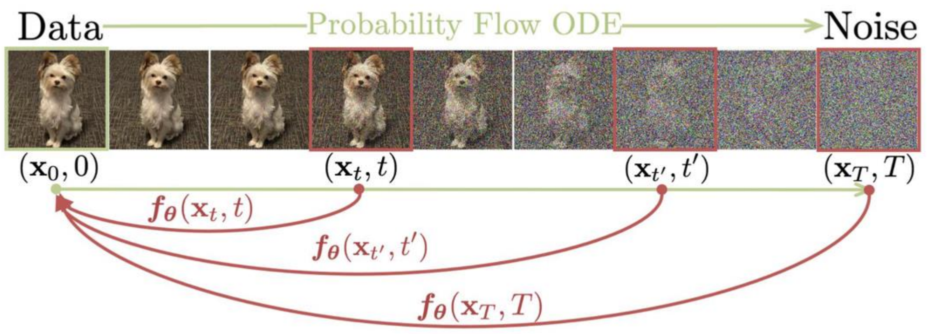
Continuous-Time Consistency Models

Guande He

2024.11.18

TEXAS
The University of Texas at Austin



The Curse of Consistency

- From an optimization perspective, it's hard to deal with when $\Delta t \to 0$ due to error accumulation:

$$\|\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_T) - \mathbf{x}_0\| \leq \sum_{i=1}^{N-1} \|\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{i+1}}) - \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_i})\| \leq N e_{\max}$$

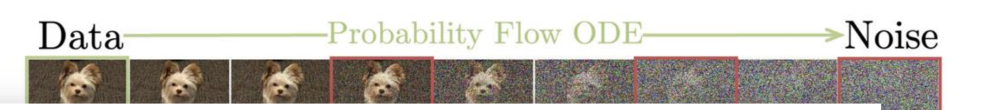
  - This $\Delta t \to 0$ condition is **required** to guarantee the correctness of the "data score" used in consistency training, i.e., the marginal score is estimated with:

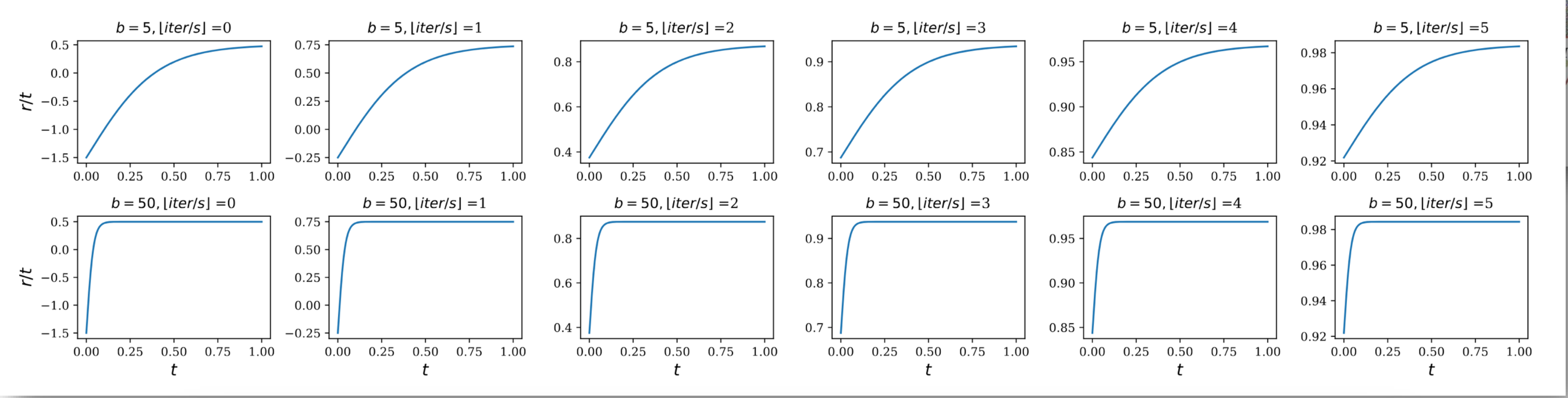
$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = -\mathbb{E}\left[\frac{\mathbf{x}_t - \alpha_t \mathbf{x}_t}{\sigma_t^2} \mid \mathbf{x}_t\right] \approx \frac{\mathbf{x}_t - \alpha_t \mathbf{x}_t}{\sigma_t^2}$$

- An expedient treatment is to manually design a "time step schedule" to gradually "shrink" Δt .
  - 😵💫 But…

