Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion

Dongjun Kim[*,†]
Sony AI
Tokyo, Japan
dongjoun57@kaist.ac.kr

Chieh-Hsin Lai[*]
Sony AI
Tokyo, Japan
chieh-hsin.Lai@sony.com

Wei-Hsiang Liao
Sony AI
Tokyo, Japan

Naoki Murata
Sony AI
Tokyo, Japan

Yuhta Takida
Sony AI
Tokyo, Japan

Yutong He[†]
Sony AI
Tokyo, Japan

Yuki Mitsufuji
Sony AI, Sony Group Corporation
Tokyo, Japan

Stefano Ermon
Stanford University
CA, USA

Abstract

Consistency Models (CM) [1] accelerate score-based diffusion model sampling at the cost of sample quality but lack a natural way to trade-off quality for speed. To address this limitation, we propose **C**onsistency **T**rajectory **M**odel (CTM), a generalization encompassing CM and score-based models as special cases. CTM trains a single neural network that can output scores (i.e., gradients of log-density) and enables unrestricted traversal between any initial and final time along the Probability Flow Ordinary Differential Equation (ODE) in a diffusion process. CTM enables the efficient combination of adversarial training and denoising score matching loss to enhance performance and sets state-of-the-art FIDs for one-step diffusion model generation on CIFAR-10 (FID 1.73) and ImageNet 64×64 (FID 2.06). CTM also enables a new family of sampling schemes, both deterministic and stochastic, involving long jumps along the ODE solution trajectories.

1 Introduction

In diffusion model (DM) [2, 3], the encoder structure is formulated using a set of continuous-time random variables defined by a fixed forward diffusion proces, $\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}_t$, initialized by the data variable, $\mathbf{x}_0 \sim p_{\text{data}}$. A reverse-time process [4] from T to 0 is established $\mathrm{d}\mathbf{x}_t = -2t\nabla \log p_t(\mathbf{x}_t)dt + \sqrt{2t}\,\mathrm{d}\bar{\mathbf{w}}_t$, where $\bar{\mathbf{w}}_t$ is the standard Wiener process in reverse-time, and $p_t(\mathbf{x})$ is the marginal density of $\mathbf{x}_t$ following the forward process. The solution of this reverse-time process aligns with that of the forward-time process marginally (in distribution) when the reverse-time process is initialized with $\mathbf{x}_T \sim p_T$. The deterministic counterpart of the reverse-time process, called the PF ODE [3], is given by $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = -t\nabla \log p_t(\mathbf{x}_t) = \frac{\mathbf{x}_t - \mathbb{E}_{p_{t0}(\mathbf{x}|\mathbf{x}_t)}[\mathbf{x}|\mathbf{x}_t]}{t}$, where $p_{t0}(\mathbf{x}|\mathbf{x}_t)$ is the probability distribution of the solution of the reverse-time stochastic process from time t to zero, initiated from $\mathbf{x}_t$. Here, $\mathbb{E}_{p_{t0}(\mathbf{x}|\mathbf{x}_t)}[\mathbf{x}|\mathbf{x}_t]$ is the denoiser function [5], an alternative expression for the score function $\nabla \log p_t(\mathbf{x}_t)$. For notational simplicity, we omit $p_{t0}(\mathbf{x}|\mathbf{x}_t)$, a subscript in the expectation of the denoiser, throughout the paper.

[*]Equal contribution
[†]Work done during an internship at SONY AI

Neural Information Processing Systems (NeurIPS) 2023 Workshop on Diffusion Models.

In practice, the denoiser $\mathbb{E}[\mathbf{x}|\mathbf{x}_t]$ is approximated using a neural network $D_{\boldsymbol{\phi}}$, obtained by minimizing the Denoising Score Matching (DSM) [6, 3] loss $\mathbb{E}_{\mathbf{x}_0,t,p_{0t}(\mathbf{x}|\mathbf{x}_0)}[\|\mathbf{x}_0 - D_{\boldsymbol{\phi}}(\mathbf{x},t)\|_2^2]$, where $p_{0t}(\mathbf{x}|\mathbf{x}_0)$ is the transition probability from time 0 to t, initiated with $\mathbf{x}_0$. With the approximated denoiser, the empirical PF ODE is given by $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\mathbf{x}_t - D_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}{t}$. Sampling from DM involves solving the PF ODE, equivalent to computing the integral

$$\int_T^0 \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\,\mathrm{d}t = \int_T^0 \frac{\mathbf{x}_t - D_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}{t}\,\mathrm{d}t \iff \mathbf{x}_0 = \mathbf{x}_T + \int_T^0 \frac{\mathbf{x}_t - D_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}{t}\,\mathrm{d}t, \tag{1}$$

where $\mathbf{x}_T$ is sampled from a prior distribution π approximating p_T. Decoding strategies of DM primarily fall into two categories: *score-based sampling* with time-discretized numerical integral solvers, and *distillation sampling* where a neural network directly estimates the integral.

Score-based Sampling Any off-the-shelf ODE solver, denoted as $\texttt{Solver}(\mathbf{x}_T, T, 0; \boldsymbol{\phi})$ (with an initial value of $\mathbf{x}_T$ at time T and ending at time 0), can be directly applied to solve Eq. (1) [3]. For instance, DDIM [7] corresponds to a 1st-order Euler solver, while EDM [8] introduces a 2nd-order Heun solver. Despite recent advancements in numerical solvers [9, 10], further improvements may be challenging due to the inherent discretization error present in all solvers [11], ultimately limiting the sample quality obtained with few NFEs.

Distillation Sampling Distillation models [12, 13] successfully amortize the sampling cost by directly estimating the integral of Eq. (1) with a single neural network evaluation. However, their multistep sampling approach [1] exhibits degrading sample quality with increasing NFE, lacking a clear trade-off between computational budget (NFE) and sample fidelity. Furthermore, multistep sampling is not deterministic, leading to uncontrollable sample variance.

2 CTM: An Unification of Score-based and Distillation Models

To address the challenges in both score-based and distillation samplings, we introduce the **C**onsistency **T**rajectory **M**odel (CTM), which seamlessly integrates both decoding strategies. Consequently, our model is versatile and can perform sampling through either SDE/ODE solving or direct prediction of intermediate points along the PF ODE trajectory.

2.1 Decoder Parametrization of Consistency Trajectory Models

CTM predicts both infinitesimal changes and intermediate points of the PF ODE trajectory. Specifically, we define $G(\mathbf{x}_t, t, s)$ as the solution of the PF ODE from t to s, initialized at $\mathbf{x}_t$:

$$G(\mathbf{x}_t, t, s) := \mathbf{x}_t + \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}|\mathbf{x}_u]}{u}\,\mathrm{d}u. \tag{2}$$

G can access any intermediate point along the trajectory by varying final time s. However, with the current expression of G, the infinitesimal change needed to recover the denoiser information (the integrand) can only be obtained by evaluating the s-derivative at time t, $\frac{\partial}{\partial s}G(\mathbf{x}_t,t,s)|_{s=t}$. Therefore, we introduce a dedicated expression for G using an auxiliary function g to enable easy access to both the integral via G and the integrand via g with Lemma 1.

Lemma 1 (Unification of score-based and distillation models)**.** *Suppose that the score satisfies* $\sup_{\mathbf{x}} \int_0^T \|\nabla \log p_u(\mathbf{x})\|_2\,\mathrm{d}u < \infty$. *The solution, $G(\mathbf{x}_t,t,s)$, defined in Eq. (2) can be expressed as:*

$$G(\mathbf{x}_t,t,s) = \frac{s}{t}\mathbf{x}_t + \left(1-\frac{s}{t}\right)g(\mathbf{x}_t,t,s) \;\; \textit{with} \;\; g(\mathbf{x}_t,t,s) = \mathbf{x}_t + \frac{t}{t-s}\int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}|\mathbf{x}_u]}{u}\,\mathrm{d}u.$$

Here, g satisfies:

(i) When $s=0$, $G(\mathbf{x}_t,t,0) = g(\mathbf{x}_t,t,0)$ is the solution of PF ODE at $s=0$, initialized at $\mathbf{x}_t$.

(ii) As $s \to t$, $g(\mathbf{x}_t,t,s) \to \mathbb{E}[\mathbf{x}|\mathbf{x}_t]$. Hence, g can be defined at $s=t$ by: $g(\mathbf{x}_t,t,t) := \mathbb{E}[\mathbf{x}|\mathbf{x}_t]$.

