, time embeddings, removing EMA from the teacher network, and replacing LPIPS with pseudo-Huber losses.
- We achieve FID=2.51 on CIFAR-10 and FID=3.25 on ImageNet 64x64 with one step sampling, which improves prior consistency training by a factor of 3.5x and 4x respectively, surpassing diffusion distillation methods.

Paper

Consistency Models

Intuition: score-based diffusion models define a one-to-one mapping between noise and data through the probability flow ODE. Consistency models learn to estimate this mapping directly.



Definition:

$$\forall t \in [0, T]: \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \mathbf{x}_0$$

Parameterization: Need to meet the boundary condition at $t = 0$.

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

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

Sampling: one-step or multi-step.

Training:

- Consistency Distillation (CD)

One ODE solver step using pretrained diffusion model

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

Weighting function — Student model — Teacher model $\boldsymbol{\theta}^- = \mathrm{EMA}(\boldsymbol{\theta})$

- **Consistency Training (CT)**
  - More accurate when $|t_{n+1} - t_n| \to 0$
  - Use a schedule to progressively reduce time gaps.

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

- Continuous-Time Consistency Training

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

Improving Consistency Training

Weighting functions, time embeddings & dropout

- Larger weighting for smaller noise.

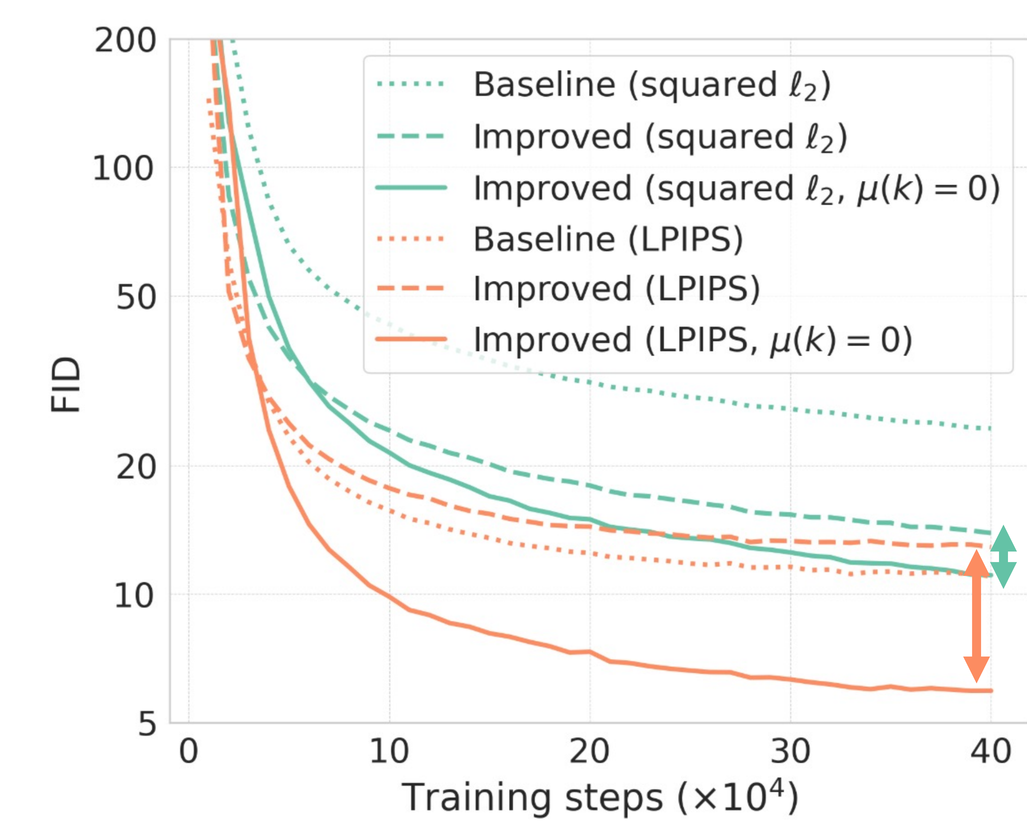
$$\lambda(t_n) = \frac{1}{t_{n+1} - t_n}$$

- Reducing the sensitivity of time embeddings
  - Continuous-time CT converges only when the sensitivity of time embeddings is sufficiently small.
- Need larger dropout than diffusion models

Removing EMA in teacher network:

- Use zero EMA in the teacher network during consistency training for better theoretical soundness.