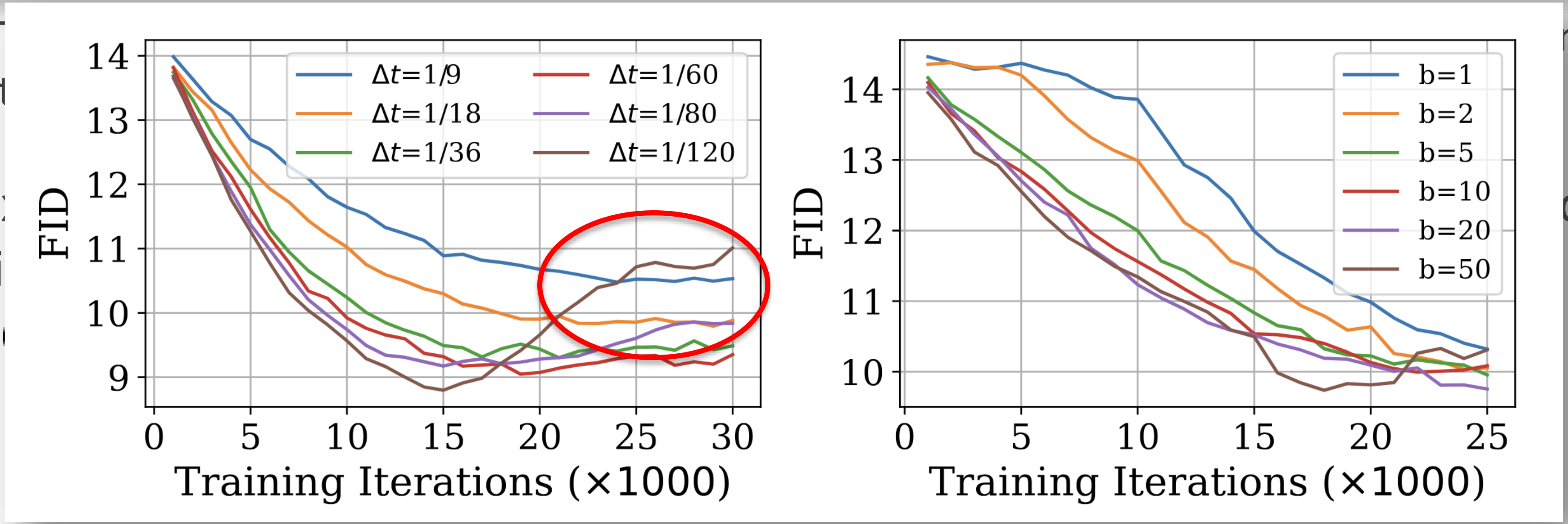
Geng, Zhengyang, et al. "Consistency Models Made Easy." (2024).

TEXAS
The University of Texas at Austin





- ➢ T… nsistency
  t…
- An e… dually
  “shri…
  - ➢ (…



He, Guande, et al. "Consistency Diffusion Bridge Models." (NeurIPS 2024)

Continuous-Time Consistency Models

- What happens to the objective when $\Delta t \to 0$? (From finite-difference to differential)
  - Recall Consistency Distillation (in L2 distance, VE noise schedule):

$$\mathcal{L}_{\text{CD}}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}[\lambda(t_n) \| \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}) - \boldsymbol{f}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}, t_n) \|_2^2]$$

  - We have $\lim_{N \to \infty} (N-1)^2 \mathcal{L}_{\text{CD}}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) = \mathcal{L}_{\text{CD}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi})$, where:

$$\mathcal{L}_{\text{CD}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}[\frac{\lambda(t)}{[(\tau^{-1})'(t)]^2} \left\| \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} - t \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\mathbf{x}_t} \boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \right\|_2^2]$$

- This is intuitive since

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \equiv \mathbf{x}_\epsilon \longrightarrow \textit{consistency condition}$$

$$\iff \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} + \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} \equiv 0$$

$$\iff \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} [-t \boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)] + \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} \equiv 0 \qquad \textit{(VE noise schedule)}$$

$$\iff \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} - t \frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \boldsymbol{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \equiv 0.$$

Song, Yang, et al. "Consistency Models." (ICML 2023)

Continuous-Time Consistency Models

- The practice on discrete-time CMs suggests that using $\mathcal{L}_{\mathrm{CD}}^{N}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]; \phi)$ instead of $\mathcal{L}_{\mathrm{CD}}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}; \phi)$ stabilizes training.
- For continuous-time CM (in L2 distance, VE noise schedule):

$$\mathcal{L}_{\mathrm{CD}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}\left[\frac{\lambda(t)}{[(\tau^{-1})'(t)]^2}\left\|\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial t} - t\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} s_{\boldsymbol{\phi}}(\mathbf{x}_t, t)\right\|_2^2\right]$$