Indeed, the G's expression in Lemma 1 is naturally linked to the Taylor approximation to the integral:

$$G(\mathbf{x}_t,t,s) = \mathbf{x}_t + \left[(s-t)\frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}|\mathbf{x}_t]}{t} + \mathcal{O}\big(|t-s|^2\big)\right] = \frac{s}{t}\mathbf{x}_t + \left(1-\frac{s}{t}\right)\underbrace{\left[\mathbb{E}[\mathbf{x}|\mathbf{x}_t] + \mathcal{O}\left(|t-s|\right)\right]}_{=g(\mathbf{x}_t,t,s)},$$

for any $s \leq t$. Here, it is evident that g includes all residual terms in Taylor expansion, which turns to be the discretization error in sampling. The goal of CTM is to approximate this g-function using a neural network $g_{\boldsymbol{\theta}}$ and estimate the solution trajectory with the parametrization inspired by Lemma 1:

$$G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) := \frac{s}{t}\mathbf{x}_t + \Big(1 - \frac{s}{t}\Big) g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s).$$

2.2 CTM Training

To achieve trajectory learning, CTM should match the model prediction to the ground truth G by $G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx G(\mathbf{x}_t, t, s)$, for any $s \leq t$. We opt to approximate G by solving the empirical PF ODE with a pre-trained score model $D_{\boldsymbol{\phi}}$. Our neural network is then trained to align with the reconstruction: $G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx \texttt{Solver}(\mathbf{x}_t, t, s; \boldsymbol{\phi})$. However, employing `Solver` throughout the trajectory can significantly increase training time. To efficiently estimate the entire solution trajectory with higher precision, we introduce *soft matching*, ensuring consistency between prediction from $\mathbf{x}_t$ and from $\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi})$ for any $u \in [s, t)$: $G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx G_{\texttt{sg}(\boldsymbol{\theta})}\big(\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi}), u, s\big)$. where $\texttt{sg}(\cdot)$ is stop-gradient. This soft matching spans two frameworks. As $u = s$, Eq. (**??**) enforces *global consistency matching*, i.e., a reconstruction loss. In contrast, as $u = t - \Delta t$, Eq. (**??**) is *local consistency matching*. Additionally, if $s = 0$, it recovers CM's distillation loss.

To quantify the dissimilarity between $G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s)$ and $G_{\texttt{sg}(\boldsymbol{\theta})}(\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi}), u, s)$ and enforce Eq. (**??**), we use the feature distance LPIPS d [14] by comparing

$$\mathbf{x}_{\text{est}}(\mathbf{x}_t, t, s) := G_{\texttt{sg}(\boldsymbol{\theta})}\Big(G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s), s, 0\Big)$$
$$\mathbf{x}_{\text{target}}(\mathbf{x}_t, t, u, s) := G_{\texttt{sg}(\boldsymbol{\theta})}\Big(G_{\texttt{sg}(\boldsymbol{\theta})}\big(\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi}), u, s\big), s, 0\Big).$$

Overall, the CTM loss is defined as

$$\mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}_{t\in[0,T]}\mathbb{E}_{s\in[0,t]}\mathbb{E}_{u\in[s,t)}\mathbb{E}_{\mathbf{x}_0, p_{0t}(\mathbf{x}|\mathbf{x}_0)}\Big[d\big(\mathbf{x}_{\text{target}}(\mathbf{x}, t, u, s), \mathbf{x}_{\text{est}}(\mathbf{x}, t, s)\big)\Big], \quad (3)$$

which leads the model's prediction, at optimum, to match with the empirical PF ODE's solution trajectory, defined by the pre-trained DM (teacher), see Appendix B (Propositions 2 and 4) for details.

2.3 Training Consistency Trajectory Models

Training CTM with Eq. (3) may empirically lead inaccurate estimation of $g_{\boldsymbol{\theta}}$ when s approaches t. This is due to the learning signal of $g_{\boldsymbol{\theta}}$ being scaled with $1 - \frac{s}{t}$ by Lemma 1, and this scale decreasing to zero as s approaches t. Consequently, although our parametrization enables the estimation of both the trajectory and its slope, the accuracy of slope (score) estimation may be degraded. To mitigate this problem, we use Lemma 1's conclusion that $g(\mathbf{x}_t, t, t) = \mathbb{E}[\mathbf{x}|\mathbf{x}_t]$ when $t = s$ and train $g_{\boldsymbol{\theta}}$ with:

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_t|\mathbf{x}_0}[\|\mathbf{x}_0 - g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t)\|_2^2].$$

Empirically, regularizing $\mathcal{L}_{\text{CTM}}$ with $\mathcal{L}_{\text{DSM}}$ improves score accuracy, which is especially important in large NFE sampling regimes.

On the other hand, CTM, distilling from the teacher model, is constrained by the teacher's $D_{\boldsymbol{\phi}}$ performance. This challenge can be mitigated with adversarial training to improve trajectory estimation. The one-step generation of CTM enables us to calculate the adversarial loss efficiently, in the similar way of conventional GAN training:

$$\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x}_0)}[\log d_{\boldsymbol{\eta}}(\mathbf{x}_0)] + \mathbb{E}_{t, \mathbf{x}_t}\big[\log\big(1 - d_{\boldsymbol{\eta}}(\mathbf{x}_{\text{est}})\big)\big],$$

where $d_{\boldsymbol{\eta}}$ is a discriminator. This adversarial training allows *the student model (CTM) to beat the teacher model (DM)*. To summarize, CTM allows the integration of reconstruction-based CTM loss, diffusion loss, and adversarial loss with weighting functions$\lambda_{\text{DSM}}, \lambda_{\text{GAN}} \geq 0$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) := \mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}; \phi) + \lambda_{\text{DSM}}\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta}), \quad (4)$$

in a single training framework, by optimizing $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta})$.

Table 1: Performance comparisons on CIFAR-10.

Model	NFE	Unconditional		Conditional
		FID↓	NLL↓	FID↓
GAN Models				
BigGAN [15]	1	8.51	✗	-
StyleGAN-Ada [16]	1	2.92	✗	2.42
StyleGAN-D2D [17]	1	-	✗	2.26
StyleGAN-XL [18]	1	-	✗	1.85
Diffusion Models – Score-based Sampling				
DDPM [19]	1000	3.17	3.75	-
DDIM [7]	100	4.16	-	-
	10	13.36	-	-
Score SDE [7]	2000	2.20	3.45	-
VDM [20]	1000	7.41	2.49	-
LSGM [21]	138	2.10	3.43	-
EDM [8]	35	2.01	2.56	1.82
Diffusion Models – Distillation Sampling				
KD [22]	1	9.36	✗	-
DFNO [23]	1	5.92	✗	-
Rectified Flow [24]	1	4.85	✗	-
PD [12]	1	9.12	✗	-
CD (official report) [1]	1	3.55	✗	-
CD (retrained)	1	10.53	✗	-
CD + GAN [25]	1	2.65	✗	-
CTM (ours)	1	1.98	**2.43**	1.73
PD [12]	2	4.51	-	-
CD [1]	2	2.93	-	-
CTM (ours)	2	**1.87**	**2.43**	**1.63**

Table 2: Performance comparisons on ImageNet 64×64.

Model	NFE	FID↓	IS↑
ADM [26]	250	2.07	-
EDM [8]	79	2.44	48.88
BigGAN-deep [15]	1	4.06	-
StyleGAN-XL [18]	1	2.09	**82.35**
Diffusion Models – Distillation Sampling			
PD [12]	1	15.39	-
BOOT [27]	1	16.3	-
CD [1]	1	6.20	40.08
CTM (ours)	1	2.06	70.86
PD [12]	2	8.95	-
CD [1]	2	4.70	-
CTM (ours)	2	**1.90**	64.14

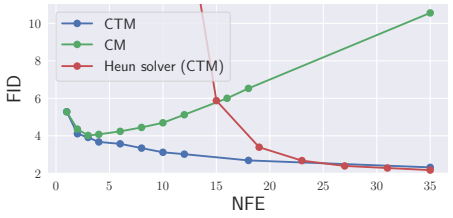


Figure 1: Comparison of samplers.

3 Sampling with CTM

Score-based Sampling CTM enables score evaluation via $g_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, t)$, supporting standard score-based sampling with ODE/SDE solvers. The result is shown as the red curve in Figure 1.

Distillation Sampling CTM additionally enables time traversal along the solution trajectory, allowing a new sampling method avoid ad-hoc noise injection as in CM. Suppose the sampling timesteps are $T = t_0 > \cdots > t_N = 0$. With $\mathbf{x}_{t_0} \sim \pi$, where π is the prior distribution, CTM denoises $\mathbf{x}_{t_0}$ to time t_1 with $G_{\boldsymbol{\theta}}(\mathbf{x}_{t_0}, t_0, t_1)$, and repeating this denoising process until reaching to time $t_N = 0$. A key distinction between the CTM's sampling and score-based sampling is that CTM avoids sampling errors by directly estimating Eq. (2). However, score-based samplers like DDIM or EDM are susceptible to discretization errors because they only estimates the denoiser term from the Taylor expansion, ignoring the residual term, which dominates the integral scale for small NFE.

Figure 1 shows that CTM's deterministic sampling (blue) reaches comparable performance as the Heun's solver (red) as NFE increases. In contrast, CM's multistep sampler (green) significantly degrades sample quality as NFE increases. This quality deterioration may be attributed to error accumulation during the iterative long "jumps" for denoising. CM's multistep sampling iteratively conducts long jumps from t_n to 0 for each step n, which aggregates the sampling error to be $\mathcal{O}(\sqrt{T + t_1 + \cdots + t_N})$ (Theorem 5). In contrast, such time overlap does not occur in CTM, eliminating the error accumulation, resulting in $\mathcal{O}(\sqrt{T})$ error.