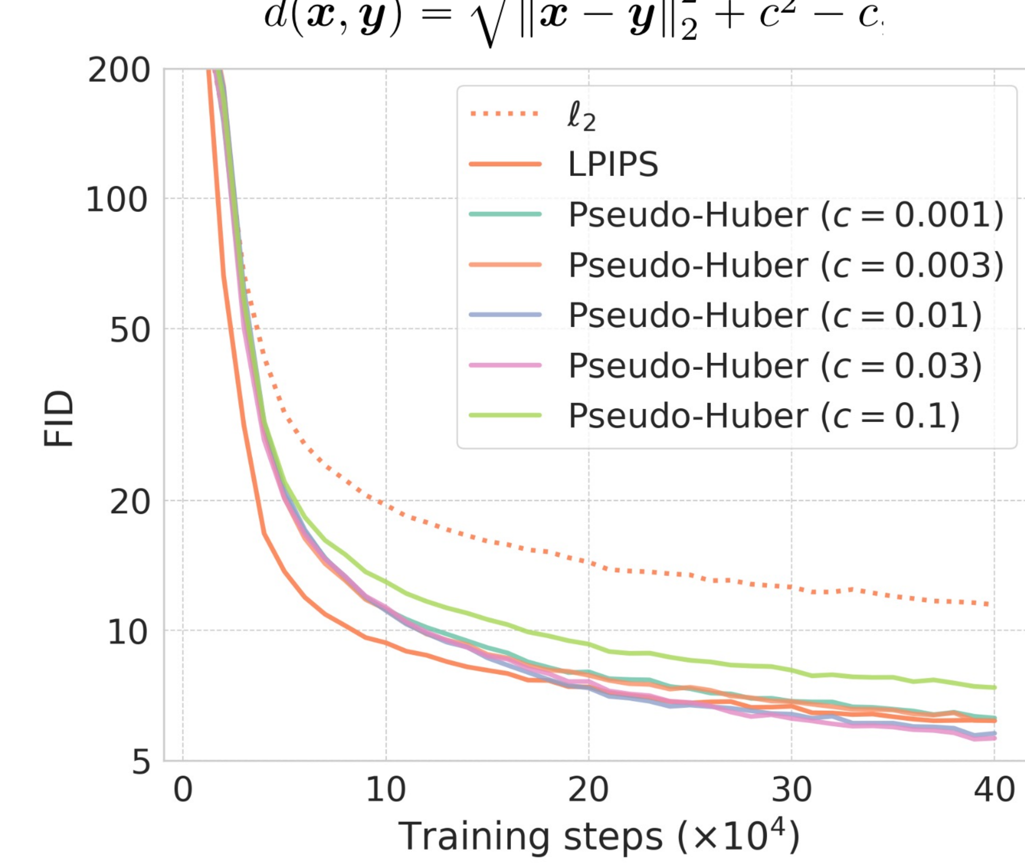
$$\boldsymbol{\theta}^- = \mathrm{stopgrad}(\boldsymbol{\theta})$$



Minimizing pseudo-Huber loss:

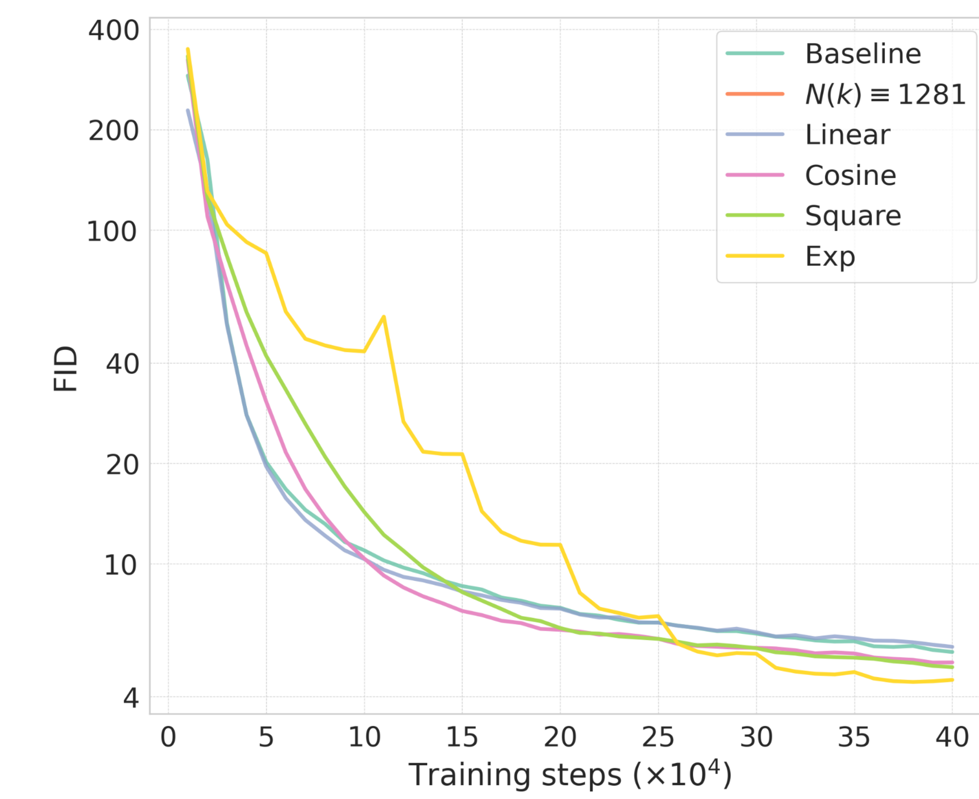
- Pseudo-Huber loss outperforms both squared loss and LPIPS in consistency training.

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



Improved time schedules

- Doubling the total number of time steps for every fixed number of training iterations.



- Sampling each individual time step according to a discretized lognormal distribution.

Putting it together

- Outperforms best-in-class diffusion distillation techniques
- Competitive with diffusion models and GANs
- Balanced precision and diversity; no mode collapse.

Results on ImageNet 64x64

METHOD	NFE (↓)	FID (↓)	Prec. (↑)	Rec. (↑)
Fast samplers & distillation for diffusion models				
Diff-Instruct (Luo et al., 2023)	1	5.57		
PD* (Salimans & Ho, 2022)	1	15.39	0.59	0.62
	2	8.95	0.63	0.65
	4	6.77	0.66	0.65
PD (LPIPS) (Song et al., 2023)	1	7.88	0.66	0.63
	2	5.74	0.67	0.65
	4	4.92	0.68	0.65
CD (LPIPS) (Song et al., 2023)	1	6.20	0.68	0.63
	2	4.70	0.69	0.64
	3	4.32	0.70	0.64
Direct Generation				
RIN (Jabri et al., 2023)	1000	1.23		
DDPM (Ho et al., 2020)	250	11.0	0.67	0.58
iDDPM (Nichol & Dhariwal, 2021)	250	2.92	0.74	0.62
ADM (Dhariwal & Nichol, 2021)	250	2.07	0.74	0.63
EDM (Karras et al., 2022)	511	1.36		
EDM* (Heun) (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 (LPIPS) (Song et al., 2023)	1	13.0	0.71	0.47
	2	11.1	0.69	0.56
iCT (ours)	1	4.02	0.70	0.63
	2	3.20	0.73	0.63
iCT-deep (ours)	1	3.25	0.72	0.63
	2	2.77	0.74	0.62