$$\mathcal{L}_{\mathrm{CD}}^{\infty}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]; \boldsymbol{\phi}) := 2\mathbb{E}\left[\frac{\lambda(t)}{[(\tau^{-1})'(t)]^2}\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)^{\top}\left(\frac{\partial \boldsymbol{f}_{\mathrm{sg}[\boldsymbol{\theta}]}(\mathbf{x}_t, t)}{\partial t} - t\frac{\partial \boldsymbol{f}_{\mathrm{sg}[\boldsymbol{\theta}]}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} s_{\boldsymbol{\phi}}(\mathbf{x}_t, t)\right)\right]$$

$$\mathcal{L}_{\mathrm{CT}}^{\infty}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]) := 2\mathbb{E}\left[\frac{\lambda(t)}{[(\tau^{-1})'(t)]^2}\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)^{\top}\left(\frac{\partial \boldsymbol{f}_{\mathrm{sg}[\boldsymbol{\theta}]}(\mathbf{x}_t, t)}{\partial t} - t\frac{\partial \boldsymbol{f}_{\mathrm{sg}[\boldsymbol{\theta}]}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \cdot \frac{\mathbf{x}_t - \mathbf{x}}{t}\right)\right]$$

- Asymptotic behavior:

No stop gradient version:

$$\lim_{N\to\infty}(N-1)^2\mathcal{L}_{\mathrm{CD}}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi}) = \mathcal{L}_{\mathrm{CD}}^{\infty}(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\phi})$$

Stop gradient version:

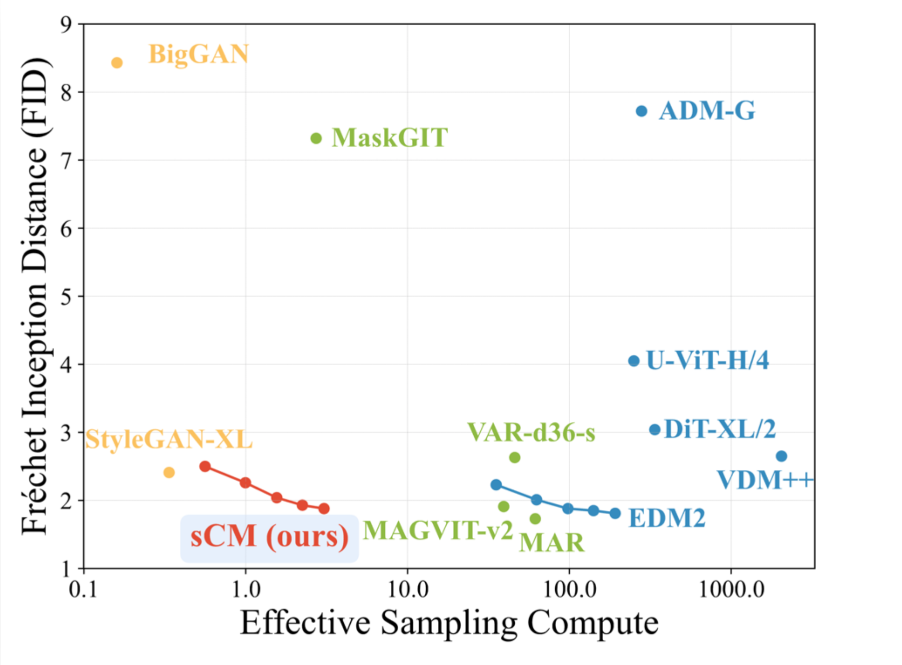
$$\lim_{N\to\infty}(N-1)^2\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CD}}^{N}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]; \boldsymbol{\phi}) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CD}}^{\infty}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]; \boldsymbol{\phi})$$

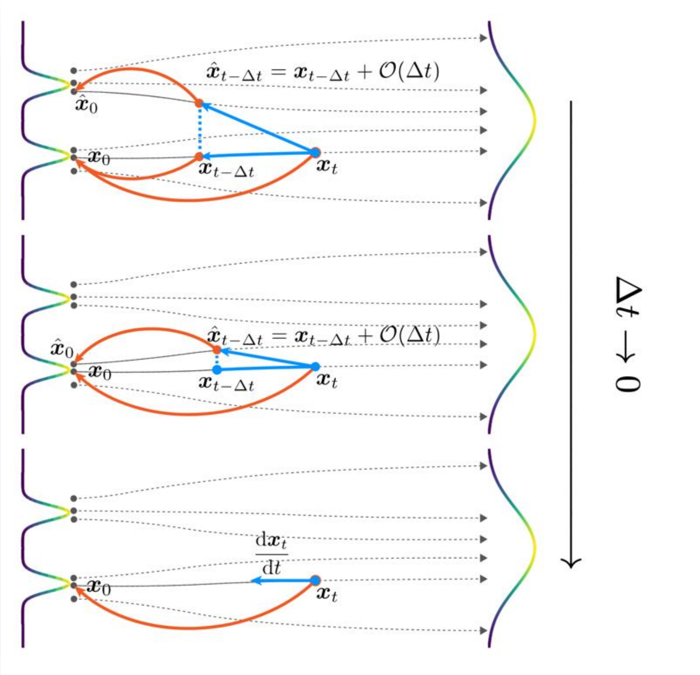
$$= \lim_{N\to\infty}(N-1)^2\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CT}}^{N}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}]) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{CT}}^{\infty}(\boldsymbol{\theta}, \mathrm{sg}[\boldsymbol{\theta}])$$

Song, Yang, et al. "Consistency Models." (ICML 2023)

Continuous-Time CMs can work!