4 Experiments – Student (CTM) beats teacher (DM)

We evaluate CTM on CIFAR-10 and ImageNet 64×64, using the pre-trained diffusion checkpoints from EDM as the teacher models. We adopt EDM's training configuration for $\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta})$ and employ StyleGAN-XL's [18] discriminator for $\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta})$ (Appendix B.3). In addition to the clear NFE-FID trade-off in Figure 1, with GAN loss, CTM achieves new SOTA FIDS with 1 NFE, beating both EDM and StyleGAN-XL. Additionally, CTM's ability to approximate scores using $g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t)$ enables evaluating Negative Log-Likelihood (NLL) [28, 29], also establishing a new SOTA NLL.

5 Conclusion

CTM allows unrestricted time traversal and seamless integration with prior models' training advantages. A universal framework for Consistency and Diffusion Models, CTM excels in both training and sampling. Remarkably, it surpasses its teacher model, achieving SOTA results in FID and likelihood for few-steps diffusion model sampling on CIFAR-10 and ImageNet 64×64, highlighting its versatility and process.

References

[1] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[2] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

[4] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[5] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

[6] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[7] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[8] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

[9] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[10] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2022.

[11] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[12] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

[13] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[14] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[15] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[16] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[17] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting acgan: Auxiliary classifier gans with stable training. *Advances in neural information processing systems*, 34: 23505–23518, 2021.

[18] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[20] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[21] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[22] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

[23] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pages 42390–42402. PMLR, 2023.

[24] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022.

[25] Haoye Lu, Yiwei Lu, Dihong Jiang, Spencer Ryan Szabados, Sun Sun, and Yaoliang Yu. Cm-gan: Stabilizing gan training with consistency models. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

[26] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[27] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, 2023.

[28] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34: 1415–1428, 2021.

[29] Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-chul Moon. Maximum likelihood training of implicit nonlinear diffusion model. *Advances in Neural Information Processing Systems*, 35:32270–32284, 2022.

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[32] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[33] Kin Wai Cheuk, Ryosuke Sawata, Toshimitsu Uesaka, Naoki Murata, Naoya Takahashi, Shusuke Takahashi, Dorien Herremans, and Yuki Mitsufuji. Diffroll: Diffusion-based generative music transcription with unsupervised pretraining capability. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[34] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

[35] Koichi Saito, Naoki Murata, Toshimitsu Uesaka, Chieh-Hsin Lai, Yuhta Takida, Takao Fukui, and Yuki Mitsufuji. Unsupervised vocal dereverberation with diffusion-based generative models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[36] Carlos Hernandez-Olivan, Koichi Saito, Naoki Murata, Chieh-Hsin Lai, Marco A Martínez-Ramirez, Wei-Hsiang Liao, and Yuki Mitsufuji. Vrdmg: Vocal restoration via diffusion posterior sampling with multiple guidance. *arXiv preprint arXiv:2309.06934*, 2023.

[37] Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Gibbsddrm: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. *arXiv preprint arXiv:2301.12686*, 2023.

[38] Dongjun Kim, Yeongmin Kim, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022.

[39] Cheng Lu, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pages 14429–14460. PMLR, 2022.

[40] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbot, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

[41] Shitong Shao, Xu Dai, Shouyi Yin, Lujun Li, Huanran Chen, and Yang Hu. Catch-up distillation: You only need to train once for accelerating sampling. *arXiv preprint arXiv:2305.10769*, 2023.

[42] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *International Conference on Machine Learning*, pages 11201–11228. PMLR, 2022.

[43] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *arXiv preprint arXiv:2302.09057*, 2023.

[44] Yangming Li, Zhaozhi Qian, and Mihaela van der Schaar. Do diffusion models suffer error propagation? theoretical analysis and consistency regularization. *arXiv preprint arXiv:2308.05021*, 2023.

[45] Chieh-Hsin Lai, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. 2023.

[46] Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Naoki Murata, Yuki Mitsufuji, and Stefano Ermon. On the equivalence of consistency-type models: Consistency models, consistent diffusion models, and fokker-planck regularization. *arXiv preprint arXiv:2306.00367*, 2023.

[47] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[49] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[50] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2022.

[51] Dennis D Boos. A converse to scheffe's theorem. *The Annals of Statistics*, pages 423–427, 1985.

[52] TJ Sweeting. On a converse to scheffé's theorem. *The Annals of Statistics*, 14(3):1252–1256, 1986.

[53] W.T. Reid. *Ordinary Differential Equations*. Applied mathematics series. Wiley, 1971.

[54] Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations*, pages 65–84. Springer, 2003.

[55] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Contents

A Related Works

Diffusion Models DMs excel in high-fidelity synthetic image and audio generation [26, 30, 31], as well as in applications like media editing, restoration [32–37]. Recent research aims to enhance DMs in sample quality [29, 38], density estimation [28, 39], and especially, sampling speed [7].

Fast Sampling of DMs The SDE framework underlying DMs [3] has driven research into various numerical methods for accelerating DM sampling, exemplified by works such as [7, 10, 9]. Notably, [9] reduced the ODE solver steps to as few as 10-15. Other approaches involve learning the solution operator of ODEs [23], discovering optimal transport paths for sampling [24], or employing distillation techniques [22, 12, 40, 41]. However, previous distillation models may experience slow convergence or extended runtime. Gu et al. [27] introduced a bootstrapping approach for data-free distillation. Furthermore, Song et al. [1] introduced CM which extracts DMs' PF ODE to establish a direct mapping from noise to clean predictions, achieving one-step sampling while maintaining good sample quality. CM has been adapted to enhance the training stability of GANs, as [25]. However, it's important to note that their focus does not revolve around achieving sampling acceleration for DMs, nor are the results restricted to simple datasets.

Consistency of DMs Score-based generative models rely on a differential equation framework, employing neural networks trained on data to model the conversion between data and noise. These networks must satisfy specific consistency requirements due to the mathematical nature of the underlying equation. Early investigations, such as [42], identified discrepancies between learned scores and ground truth scores. Recent developments have introduced various consistency concepts, showing their ability to enhance sample quality [43, 44], accelerate sampling speed [1], and improve density estimation in diffusion modeling [45]. Notably, Lai et al. [46] established the theoretical equivalence of these consistency concepts, suggesting the potential for a unified framework that can empirically leverage their advantages. CTM can be viewed as the first framework which achieves all the desired properties.

B Theoretical Insights on CTM

In this section, we explore several theoretical aspects of CTM, encompassing convergence analysis (Section B.1), properties of well-trained CTM, and accumulated errors analysis during sampling.

We first introduce and review some notions. Starting at time t with an initial value of $\mathbf{x}_t$ and ending at time s, recall that $G(\mathbf{x}_t, t, s)$ represents the true solution of the PF ODE, and $G(\mathbf{x}_t, t, s; \boldsymbol{\phi})$ is the solution function of the following empirical PF ODE.

$$\frac{\mathrm{d}\mathbf{x}_u}{\mathrm{d}u} = \frac{\mathbf{x}_u - D_{\boldsymbol{\phi}}(\mathbf{x}_u, u)}{u}, \quad u \in [0, T]. \tag{5}$$

Here $\boldsymbol{\phi}$ denotes the teacher model's weights learned from DSM. Thus, $G(\mathbf{x}_t, t, s; \boldsymbol{\phi})$ can be expressed as

$$G(\mathbf{x}_t, t, s; \boldsymbol{\phi}) = \frac{s}{t}\mathbf{x}_t + (1 - \frac{s}{t})g(\mathbf{x}_t, t, s; \boldsymbol{\phi}),$$

where $g(\mathbf{x}_t, t, s; \boldsymbol{\phi}) = \mathbf{x}_t + \frac{t}{t-s}\int_t^s \frac{\mathbf{x}_u - D_{\boldsymbol{\phi}}(\mathbf{x}_u, u)}{u}\,\mathrm{d}u$.

B.1 Convergence Analysis – Distillation from Teacher Models

Convergence along Trajectory in a Time Discretization Setup. CTM's practical implementation follows CM's one, utilizing discrete timesteps $t_0 = 0 < t_1 < \cdots < t_N = T$ for training. Initially, we assume local consistency matching for simplicity, but this can be extended to soft matching. This transforms the CTM loss in Eq. (3) to the discrete time counterpart:

$$\mathcal{L}^N_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}_{n\in[\![1,N]\!]}\mathbb{E}_{m\in[\![0,n]\!]}\mathbb{E}_{\mathbf{x}_0, p_{0t_n}(\mathbf{x}|\mathbf{x}_0)}\Big[d\big(\mathbf{x}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_m), \mathbf{x}_{\text{est}}(\mathbf{x}_{t_n}, t_n, t_m)\big)\Big],$$

where $d(\cdot,\cdot)$ is a metric, and

$$\mathbf{x}_{\text{est}}(\mathbf{x}_{t_n}, t_n, t_m) := G_{\boldsymbol{\theta}}\Big(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0\Big)$$

$$\mathbf{x}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_{n-1}, t_m) := G_{\boldsymbol{\theta}}\Big(G_{\boldsymbol{\theta}}\big(\texttt{Solver}(\mathbf{x}_{t_n}, t_n, t_{n-1}; \boldsymbol{\phi}), t_{n-1}, t_m\big), t_m, 0\Big).$$

In the following theorem, we demonstrate that irrespective of the initial time t_n and end time t_m, CTM $G_{\boldsymbol{\theta}}(\cdot, t_n, t_m; \boldsymbol{\phi})$, will eventually converge to its teacher model, $G(\cdot, t_n, t_m; \boldsymbol{\phi})$.

Proposition 2. *Define $\Delta_N t := \max_{n \in [\![1,N]\!]} \{|t_{n+1} - t_n|\}$. Assume that $G_{\boldsymbol{\theta}}$ is uniform Lipschitz in $\boldsymbol{x}$ and that the ODE solver admits local truncation error bounded uniformly by $\mathcal{O}((\Delta_N t)^{p+1})$ with $p \geq 1$. If there is a $\boldsymbol{\theta}_N$ so that $\mathcal{L}^N_{CTM}(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$, then for any $n \in [\![1, N]\!]$ and $m \in [\![1, n]\!]$*

$$\sup_{\mathbf{x} \in \mathbb{R}^D} d\big(G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t_n, t_m), t_m, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t_n, t_m; \boldsymbol{\phi}), t_m, 0)\big) = \mathcal{O}((\Delta_N t)^p)(t_n - t_m).$$

Similar argument applies, confirming convergence along the PF ODE trajectory, ensuring Eq. (**??**) with $\boldsymbol{\theta}$ replacing $\texttt{sg}(\boldsymbol{\theta})$:

$$G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx G_{\boldsymbol{\theta}}(\texttt{Solver}(\mathbf{x}_t, t, t - \Delta t; \boldsymbol{\phi}), t - \Delta t, s)$$

by enforcing the following loss

$$\tilde{\mathcal{L}}^N_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}_{n \in [\![1,N]\!]} \mathbb{E}_{m \in [\![0,n]\!]} \mathbb{E}_{\mathbf{x}_0, p_{0t_n}(\mathbf{x}|\mathbf{x}_0)} \Big[d\big(\tilde{\mathbf{x}}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_m), \tilde{\mathbf{x}}_{\text{est}}(\mathbf{x}_{t_n}, t_n, t_m)\big)\Big],$$

where

$$\begin{aligned}
\tilde{\mathbf{x}}_{\text{est}}(\mathbf{x}_{t_n}, t_n, t_m) &:= G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m) \\
\tilde{\mathbf{x}}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_{n-1}, t_m) &:= G_{\boldsymbol{\theta}}\big(\texttt{Solver}(\mathbf{x}_{t_n}, t_n, t_{n-1}; \boldsymbol{\phi}), t_{n-1}, t_m\big).
\end{aligned}$$

Proposition 3. *If there is a $\boldsymbol{\theta}_N$ so that $\tilde{\mathcal{L}}^N_{CTM}(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$, then for any $n \in [\![1, N]\!]$ and $m \in [\![1, n]\!]$*

$$\sup_{\mathbf{x} \in \mathbb{R}^D} d\big(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t_n, t_m), G(\mathbf{x}, t_n, t_m; \boldsymbol{\phi})\big) = \mathcal{O}((\Delta_N t)^p)(t_n - t_m).$$

Convergence of Densities. In Proposition 2, we demonstrated point-wise trajectory convergence, from which we infer that CTM may converge to its training target in terms of density. More precisely, in Proposition 4, we establish that if CTM's target $\mathbf{x}_{\text{target}}$ is derived from the teacher model (as defined above), then the data density induced by CTM will converge to that of the teacher model. Specifically, if the target $\mathbf{x}_{\text{target}}$ perfectly approximates the true G-function:

$$\mathbf{x}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_{n-1}, t_m) \equiv G(\mathbf{x}_{t_n}, t_n, t_m), \quad \text{for all } n \in [\![1, N]\!], m \in [\![0, n]\!], N \in \mathbb{N}. \tag{6}$$

Then the data density generated by CTM will ultimately learn the data distribution p_{data}.

Simplifying, we use the ℓ_2 for the distance metric d and consider the prior distribution π to be p_T, which is the marginal distribution at time $t = T$ defined by the diffusion process in Eq. (**??**).

Proposition 4. *Suppose that*

(i) The uniform Lipschitzness of $G_{\boldsymbol{\theta}}$ (and G),

$$\sup_{\boldsymbol{\theta}} \|G_{\boldsymbol{\theta}}(\mathbf{x}, t, s) - G_{\boldsymbol{\theta}}(\mathbf{x}', t, s)\|_2 \leq L \|\mathbf{x} - \mathbf{x}'\|_2, \quad \textit{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^D, t, s \in [0, T],$$

(ii) The uniform boundedness in $\boldsymbol{\theta}$ of $G_{\boldsymbol{\theta}}$: there is a $L(\mathbf{x}) \geq 0$ so that

$$\sup_{\boldsymbol{\theta}} \|G_{\boldsymbol{\theta}}(\mathbf{x}, t, s)\|_2 \leq L(\mathbf{x}) < \infty, \quad \textit{for all } \mathbf{x} \in \mathbb{R}^D, t, s \in [0, T]$$

If for any N, there is a $\boldsymbol{\theta}_N$ such that $\mathcal{L}^N_{CTM}(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$. Let $p_{\boldsymbol{\theta}_N}(\cdot)$ denote the pushforward distribution of p_T induced by $G_{\boldsymbol{\theta}_N}(\cdot, T, 0)$. Then, as $N \to \infty$, $\|p_{\boldsymbol{\theta}_N}(\cdot) - p_{\boldsymbol{\phi}}(\cdot)\|_\infty \to 0$. Particularly, if the condition in Eq. (6) is satisfied, then $\|p_{\boldsymbol{\theta}_N}(\cdot) - p_{data}(\cdot)\|_\infty \to 0$ as $N \to \infty$.

B.2 Accumulated Errors analysis in sampling.

We begin by clarifying the concept of "density transition by a function". For a measurable mapping $\mathcal{T}: \Omega \to \Omega$ and a measure ν on the measurable space Ω, the notation $\mathcal{T}\sharp\nu$ denotes the pushforward measure, indicating that if a random vector X follows the distribution ν, then $\mathcal{T}(X)$ follows the distribution $\mathcal{T}\sharp\nu$.

Given a sampling timestep $T = t_0 > t_1 > \cdots > t_N = 0$. Let $p_{\boldsymbol{\theta}^*,N}$ represent the density resulting from N-steps of γ-sampling initiated at p_T. That is,

$$p_{\boldsymbol{\theta}^*,N} := \bigcirc_{n=0}^{N-1} \left(\mathcal{T}^{\boldsymbol{\theta}^*}_{\sqrt{1-\gamma^2}t_{n+1} \to t_{n+1}} \circ \mathcal{T}^{\boldsymbol{\theta}^*}_{t_n \to \sqrt{1-\gamma^2}t_{n+1}} \right) \sharp p_T.$$

Here, $\bigcirc_{n=0}^{N-1}$ denotes the sequential composition. We assume an optimal CTM which precisely distills information from the teacher model $G_{\boldsymbol{\theta}^*}(\cdot) = G(\cdot, t, s; \boldsymbol{\phi})$ for all $t, s \in [0, T]$.

Theorem 5 (Accumulated errors of N-steps γ-sampling)**.** *Let $\gamma \in [0, 1]$.*

$$D_{TV}\left(p_{data}, p_{\boldsymbol{\theta}^*,N}\right) = \mathcal{O}\left(\sum_{n=0}^{N-1} \sqrt{t_n - \sqrt{1-\gamma^2}t_{n+1}}\right).$$

*Here, $\mathcal{T}_{t\to s}: \mathbb{R}^D \to \mathbb{R}^D$ denotes the oracle transition mapping from t to s, determined by Eq. (**??**). The pushforward density via $\mathcal{T}_{t\to s}$ is denoted as $\mathcal{T}_{t\to s}\sharp p_t$, with similar notation applied to $\mathcal{T}^{\boldsymbol{\theta}^*}_{t\to s}\sharp p_t$, where $\mathcal{T}^{\boldsymbol{\theta}^*}_{t\to s}$ denotes the transition mapping associated with the optimal CTM trained from Eq. (4).*

B.3 Training Details

Following Karras et al. [8], we utilize the EDM's skip scale and output scale for $g_{\boldsymbol{\theta}}$ modeling as

$$g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) = \frac{\sigma_{\text{data}}^2}{t^2 + \sigma_{\text{data}}^2}\mathbf{x}_t + \frac{t\sigma_{\text{data}}}{\sqrt{t^2 + \sigma_{\text{data}}^2}}\text{NN}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s),$$

where $\text{NN}_{\boldsymbol{\theta}}$ refers to a neural network that takes the same input arguments as $g_{\boldsymbol{\theta}}$. The advantage of this EDM-style skip and output scaling is that if we copy the teacher model's parameters to the student model's parameters, except student model's s-embedding structure, $g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t)$ initialized with $\boldsymbol{\phi}$ would be close to the teacher denoiser $D_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$. This good initialization partially explains the fast convergence speed.