- Although the continuous-time CM formulation is proposed on early 2023, there is no empirical practice successfully showing its effectiveness until Oct. 2024.
  - ➢ Developing empirical & engineering techniques tailored for the continuous-time CM objective!





Lu, Cheng, and Yang Song. “Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models.” (2024).

Noise Schedule, Model Parameterization & Network Preconditioning (Empirical Design Space)

- Forward process / interpolation:

$$\boldsymbol{x}_t = \cos(t)\boldsymbol{x}_0 + \sin(t)\boldsymbol{z}, \quad \boldsymbol{x}_0 \sim p_d(\boldsymbol{x}_0), \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \sigma_d^2 \boldsymbol{I}), t \in [0, \frac{\pi}{2}]$$

- The diffusion model (i.e., the velocity field in Rectified Flow) is parameterized as:

$$\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \sigma_d \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{x}_t/\sigma_d, c_{\text{noise}}(t))$$

where $\boldsymbol{F}_{\boldsymbol{\theta}}$ is a neural network. The PF-ODE is given by: $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \sigma_d \boldsymbol{F}_{\boldsymbol{\theta}}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$

- Parameterization of consistency function:

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \cos(t)\boldsymbol{x}_t - \sin(t)\sigma_d \boldsymbol{F}_{\boldsymbol{\theta}}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, c_{\text{noise}}(t)\right)$$

  - Insight here: using the DDIM-style first-order ODE discretization will automatically enforce the boundary condition $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_0, 0) \equiv 0$.
  - Note: $\boldsymbol{f}_{\boldsymbol{\theta}}$ is proportional to $\sin(t)\boldsymbol{F}_{\boldsymbol{\theta}}$.

Lu, Cheng, and Yang Song. "Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models." (2024).

Stabilizing Continuous-time CMs

$$\frac{\mathrm{d}\boldsymbol{f_\theta}(\boldsymbol{x}_t,t)}{\mathrm{d}t} = \frac{\partial \boldsymbol{f_\theta}(\boldsymbol{x}_t,t)}{\partial \boldsymbol{x}_t}\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \frac{\partial \boldsymbol{f_\theta}(\boldsymbol{x}_t,t)}{\partial t} \equiv 0$$

- Consider the "consistency condition" (tangent) term under the proposed design:

$$\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}t} = -\cos(t)\left(\sigma_d \boldsymbol{F}_{\boldsymbol{\theta}^-}\left(\frac{\boldsymbol{x}_t}{\sigma_d}\right) - \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}\right) - \sin(t)\left(\boldsymbol{x}_t + \sigma_d \frac{\mathrm{d}\boldsymbol{F}_{\boldsymbol{\theta}^-}(\frac{\boldsymbol{x}_t}{\sigma_d},t)}{\mathrm{d}t}\right), \quad \boldsymbol{\theta}^- := \mathrm{sg}[\boldsymbol{\theta}]$$

  - The instability mainly comes from the scaled time derivative of the neural network $\sin(t)\partial_t \boldsymbol{F}_{\boldsymbol{\theta}^-}$:

$$\sin(t)\partial_t \boldsymbol{F}_{\boldsymbol{\theta}^-} = \sin(t)\frac{\partial c_{\text{noise}}(t)}{\partial t}\cdot\frac{\partial \mathrm{emb}(c_{\text{noise}})}{\partial c_{\text{noise}}}\cdot\frac{\partial \boldsymbol{F}_{\boldsymbol{\theta}^-}}{\partial \mathrm{emb}(c_{\text{noise}})}$$

- Proposed treatment:
  - Identity Time Transformation: $c_{\text{noise}}(t) = t$
  - Re-design time embeddings $\mathrm{emb}(c)$ with smaller gradient magnitudes.
  - Adaptative Double Normalization Layer:

$$\boldsymbol{y} = \mathrm{norm}(\boldsymbol{x}) \odot \mathrm{pnorm}(\boldsymbol{s}(t)) + \mathrm{pnorm}(b(t))$$

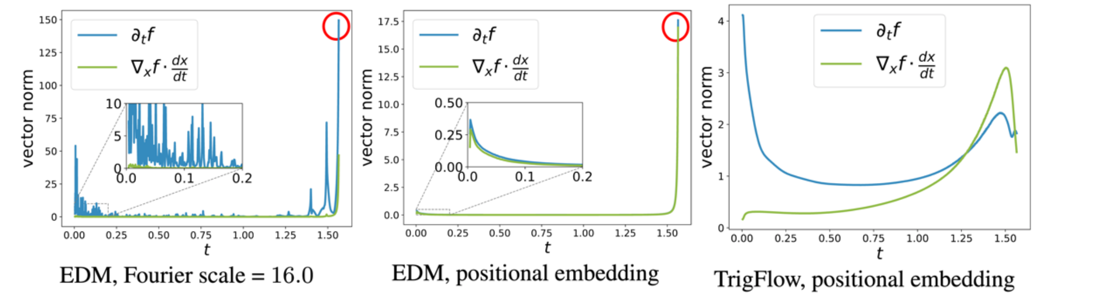
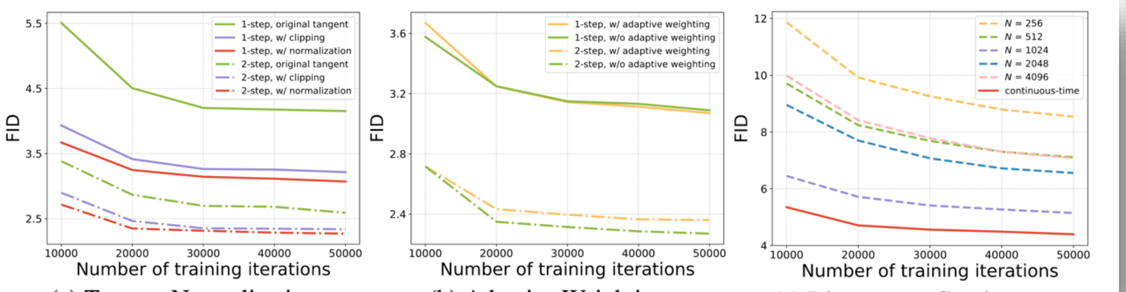


Figure 4: **Stability of different formulations.** We show the norms of both terms in $\frac{\mathrm{d}\boldsymbol{f}_{\theta^-}}{\mathrm{d}t} = \nabla_{\boldsymbol{x}}\boldsymbol{f}_{\theta^-}\cdot\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \partial_t \boldsymbol{f}_{\theta^-}$ for diffusion models trained with the EDM ($c_{\text{noise}}(t) = \log(\sigma_d \tan(t))$) and TrigFlow ($c_{\text{noise}}(t) = t$) formulations using different time embeddings. We observe that large Fourier scales in Fourier embeddings cause instabilities. In addition, the EDM formulation suffers from numerical issues when $t \to \frac{\pi}{2}$, while TrigFlow (using positional embeddings) has stable partial derivatives for both $\boldsymbol{x}_t$ and t.

Lu, Cheng, and Yang Song. "Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models." (2024).

Stabilizing Continuous-time CMs



(a) Tangent Normalization (b) Adaptive Weighting (c) Discrete vs. Continuous

- **Tangent Normalization**
  - ➢ Explicitly normalizing $\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}t}$ with $\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}t} / \left(\|\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}t}\| + c\right)$
  - ➢ Clipping $\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}}{\mathrm{d}t}$ within $[-1, 1]$.
- **Loss Trick & Adaptive Weighting**
  - ➢ Convert $\mathcal{L}_{\mathrm{CM}}^{\infty}$ into a MSE loss using the trick $\nabla_{\boldsymbol{\theta}}\mathbb{E}[\boldsymbol{F}_{\boldsymbol{\theta}}^{\top}\boldsymbol{y}] = \frac{1}{2}\nabla_{\boldsymbol{\theta}}\mathbb{E}[\|\boldsymbol{F}_{\boldsymbol{\theta}} - \boldsymbol{F}_{\mathrm{sg}[\boldsymbol{\theta}]} + \boldsymbol{y}\|_2^2]$.
  - ➢ Learn adaptive loss weighting during training:

$$\mathcal{L}_{\mathrm{sCM}}(\boldsymbol{\theta}, \phi) := \mathbb{E}_{\boldsymbol{x}_t, t}\left[\frac{e^{\omega_\phi(t)}}{D_{\boldsymbol{x}_0}}\left\|\boldsymbol{F}_{\boldsymbol{\theta}}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, t\right) - \boldsymbol{F}_{\boldsymbol{\theta}^-}\left(\frac{\boldsymbol{x}_t}{\sigma_d}, t\right) - \cos(t)\frac{\mathrm{d}\boldsymbol{f}_{\theta^-}(\boldsymbol{x}_t, t)}{\mathrm{d}t}\right\|_2^2 - \omega_\phi(t)\right]$$

- **Diffusion Fine-tuning & Tangent Warmup**
  - ➢ Fine-tuning from pre-trained diffusion models instead of training of scratch.
  - ➢ Warm up the instable term $\sin(t)(\boldsymbol{x}_t + \sigma_d \frac{\mathrm{d}\boldsymbol{F}_{\boldsymbol{\theta}^-}}{\mathrm{d}t})$ by using $r \cdot \sin(t)$, r increases linearly from 0 to 1 over the first 100k training iterations.