We use 4×V100 (16G) GPUs for CIFAR-10 experiments and 8×A100 (40G) GPUs for ImageNet experiments. We use the warm-up for λ_{GAN} hyperparameter. On CIFAR-10, we deactivate GAN training with $\lambda_{\text{GAN}} = 0$ until 50k training iterations and activate the generator training with the adversarial loss (added to CTM and DSM losses) by increasing λ_{GAN} to one. The minibatch per GPU is 16 in the CTM+DSM training phase, and 11 in the CTM+DSM+GAN training phase. On ImageNet, due to the excessive training budget, we deactivate GAN only for 10k iterations and activate GAN training afterwards. We fix the minibatch to be 11 throughout the CTM+DSM or the CTM+DSM+GAN training in ImageNet.

We follow the training configuration mainly from CM, but for the discriminator training, we follow that of StyleGAN-XL [18]. For $\mathcal{L}_{\text{CTM}}$ calculation, we use LPIPS [14] as a feature extractor. We choose t and s from the N-discretized timesteps to calculate $\mathcal{L}_{\text{CTM}}$, following CM. Across the training, we choose the maximum number of ODE steps to prevent a single iteration takes too long time. For CIFAR-10, we choose $N = 18$ and the maximum number of ODE steps to be 17. For ImageNet, we choose $N = 40$ and the maximum number of ODE steps to be 20. We find the tendency that the training performance is improved by the number of ODE steps, so one could possibly improve our ImageNet result by choosing larger maximum ODE steps.

For $\mathcal{L}_{\text{DSM}}$ calculation, we select 50% of time sampling from EDM's original scheme of $t \sim \mathcal{N}(-1.2, 1.2^2)$. For the other half time, we first draw sample from $\xi \sim [0, 0.7]$ and transform it using $(\sigma_{\text{max}}^{1/\rho} + \xi(\sigma_{\text{min}}^{1/\rho} - \sigma_{\text{max}}^{1/\rho}))^\rho$. This specific time sampling blocks the neural network to forget the denoiser information for large time. For $\mathcal{L}_{\text{GAN}}$ calculation, we use two feature extractors to

transform GAN input to the feature space: the EfficientNet [47] and DeiT-base [48]. Before obtaining an input's feature, we upscale the image to 224x224 resolution with bilinear interpolation. After transforming to the feature space, we apply the cross-channel mixing and cross-scale mixing to represent the input with abundant and non-overlapping features. The output of the cross-scale mixing is a feature pyramid consisting of four feature maps at different resolutions [18]. In total, we use eight discriminators (four for EfficientNet features and the other four for DeiT-base features) for GAN training.

Following CM, we apply Exponential Moving Average (EMA) to update $\texttt{sg}(\boldsymbol{\theta})$ by

$$\texttt{sg}(\boldsymbol{\theta}) \leftarrow \texttt{stopgrad}(\mu\texttt{sg}(\boldsymbol{\theta}) + (1-\mu)\boldsymbol{\theta}).$$

However, unlike CM, we find that our model bestly works with $\mu = 0.999$ or $\mu = 0.9999$, which largely remedy the subtle instability arise from GAN training. Except for the unconditional CIFAR-10 training with $\boldsymbol{\phi}$, we set μ to be 0.999 as default. Throughout the experiments, we use $\sigma_{\text{min}} = 0.002$, $\sigma_{\text{max}} = 80$, $\rho = 7$, and $\sigma_{\text{data}} = 0.5$.

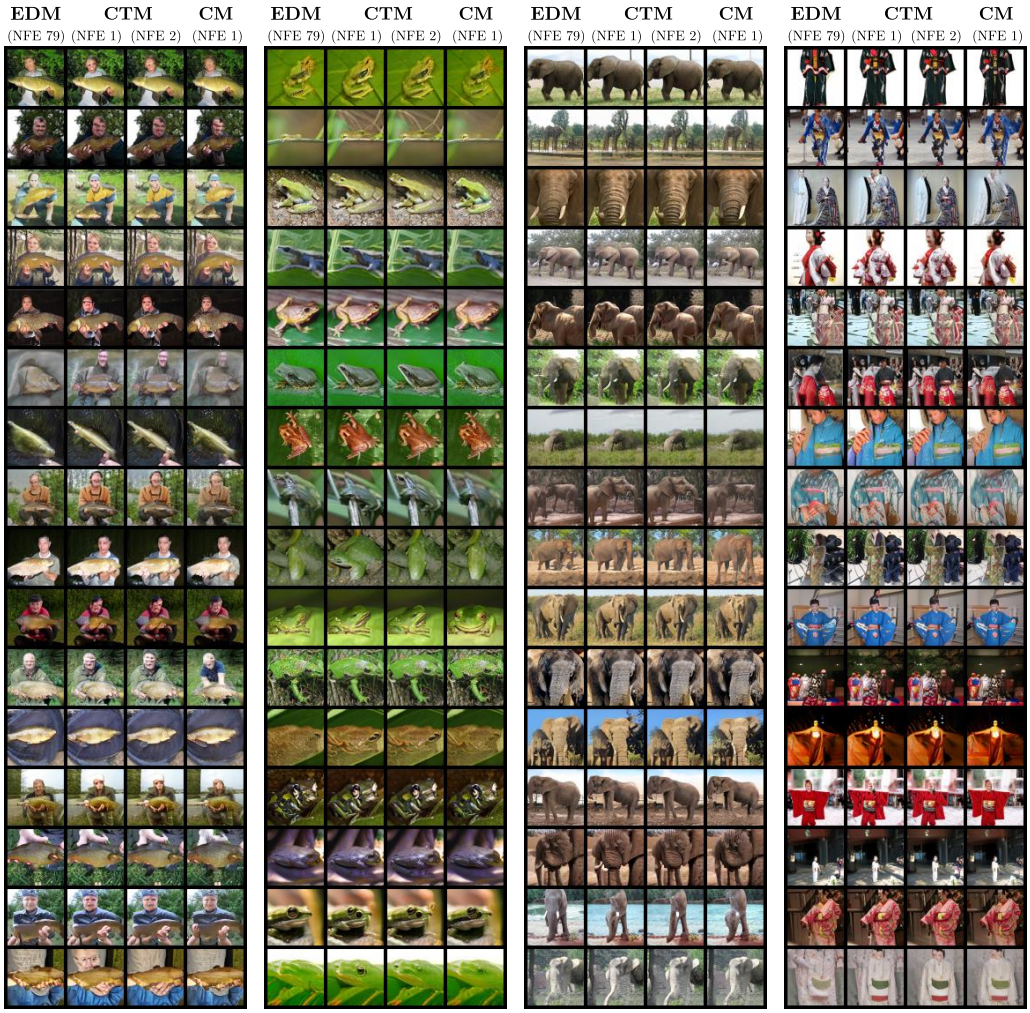
B.4 Evaluation Details

For likelihood evaluation, we solve the PF ODE, following the practice suggested in Kim et al. [29] with the RK45 [49] ODE solver of $\texttt{tol} = 1e-3$ and $t_{\text{min}} = 0.002$.

Throughout the paper, we choose $\gamma = 0$ otherwise stated. In particular, for Tables 1 and 2, we report the sample quality metrics based on either the one-step sampling of CM or the $\gamma = 0$ sampling for NFE 2 case. For CIFAR-10, we calculate the FID score based on Karras et al. [8] statistics. For ImageNet, we compute the metrics following Dhariwal and Nichol [26] and their pre-calculated statistics. For the StyleGAN-XL ImageNet result, we recalculated the metrics based on the statistics released by Dhariwal and Nichol [26], using StyleGAN-XL's official checkpoint.

For large-NFE sampling, we follow the EDM's time discretization. Namely, if we draw n-NFE samples, we equi-divide $[0, 1]$ with n points and transform it (say ξ) to the time scale by $(\sigma_{\text{max}}^{1/\rho} + (\sigma_{\text{min}}^{1/\rho} - \sigma_{\text{max}}^{1/\rho})\xi)^{\rho}$. However, we emphasize the time discretization for both training and sampling is a modeler's choice.

C Additional Generated Samples



(a) Tench (b) Tree frog (c) Elephant (d) Kimono

Figure 2: Uncurated sample comparisons with identical starting points, generated by EDM (FID 2.44) with NFE 79, CTM (FID 2.19) with NFE 1, CTM (FID 1.90) with NFE 2, and CM (FID 6.20) with NFE 1, on (a) tench (class id: 0), (b) tree frog (class id: 31), (c) elephant (class id: 386), and (d) kimono (class id: 614).

D Theoretical Supports and Proofs

D.1 Proof of Lemma 1

Proof of Lemma 1. As the score, $\nabla \log p_t(\mathbf{x})$, is integrable, the Fundamental Theorem of Calculus applies, leading to

$$\begin{aligned}\lim_{s\to t} g(\mathbf{x}_t, t, s) &= \mathbf{x}_t + t \lim_{s\to t} \frac{1}{t-s} \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_u]}{u}\, \mathrm{d}u \\ &= \mathbf{x}_t - t\frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]}{t} \\ &= \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t].\end{aligned}$$

■

D.2 Proof of Theorem ??