Efficient Jacobian-Vector Product (JVP) Computation

- **JVP Rearrangement**
  - Vanilla calculation of $\frac{\mathrm{d}\boldsymbol{F}_{\theta^-}}{\mathrm{d}t} = \nabla_{\boldsymbol{x}_t}\boldsymbol{F}_{\boldsymbol{\theta}^-} \cdot \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \partial_t \boldsymbol{F}_{\boldsymbol{\theta}^-}$ using JVP of $\boldsymbol{F}_{\boldsymbol{\theta}^-}(\frac{\cdot}{\sigma_d}, \cdot)$ with input $(\boldsymbol{x}_t, t)$ and tangent vector $(\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}, 1)$ is prone to overflow.
  - Using the fact that $\boldsymbol{f}_{\boldsymbol{\theta}}$ is proportional to $\sin(t)\boldsymbol{F}_{\boldsymbol{\theta}}$ and the sCM loss calculates $\cos(t)\frac{\mathrm{d}\boldsymbol{f}_{\boldsymbol{\theta}^-}(\boldsymbol{x}_t,t)}{\mathrm{d}t}$, the JVP can be implemented as:

$$\cos(t)\sin(t)\frac{\mathrm{d}\boldsymbol{F}_{\boldsymbol{\theta}^-}}{\mathrm{d}t} = \left(\nabla_{\frac{\boldsymbol{x}_t}{\sigma_d}}\boldsymbol{F}_{\theta^-}\right) \cdot \left(\cos(t)\sin(t)\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}\right) + \partial_t \boldsymbol{F}_{\boldsymbol{\theta}^-} \cdot (\cos(t)\sin(t)\sigma_d)$$

which is the JVP of $\boldsymbol{F}_{\boldsymbol{\theta}^-}(\cdot, \cdot)$ with input $(\frac{\boldsymbol{x}_t}{\sigma_d}, t)$ with tangent vector $(\cos(t)\sin(t)\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}, \cos(t)\sin(t)\sigma_d)$

- The authors also modifies the Flash Attention to simultaneously compute softmax self-attention and its JVP in a single forward pass.

Lu, Cheng, and Yang Song. "Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models." (2024).

Benchmark Results

Table 1: Sample quality on unconditional CIFAR-10 and class-conditional ImageNet 64× 64.

Unconditional CIFAR-10		
METHOD	**NFE (↓)**	**FID (↓)**
Diffusion models & Fast Samplers		
Score SDE (deep) (Song et al., 2021b)	2000	2.20
EDM (Karras et al., 2022)	35	2.01
Flow Matching (Lipman et al., 2022)	142	6.35
DPM-Solver (Lu et al., 2022a)	10	4.70
DPM-Solver++ (Lu et al., 2022b)	10	2.91
DPM-Solver-v3 (Zheng et al., 2023c)	10	2.51
Joint Training		
Diffusion GAN (Xiao et al., 2022)	4	3.75
Diffusion StyleGAN (Wang et al., 2022)	1	3.19
StyleGAN-XL (Sauer et al., 2022)	1	1.52
CTM (Kim et al., 2023)	1	**1.87**
Diff-Instruct (Luo et al., 2024)	1	4.53
DMD (Yin et al., 2024b)	1	3.77
SiD (Zhou et al., 2024)	1	1.92
Diffusion Distillation		
DFNO (LPIPS) (Zheng et al., 2023b)	1	3.78
2-Rectified Flow (Liu et al., 2022)	1	4.85
PID (LPIPS) (Tee et al., 2024)	1	3.92
BOOT (LPIPS) (Gu et al., 2023)	1	4.38
Consistency-FM (Yang et al., 2024)	2	5.34
PD (Salimans & Ho, 2022)	1	8.34
	2	5.58
TRACT (Berthelot et al., 2023)	1	3.78
	2	3.32
CD (LPIPS) (Song et al., 2023)	1	**3.55**
	2	2.93
sCD (ours)	1	3.66
	2	**2.52**
Consistency Training		
iCT (Song & Dhariwal, 2023)	1	2.83
	2	2.46
iCT-deep (Song & Dhariwal, 2023)	1	**2.51**
	2	2.24
ECT (Geng et al., 2024)	1	3.60
	2	2.11
sCT (ours)	1	2.97
	2	**2.06**