Proof of Theorem ??. Define $\mathcal{T}_{t\to s}$ as the oracle transition mapping from t to s via the diffusion process Eq. (**??**). Let $\mathcal{T}^{\boldsymbol{\theta}^*}_{t\to s}(\cdot)$ represent the transition mapping from the optimal CTM, and $\mathcal{T}^{\boldsymbol{\phi}}_{t\to s}(\cdot)$ represent the transition mapping from the empirical probability flow ODE. Since all processes start at point T with initial probability distribution p_T and $\mathcal{T}^{\boldsymbol{\theta}^*}_{t\to s}(\cdot) = \mathcal{T}^{\boldsymbol{\phi}}_{t\to s}(\cdot)$, Theorem 2 in [50] and $\mathcal{T}_{T\to t}\sharp p_T = p_t$ from Proposition **??** tell us that for $t > s$

$$D_{TV}\left(\mathcal{T}_{t\to s}\sharp p_t, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to s}\sharp p_t\right) = D_{TV}\left(\mathcal{T}_{t\to s}\sharp p_t, \mathcal{T}^{\boldsymbol{\phi}}_{t\to s}\sharp p_t\right) = \mathcal{O}(t-s). \tag{7}$$

$$\begin{aligned}&D_{TV}\left(\mathcal{T}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}^{\boldsymbol{\theta}^*}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T\right) \\ \overset{(a)}{\leq}& D_{TV}\left(\mathcal{T}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T\right) \\ &+D_{TV}\left(\mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}^{\boldsymbol{\theta}^*}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T\right) \\ \overset{(b)}{=}& D_{TV}\left(\mathcal{T}_{t\to 0}\mathcal{T}_{T\to t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\mathcal{T}_{T\to t}\sharp p_T\right) + D_{TV}\left(\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T\right) \\ \overset{(c)}{=}& D_{TV}\left(\mathcal{T}_{t\to 0}\sharp p_t, \mathcal{T}^{\boldsymbol{\theta}^*}_{t\to 0}\sharp p_t\right) + D_{TV}\left(\mathcal{T}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T, \mathcal{T}^{\boldsymbol{\theta}^*}_{T\to\sqrt{1-\gamma^2}t}\sharp p_T\right) \\ \overset{(d)}{=}& \mathcal{O}(\sqrt{t}) + \mathcal{O}(\sqrt{T - \sqrt{1-\gamma^2}t}).\end{aligned}$$

Here (a) is obtained from the triangular inequality, (b) and (c) are due to $\mathcal{T}_{\sqrt{1-\gamma^2}t\to t}\mathcal{T}_{T\to\sqrt{1-\gamma^2}t} = \mathcal{T}_{T\to t}$ and $\mathcal{T}_{T\to t}\sharp p_T = p_t$ from Proposition **??**, and (d) comes from Eq. (7).

■

D.3 Proof of Proposition 2

Proof of Proposition 2. Consider a LPIPS-like metric, denoted as $d(\cdot,\cdot)$, determined by a feature extractor $\mathcal{F}$ of p_{data}. That is, $d(\mathbf{x},\mathbf{y}) = \|\mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y})\|_q$ for $q \geq 1$. For simplicity of notation, we denote $\boldsymbol{\theta}_N$ as $\boldsymbol{\theta}$. Since $\mathcal{L}^N_{\text{CTM}}(\boldsymbol{\theta};\boldsymbol{\phi}) = 0$, it implies that for any $\mathbf{x}_{t_n}$, $n \in [\![1, N]\!]$, and $m \in [\![1, n]\!]$

$$\mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_m), t_m, 0)\big) = \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}^{\boldsymbol{\phi}}_{t_n}, t_n, t_m), t_m, 0)\big) \tag{8}$$

Denote

$$\mathbf{e}_{n,m} := \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G(\mathbf{x}_{t_n}, t_n, t_m;\boldsymbol{\phi}), t_m, 0)\big).$$

Then due to Eq. (8) and G is an ODE-trajectory function that $G(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_m; \boldsymbol{\phi}) = G(\mathbf{x}_{t_n}, t_n, t_m; \boldsymbol{\phi})$, we have

$$\begin{aligned}
\mathbf{e}_{n+1,m} &= \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_m; \boldsymbol{\phi}), t_m, 0)\big) \\
&= \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}^{\boldsymbol{\phi}}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G(\mathbf{x}_{t_n}, t_n, t_m; \boldsymbol{\phi}), t_m, 0)\big) \\
&= \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}^{\boldsymbol{\phi}}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0)\big) \\
&+ \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G(\mathbf{x}_{t_n}, t_n, t_m; \boldsymbol{\phi}), t_m, 0)\big) \\
&= \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}^{\boldsymbol{\phi}}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0)\big) + \mathbf{e}_{n,m}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\mathbf{e}_{n+1,m}\|_q &\le \left\|\mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}^{\boldsymbol{\phi}}_{t_n}, t_n, t_m), t_m, 0)\big) - \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, t_m), t_m, 0)\big)\right\|_q + \|\mathbf{e}_{n,m}\|_q \\
&\le L_1 L_2^2 \left\|\mathbf{x}^{\boldsymbol{\phi}}_{t_n} - \mathbf{x}_{t_n}\right\|_q + \|\mathbf{e}_{n,m}\|_q \\
&= \mathcal{O}((t_{n+1} - t_n)^{p+1}) + \|\mathbf{e}_{n,m}\|_q .
\end{aligned}$$

Notice that since $G_{\boldsymbol{\theta}}(\mathbf{x}_{t_m}, t_m, t_m) = \mathbf{x}_{t_m} = G(\mathbf{x}_{t_m}, t_m, t_m; \boldsymbol{\phi})$, $\mathbf{e}_{m,m} = \mathbf{0}$.

So we can obtain via induction that

$$\begin{aligned}
\|\mathbf{e}_{n+1,m}\|_q &\le \|\mathbf{e}_{m,m}\|_q + \sum_{k=m}^{n-1} \mathcal{O}((t_{k+1} - t_k)^{p+1}) \\
&= \sum_{k=m}^{n-1} \mathcal{O}((t_{k+1} - t_k)^{p+1}) \\
&\le \mathcal{O}((\Delta_N t)^p)(t_n - t_m).
\end{aligned}$$

■

Indeed, an analogue of Proposition 2 holds for time-conditional feature extractors.

Let $d_t(\cdot, \cdot)$ be a LPIPS-like metric determined by a time-conditional feature extractor $\mathcal{F}_t$. That is, $d_t(\mathbf{x}, \mathbf{y}) = \|\mathcal{F}_t(\mathbf{x}) - \mathcal{F}_t(\mathbf{y})\|_q$ for $q \ge 1$. We can similarly derive

$$\sup_{\mathbf{x} \in \mathbb{R}^D} d_{t_m}\big(G_{\boldsymbol{\theta}}(\mathbf{x}, t_n, t_m), G(\mathbf{x}, t_n, t_m; \boldsymbol{\phi})\big) = \mathcal{O}((\Delta_N t)^p)(t_n - t_m).$$

D.4 Proof of Proposition 4

Proof of Proposition 4. We first prove that for any $t \in [0, T]$ and $s \le t$, as $N \to \infty$,

$$\sup_{\mathbf{x} \in \mathbb{R}^D} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t, s), s, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t, s; \boldsymbol{\phi}), s, 0)\|_2 \to 0. \tag{9}$$

We may assume $\{t_n\}_{n=1}^N$ so that $t_m = s$, $t_n = t$, and $t_{m+1} \to s$, $t_{n+1} \to t$ as $\Delta_N t \to \infty$.

$$\begin{aligned}
&\sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t, s), s, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t, s; \boldsymbol{\phi}), s, 0)\|_2 \\
\le &\sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t, s), s, 0), G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t_{n+1}, t_{m+1}; \boldsymbol{\phi}), t_{m+1}, 0)\|_2 \\
+ &\sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t_{n+1}, t_{m+1}; \boldsymbol{\phi}), t_{m+1}, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t_{n+1}, t_{m+1}; \boldsymbol{\phi}), t_{m+1}, 0)\|_2 \\
+ &\sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t_{n+1}, t_{m+1}; \boldsymbol{\phi}), t_{m+1}, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t, s; \boldsymbol{\phi}), s, 0)\|_2
\end{aligned}$$

Since both G and $G_{\boldsymbol{\theta}_N}$ are uniform continuous on $\mathbb{R}^D \times [0, T] \times [0, T]$, together with Proposition 2, we obtain Eq. (9) as $\Delta_N t \to \infty$.

In particular, Eq. (9) implies that when $N \to \infty$

$$\sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, T, 0), 0, 0) - G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, T, 0; \boldsymbol{\phi}), 0, 0)\|_2$$

$$= \sup_{\mathbf{x}} \|G_{\boldsymbol{\theta}_N}(\mathbf{x}, T, 0) - G(\mathbf{x}, T, 0; \boldsymbol{\phi})\|_2 \to 0.$$

This implies that $p_{\boldsymbol{\theta}_N}(\cdot)$, the pushforward distribution of p_T induced by $G_{\boldsymbol{\theta}_N}(\cdot, T, 0)$, converges in distribution to $p_\phi(\cdot)$. Note that since $\{G_{\boldsymbol{\theta}_N}\}_N$ is uniform Lipschitz

$$\|G_{\boldsymbol{\theta}}(\mathbf{x}, t, s) - G_{\boldsymbol{\theta}}(\mathbf{x}', t, s)\|_2 \le L \|\mathbf{x} - \mathbf{x}'\|_2, \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^D, t, s \in [0, T], \text{ and } \boldsymbol{\theta},$$

$\{G_{\boldsymbol{\theta}_N}\}_N$ is asymptotically uniformly equicontinuous. Moreover, $\{G_{\boldsymbol{\theta}_N}\}_N$ is uniform bounded in $\boldsymbol{\theta}_N$. Therefore, the converse of Scheffé's theorem [51, 52] implies that $\|p_{\boldsymbol{\theta}_N}(\cdot) - p_{\boldsymbol{\phi}}(\cdot)\|_\infty \to 0$ as $N \to \infty$. Similar argument can be adapted to prove $\|p_{\boldsymbol{\theta}_N}(\cdot) - p_{\text{data}}(\cdot)\|_\infty \to 0$ as $N \to \infty$ if the regression target $p_{\boldsymbol{\phi}}(\cdot)$ is replaced with $p_{\text{data}}(\cdot)$. ■

D.5 Proof of Proposition ??

Lemma 6. *Let $f\colon \mathbb{R}^D \times [0, T] \to \mathbb{R}^D$ be a function which satisfies the following conditions:*

(a) $f(\cdot, t)$ is Lipschitz for any $t \in [0, T]$: there is a function $L(t) \ge 0$ so that for any $t \in [0, T]$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$

$$\|f(\mathbf{x}, t) - f(\mathbf{y}, t)\| \le L(t) \|\mathbf{x} - \mathbf{y}\|,$$

(b) Linear growth in $\mathbf{x}$: there is a L^1- integrable function $M(t)$ so that for any $t \in [0, T]$ and $\mathbf{x} \in \mathbb{R}^D$

$$\|f(\mathbf{x}, t)\| \le M(t)(1 + \|\mathbf{x}\|).$$

Consider the following ODE

$$\mathbf{x}'(\tau) = f(\mathbf{x}(\tau), \tau) \quad \text{on } [0, T]. \tag{10}$$

Fix a $t \in [0, T]$, the solution operator $\mathcal{T}$ of Eq. (10) with an initial condition $\mathbf{x}_t$ is defined as

$$\mathcal{T}[\mathbf{x}_t](s) := \mathbf{x}_t + \int_t^s f(\mathbf{x}(\tau; \mathbf{x}_t), \tau) \, \mathrm{d}\tau, \quad s \in [t, T]. \tag{11}$$

Here $\mathbf{x}(\tau; \mathbf{x}_t)$ denotes the solution at time τ starting from the initial value $\mathbf{x}_t$. Then $\mathcal{T}$ is an injective operator. Moreover, $\mathcal{T}[\cdot](s)\colon \mathbb{R}^D \to \mathbb{R}^D$ is bi-Lipschitz; that is, for any $\mathbf{x}_t, \hat{\mathbf{x}}_t \in \mathbb{R}^D$

$$e^{-L(s-t)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \le \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 \le e^{L(t-s)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2. \tag{12}$$

Here $L := \sup_{t \in [0,T]} L(t) < \infty$. In particular, if $\mathbf{x}_t \ne \hat{\mathbf{x}}_t$, $\mathcal{T}[\mathbf{x}_t](s) \ne \mathcal{T}[\hat{\mathbf{x}}_t](s)$ for all $s \in [t, T]$.

Proof of Lemma 6. Assumptions (a) and (b) ensure the solution operator in Eq. (11) is well-defined by applying Carathéodory-type global existence theorem [53]. We denote $\mathcal{T}[\mathbf{x}_t](s)$ as $\mathbf{x}(s; \mathbf{x}_t)$. We need to prove that for any distinct initial values $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ starting from t, $\mathcal{T}[\mathbf{x}_t] \not\equiv \mathcal{T}[\hat{\mathbf{x}}_t]$. Suppose on the contrary that there is an $s_0 \in [t, T]$ so that $\mathcal{T}[\mathbf{x}_t](s_0) = \mathcal{T}[\hat{\mathbf{x}}_t](s_0)$. For $s \in [t_0, s_0]$, consider $\mathbf{y}(s; \mathbf{x}_t) := \mathbf{x}(t + s_0 - s; \mathbf{x}_t)$ and $\mathbf{y}(s; \hat{\mathbf{x}}_t) := \mathbf{x}(t_0 + s_0 - s; \hat{\mathbf{x}}_t)$. Then both $\mathbf{y}(s; \mathbf{x}_t)$ and $\mathbf{y}(s; \hat{\mathbf{x}}_t)$ satisfy the following ODE

$$\begin{cases} \mathbf{y}'(s) = -f(\mathbf{y}(s), s), & s \in [t, s_0] \\ \mathbf{y}(t) = \mathcal{T}[\mathbf{x}_t](s_0) = \mathcal{T}[\hat{\mathbf{x}}_t](s_0) \end{cases} \tag{13}$$

Thus, the uniqueness theorem of solution to Eq. (13) leads to $\mathbf{y}(s_0; \mathbf{x}_t) = \mathbf{y}(s_0; \hat{\mathbf{x}}_t)$, which means $\mathbf{x}_t = \hat{\mathbf{x}}_t$. This contradicts to the assumption. Hence, $\mathcal{T}$ is injective.

Now we show that $\mathcal{T}[\cdot](s)\colon \mathbb{R}^D \to \mathbb{R}^D$ is bi-Lipschitz for any $s \in [t, T]$. For any $\mathbf{x}_t, \hat{\mathbf{x}}_t \in \mathbb{R}^D$,

$$\begin{aligned} \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 &= \|\mathbf{x}(s; \mathbf{x}_t) - \hat{\mathbf{x}}(s; \hat{\mathbf{x}}_t)\|_2 \\ &\le \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \int_t^s \|f(\mathbf{x}(\tau; \mathbf{x}_t), \tau) - f(\hat{\mathbf{x}}(\tau; \hat{\mathbf{x}}_t), \tau)\|_2 \, \mathrm{d}\tau \\ &\le \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + L \int_t^s \|\mathbf{x}(\tau; \mathbf{x}_t) - \hat{\mathbf{x}}(\tau; \hat{\mathbf{x}}_t)\|_2 \, \mathrm{d}\tau. \end{aligned}$$

By applying Gröwnwall's lemma, we obtain

$$\|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 = \|\mathbf{x}(s;\mathbf{x}_t) - \hat{\mathbf{x}}(s;\hat{\mathbf{x}}_t)\|_2 \leq e^{L(s-t)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 . \tag{14}$$

On the other hand, consider the reverse time ODE of Eq. (10) by setting $\tau = \tau(u) := t + s - u$, $\mathbf{y}(u) := \mathbf{x}(t+s-u)$, and $h(\mathbf{y}(u), u) := -f(\mathbf{y}(u), t+s-u)$, then $\mathbf{y}$ satisfies the following equation

$$\mathbf{y}'(u) = h(\mathbf{y}(u), u), \quad u \in [t, s]. \tag{15}$$

Similarly, we define the solution operator to Eq. (15) as

$$\mathcal{S}[\mathbf{y}_t](s) := \mathbf{y}_t + \int_t^s h(\mathbf{y}(u;\mathbf{y}_t), u)\, \mathrm{d}u. \tag{16}$$

Here $\mathbf{y}_t$ denotes the initial value of Eq. (15) and $\mathbf{y}(u;\mathbf{y}_t)$ is the solution starting from $\mathbf{y}_t$. Due to the Carathéodory-type global existence theorem, the operator $\mathcal{S}[\cdot](s)$ is well-defined and

$$\mathcal{S}[\mathbf{x}(s;\mathbf{x}_t)](s) = \mathbf{x}_t, \quad \mathcal{S}[\hat{\mathbf{x}}(s;\mathbf{x}_t)](s) = \hat{\mathbf{x}}_t.$$