Class-Conditional ImageNet 64×64		
METHOD	**NFE (↓)**	**FID (↓)**
Diffusion models & Fast Samplers		
ADM (Dhariwal & Nichol, 2021)	250	2.07
RIN (Jabri et al., 2022)	1000	1.23
DPM-Solver (Lu et al., 2022a)	20	3.42
EDM (Heun) (Karras et al., 2022)	79	2.44
EDM2 (Heun) (Karras et al., 2024)	63	1.33
Joint Training		
StyleGAN-XL (Sauer et al., 2022)	1	1.52
Diff-Instruct (Luo et al., 2024)	1	5.57
EMD (Xie et al., 2024b)	1	2.20
DMD (Yin et al., 2024b)	1	2.62
DMD2 (Yin et al., 2024a)	1	**1.28**
SiD (Zhou et al., 2024)	1	1.52
CTM (Kim et al., 2023)	1	1.92
	2	1.73
Moment Matching (Salimans et al., 2024)	1	3.00
	2	3.86
Diffusion Distillation		
DFNO (LPIPS) (Zheng et al., 2023b)	1	7.83
PID (LPIPS) (Tee et al., 2024)	1	9.49
TRACT (Berthelot et al., 2023)	1	7.43
	2	4.97
PD (Salimans & Ho, 2022)	1	10.70
(reimpl. from Heek et al. (2024))	2	4.70
CD (LPIPS) (Song et al., 2023)	1	6.20
	2	4.70
MultiStep-CD (Heek et al., 2024)	1	3.20
	2	1.90
sCD (ours)	1	**2.44**
	2	**1.66**
Consistency Training		
iCT (Song & Dhariwal, 2023)	1	4.02
	2	3.20
iCT-deep (Song & Dhariwal, 2023)	1	3.25
	2	2.77
ECT (Geng et al., 2024)	1	2.49
	2	1.67
sCT (ours)	1	**2.04**
	2	**1.48**

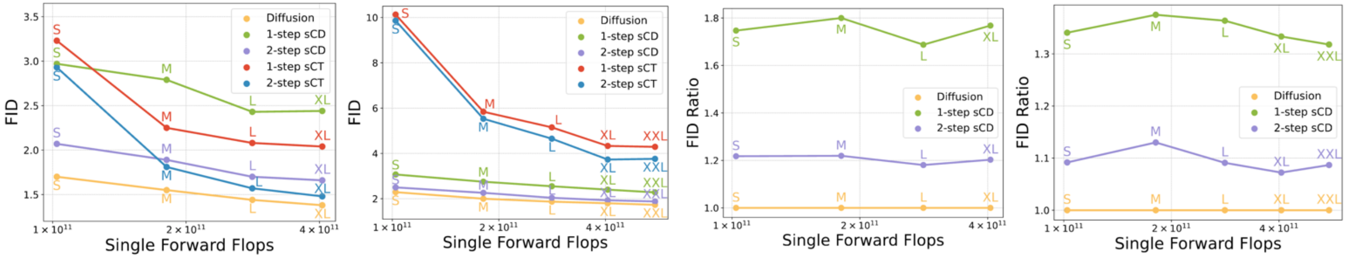
Table 2: Sample quality on class-conditional ImageNet 512× 512. †Our reimplemented teacher diffusion model based on EDM2 (Karras et al., 2024) but with modifications in Sec. 4.1.

METHOD	**NFE (↓)**	**FID (↓)**	**#Params**
Diffusion models			
ADM-G (Dhariwal & Nichol, 2021)	250×2	7.72	559M
RIN (Jabri et al., 2022)	1000	3.95	320M
U-ViT-H/4 (Bao et al., 2023)	250×2	4.05	501M
DiT-XL/2 (Peebles & Xie, 2023)	250×2	3.04	675M
SimDiff (Hoogeboom et al., 2023)	512×2	3.02	2B
VDM++ (Kingma & Gao, 2024)	512×2	2.65	2B
DiffiT (Hatamizadeh et al., 2023)	250×2	2.67	561M
DiMR-XL/3R (Liu et al., 2024)	250×2	2.89	525M
DiffuSSM-XL (Yan et al., 2024)	250×2	3.41	673M
DiM-H (Teng et al., 2024)	250×2	3.78	860M
U-DiT (Tian et al., 2024b)	250	15.39	204M
SiT-XL (Ma et al., 2024)	250×2	2.62	675M
Large-DiT (Alpha-VLLM, 2024)	250×2	2.52	3B
MaskDiT (Zheng et al., 2023a)	79×2	2.50	736M
DiS-H/2 (Fei et al., 2024a)	250×2	2.88	900M
DRWKV-H/2 (Fei et al., 2024b)	250×2	2.95	779M
EDM2-S (Karras et al., 2024)	63×2	2.23	280M
EDM2-M (Karras et al., 2024)	63×2	2.01	498M
EDM2-L (Karras et al., 2024)	63×2	1.88	778M
EDM2-XL (Karras et al., 2024)	63×2	1.85	1.1B
EDM2-XXL (Karras et al., 2024)	63×2	**1.81**	1.5B
GANs & Masked Models			
BigGAN (Brock, 2018)	1	8.43	160M
StyleGAN-XL (Sauer et al., 2022)	1×2	2.41	168M
VQGAN (Esser et al., 2021)	1024	26.52	227M
MaskGIT (Chang et al., 2022)	12	7.32	227M
MAGVIT-v2 (Yu et al., 2023)	64×2	1.91	307M
MAR (Li et al., 2024)	64×2	**1.73**	481M
VAR-d36-s (Tian et al., 2024a)	10×2	2.63	2.3B