For simplicity, let $\mathbf{y}_t := \mathbf{x}(s;\mathbf{x}_t)$ and $\hat{\mathbf{y}}_t := \hat{\mathbf{x}}(s;\mathbf{x}_t)$. Also, denote the solutions starting from initial values $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ as $\mathbf{y}(u;\mathbf{y}_t)$ and $\hat{\mathbf{y}}(u;\hat{\mathbf{y}}_t)$, respectively. Therefore, using a similar argument, we obtain

$$\begin{aligned}
\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 &= \|\mathcal{S}[\mathbf{y}_t](s) - \mathcal{S}[\hat{\mathbf{y}}_t](s)\|_2 \\
&\leq \|\mathbf{x}(s;\mathbf{x}_t) - \hat{\mathbf{x}}(s;\mathbf{x}_t)\|_2 + \int_t^s \|h(\mathbf{y}(u;\mathbf{y}_t), u) - h(\hat{\mathbf{y}}(u;\hat{\mathbf{y}}_t), u)\|_2 \,\mathrm{d}u \\
&\leq \|\mathbf{x}(s;\mathbf{x}_t) - \hat{\mathbf{x}}(s;\mathbf{x}_t)\|_2 + L\int_t^s \|\mathbf{y}(u;\mathbf{y}_t) - \hat{\mathbf{y}}(u;\hat{\mathbf{y}}_t)\|_2 \,\mathrm{d}u. \\
&= \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 + L\int_t^s \|\mathbf{y}(u;\mathbf{y}_t) - \hat{\mathbf{y}}(u;\hat{\mathbf{y}}_t)\|_2 \,\mathrm{d}u.
\end{aligned}$$

By applying Gröwnwall's lemma, we obtain

$$\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \leq e^{L(s-t)} \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 .$$

Therefore,

$$e^{-L(s-t)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \leq \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 .$$

■

Proof of Proposition ??. With the definition of $G(\mathbf{x}_t, t, s; \boldsymbol{\phi})$, we obtain

$$\begin{aligned}
G(\mathbf{x}_t, t, s; \boldsymbol{\phi}) &= \frac{s}{t}\mathbf{x}_t + (1 - \frac{s}{t}) g(\mathbf{x}_t, t, s; \boldsymbol{\phi}) \\
&= \mathbf{x}_t + \int_t^s \frac{\mathbf{x}_u - D_{\boldsymbol{\phi}}(\mathbf{x}_u, u)}{u} \,\mathrm{d}u.
\end{aligned}$$

Here, $g(\mathbf{x}_t, t, s; \boldsymbol{\phi}) = \mathbf{x}_t + \frac{t}{t-s}\int_t^s \frac{\mathbf{x}_u - D_{\boldsymbol{\phi}}(\mathbf{x}_u, u)}{u} \,\mathrm{d}u$. Thus, the result follows by applying Lemma 6 to the integral form of $G(\mathbf{x}_t, t, s; \boldsymbol{\phi})$.

■

D.6 Proof of Proposition ??

Lemma 7. *Let X be a random vector on $\mathbb{R}^D$ and $h\colon \mathbb{R}^D \to \mathbb{R}^D$ be a bi-Lipschitz mapping with Lipschitz constant $L > 0$; namely, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$*

$$L^{-1} \|\mathbf{x} - \mathbf{y}\|_2 \leq \|h(\mathbf{x}) - h(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2 .$$

Then

$$L^{-2} Var(X) \leq Var(h(X)) \leq L^2 Var(X).$$

Proof of Lemma 7. Let Y be an i.i.d. copy of X. Then $h(X)$ and $h(Y)$ are also independent. Thus, $\text{cov}(X, Y) = 0$ and $\text{cov}(h(X), h(Y)) = 0$.

$$\begin{aligned} 2\text{Var}\left(h(X)\right) &= \text{Var}\left(h(X) - h(Y)\right) \\ &= \mathbb{E}\left[\left(h(X) - h(Y)\right)^2\right] - \left(\mathbb{E}\left[h(X) - h(Y)\right]\right)^2. \end{aligned} \tag{17}$$

Since $h(X)$ and $h(Y)$ are identically distributed, $\mathbb{E}\left[h(X) - h(Y)\right] = \mathbb{E}\left[h(X)\right] - \mathbb{E}\left[h(Y)\right] = 0$. Thus, by Lipschitzness of h

$$\begin{aligned} 2\text{Var}\left(h(X)\right) &= \mathbb{E}\left[\left(h(X) - h(Y)\right)^2\right] \\ &\leq L^2\mathbb{E}\left[\left(X - Y\right)^2\right] \\ &= 2L^2\text{Var}\left(X\right). \end{aligned} \tag{18}$$

The final equality follows the same reasoning as in Eq. (17). Likewise, we can apply the argument from Eq. (18) to show that

$$\begin{aligned} 2\text{Var}\left(h(X)\right) &= \mathbb{E}\left[\left(h(X) - h(Y)\right)^2\right] \\ &\geq L^{-2}\mathbb{E}\left[\left(X - Y\right)^2\right] \\ &= 2L^{-2}\text{Var}\left(X\right). \end{aligned}$$

Therefore, $L^{-2}\text{Var}\left(X\right) \leq \text{Var}\left(X\right) \leq L^2\text{Var}\left(X\right)$. ■

Proof of Proposition ??. For any $n \in \mathbb{N}$, since $G_{\boldsymbol{\theta}^*}(X_n, t_n, \sqrt{1-\gamma^2}t_{n+1})$ and Z_{n+1} are independent,

$$\begin{aligned} \text{Var}\left(X_{n+1}\right) &= \text{Var}\left(G_{\boldsymbol{\theta}^*}(X_n, t_n, \sqrt{1-\gamma^2}t_{n+1})\right) + \text{Var}\left(Z_{n+1}\right) \\ &= \text{Var}\left(G_{\boldsymbol{\theta}^*}(X_n, t_n, \sqrt{1-\gamma^2}t_{n+1})\right) + \gamma^2\sigma^2(t_{n+1}). \end{aligned} \tag{19}$$

Proposition **??** implies that $G_{\boldsymbol{\theta}^*}(\cdot, t_n, \sqrt{1-\gamma^2}t_{n+1})$ is bi-Lipschitz and that for any $\mathbf{x}, \mathbf{y}$

$$\begin{aligned} \zeta^{-1}(t_n, t_{n+1}, \gamma)\left\|\mathbf{x} - \mathbf{y}\right\|_2 &\leq \left\|G_{\boldsymbol{\theta}^*}(\mathbf{x}, t_n, \sqrt{1-\gamma^2}t_{n+1}) - G_{\boldsymbol{\theta}^*}(\mathbf{y}, t_n, \sqrt{1-\gamma^2}t_{n+1})\right\|_2 \\ &\leq \zeta(t_n, t_{n+1}, \gamma)\left\|\mathbf{x} - \mathbf{y}\right\|_2, \end{aligned} \tag{20}$$

where $\zeta(t_n, t_{n+1}, \gamma) = \exp\left(2L_{\boldsymbol{\phi}}(t_n - \sqrt{1-\gamma^2}t_{n+1})\right)$. Proposition **??** follows immediately from the inequalities (19) and (20). ■

D.7 Proof of Proposition ??

Proof of Proposition ??. $\{p_t\}_{t=0}^T$ is known to satisfy the Fokker-Planck equation [54] (under some technical regularity conditions). In addition, we can rewrite the Fokker-Planck equation of $\{p_t\}_{t=0}^T$ as the following equation (see Eq. (37) in [3])

$$\frac{\partial p_t}{\partial t} = -\text{div}\left(\mathbf{W}_t p_t\right), \quad \text{in } (0, T) \times \mathbb{R}^D \tag{21}$$

where $\mathbf{W}_t := -t\nabla \log p_t$.

Now consider the continuity equation for μ_t defined by $\mathbf{W}_t$

$$\frac{\partial \mu_t}{\partial t} = -\text{div}\left(\mathbf{W}_t \mu_t\right) \quad \text{in } (0, T) \times \mathbb{R}^D. \tag{22}$$

Since the score $\nabla \log p_t$ is of linear growth in $\mathbf{x}$ and upper bounded by a summable function in t, the vector field $\mathbf{W}_t := -t\nabla \log p_t \colon [0, T] \times \mathbb{R}^D \to \mathbb{R}^D$ satisfies that

$$\int_0^T \left(\sup_{\mathbf{x} \in K} \left\|\mathbf{W}_t(\mathbf{x})\right\|_2 + \text{Lip}(\mathbf{W}_t, K)\ \mathrm{d}t\right) < \infty,$$

for any compact set $K \subset \mathbb{R}^D$. Here $\mathrm{Lip}(\mathbf{W}_t, K)$ denotes the Lipschitz constant of $\mathbf{W}_t$ on K.

Thus, Proposition 8.1.8 of [55] implies that for p_T-a.e. $\mathbf{x}$, the following reverse time ODE (which is the Eq. (**??**)) admits a unique solution on $[0, T]$

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t} X_t(\mathbf{x}) = \mathbf{W}_t \left(X_t(\hat{\mathbf{x}}) \right) \\ X_T(\hat{\mathbf{x}}) = \mathbf{x}. \end{cases} \tag{23}$$

Moreover, $\mu_t = X_t \sharp p_T$, for $t \in [0, T]$. By applying the uniqueness for the continuity equation (Proposition 8.1.7 of [55]) and the uniqueness of Eq. (23), we have $p_t = \mu_t = X_t \sharp p_T = \mathcal{T}_{T \to t} \sharp p_T$ for $t \in [0, T]$. Again, since the uniqueness theorem with the given p_T, we obtain $p_s = \mathcal{T}_{t \to s} \sharp p_t$ for any $t \in [0, T]$ and $s \in [0, t]$.

■