METHOD	**NFE (↓)**	**FID (↓)**	**#Params**
†Teacher Diffusion Model			
EDM2-S (Karras et al., 2024)	63×2	2.29	280M
EDM2-M (Karras et al., 2024)	63×2	2.00	498M
EDM2-L (Karras et al., 2024)	63×2	1.87	778M
EDM2-XL (Karras et al., 2024)	63×2	1.80	1.1B
EDM2-XXL (Karras et al., 2024)	63×2	1.73	1.5B
Consistency Training (sCT, ours)			
sCT-S (ours)	1	10.13	280M
	2	9.86	280M
sCT-M (ours)	1	5.84	498M
	2	5.53	498M
sCT-L (ours)	1	5.15	778M
	2	4.65	778M
sCT-XL (ours)	1	4.33	1.1B
	2	3.73	1.1B
sCT-XXL (ours)	1	4.29	1.5B
	2	3.76	1.5B
Consistency Distillation (sCD, ours)			
sCD-S	1	3.07	280M
	2	2.50	280M
sCD-M	1	2.75	498M
	2	2.26	498M
sCD-L	1	2.55	778M
	2	2.04	778M
sCD-XL	1	2.40	1.1B
	2	1.93	1.1B
sCD-XXL	1	**2.28**	1.5B
	2	**1.88**	1.5B

Lu, Cheng, and Yang Song. “Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models.” (2024).

Scalability & Sample Diversity



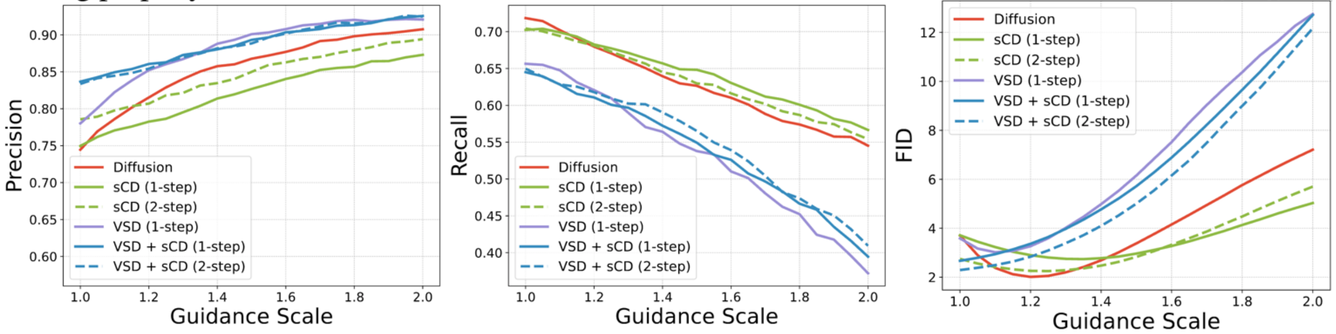
ImageNet 64× 64 ImageNet 512× 512

(a) FID (↓) as a function of single forward flops.

ImageNet 64× 64 ImageNet 512× 512

(b) FID ratio (↓) as a function of single forward flops.

Figure 6: **sCD scales commensurately with teacher diffusion models**. We plot the (a) FID and (b) FID ratio against the teacher diffusion model (at the same model size) on ImageNet 64×64 and 512×512. sCD scales better than sCT, and has a *constant offset* in the FID ratio across all model sizes, implying that sCD has the same scaling property of the teacher diffusion model. Furthermore, the offset diminishes with more sampling steps.



(a) Precision score (↑) (b) Recall score (↑) (c) FID score (↓)

Figure 7: **sCD has higher diversity compared to VSD**: Sample quality comparison of the EDM2 (Karras et al., 2024) diffusion model, VSD (Wang et al., 2024; Yin et al., 2024b), sCD, and the combination of VSD and sCD, across varying guidance scales. All models are of EDM2-M size and trained on ImageNet 512×512.

Lu, Cheng, and Yang Song. “Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models.” (2024).

Thanks!