Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion

Dongjun Kim[*][†] **& Chieh-Hsin Lai**[*]
Sony AI
Tokyo, Japan
`dongjoun57@kaist.ac.kr, chieh-hsin.lai@sony.com`

Wei-Hsiang Liao & Naoki Murata & Yuhta Takida & Toshimitsu Uesaka & 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) (Song et al., 2023) 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 – in a single forward pass – 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 achieves new state-of-the-art FIDs for single-step diffusion model sampling on CIFAR-10 (FID 1.73) and ImageNet at 64×64 resolution (FID 2.06). CTM also enables a new family of sampling schemes, both deterministic and stochastic, involving long jumps along the ODE solution trajectories. It consistently improves sample quality as computational budgets increase, avoiding the degradation seen in CM. Furthermore, CTM's access to the score accommodates all diffusion model inference techniques, including exact likelihood computation.

1 Introduction

Deep generative models encounter distinct training and sampling challenges. Variational Autoencoder (VAE) (Kingma & Welling, 2013) can be trained easily but may suffer from posterior collapse, resulting in blurry samples, while Generative Adversarial Network (GAN) (Goodfellow et al., 2014) generates high-quality samples but faces training instability. Conversely, Diffusion Model (DM) (Sohl-Dickstein et al., 2015;
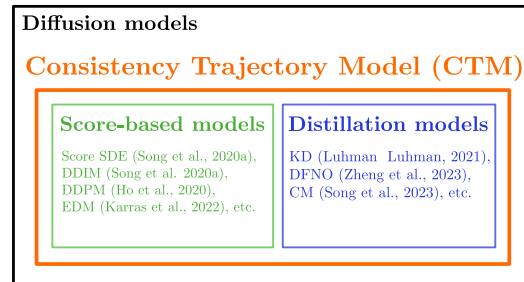
Figure 1: A schematic illustration of CTM.

---

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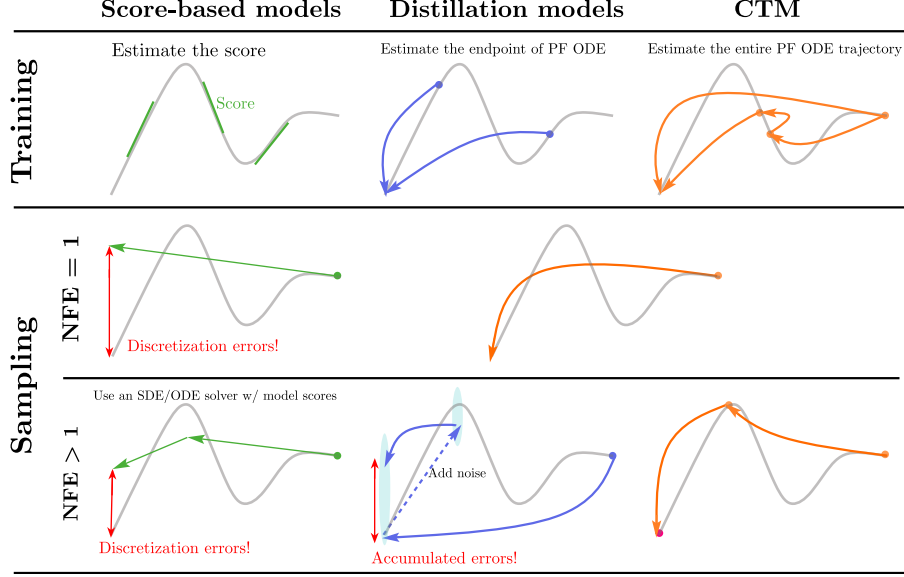[†]Work done during an internship at SONY AI

Figure 2: Training and sampling comparisons of score-based and distillation models with CTM. Score-based models exhibit discretization errors during SDE/ODE solving, while distillation models can accumulate errors in multistep sampling. CTM mitigates these issues with γ-sampling ($\gamma = 0$).

Ho et al., 2020; Song et al., 2020b) addresses these issues by learning the score (i.e., gradient of log-density) (Song & Ermon, 2019), which can generate high quality samples. However, compared to VAE and GAN excelling at fast sampling, DM involves a gradual denoising process that slows down sampling, requiring numerous model evaluations.

Score-based diffusion models synthesize data by solving the reverse-time (stochastic or deterministic) process corresponding to a prescribed forward process that adds noise to the data (Song & Ermon, 2019; Song et al., 2020b). Although advanced numerical solvers (Lu et al., 2022b; Zhang & Chen, 2022) of Stochastic Differential Equations (SDE) or Ordinary Differential Equations (ODE) substantially reduce the required Number of Function Evaluations (NFE), further improvements are challenging due to the intrinsic discretization error present in all solvers (De Bortoli et al., 2021). Recent developments in sample efficiency thus focus on directly estimation of the integral along the sample trajectory, amortizing the computational cost of numerical solvers. *Distillation models* (Salimans & Ho, 2021) in Figure 1, exemplified by the Consistency Model (CM) (Song et al., 2023), presents a promising approach for estimating the integration with a single NFE (Figure 2). However, their generation quality does not improve as NFE increase, and there is no straightforward mechanism for balancing computational resources (NFE) with quality.

This paper introduces the *Consistency Trajectory Model* (CTM) as a unified framework simultaneously assessing both the integrand (score function) and the integral (sample) of the Probability Flow (PF) ODE, thus bridging score-based and distillation models (Figure 1). CTM estimates both infinitesimal steps (score function) and long steps (integral over any time horizon) of the PF ODE from any initial condition, providing increased flexibility at inference time. Its score evaluation capability accommodates a range of score-based sampling algorithms based on solving differential

Figure 3: SOTA on CIFAR-10. Closeness to the origin indicates better performance.

equations (Song et al., 2020b), expanding its applicability across various domains (Saharia et al., 2022a). In particular, CTM enables exact likelihood computation, setting it apart from previous distillation models. Additionally, its integral approximation capability facilitates the incorporation of distillation sampling methods (Salimans & Ho, 2021; Song et al., 2023) that involve long "jumps" along the solution trajectory. This unique feature enables a novel sampling method called γ-*sampling*,
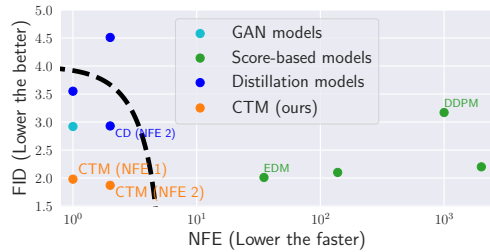
which alternates forward and backward jumps along the solution trajectory, with γ governing the level of stochasticity.

CTM's dual modeling capability for both infinitesimal and long steps of the PF ODE greatly enhances its training flexibility as well. It allows concurrent training with reconstruction loss, denoising diffusion loss, and adversarial loss within a unified framework. Notably, by incorporating CTM's training approach with γ-sampling, we achieve the new State-Of-The-Art (SOTA) performance in both density estimation and image generation for CIFAR-10 (Krizhevsky et al., 2009) (Figure 3) and ImageNet (Russakovsky et al., 2015) at a resolution of 64×64 (Table 2).

2 PRELIMINARY

In DM (Sohl-Dickstein et al., 2015; Song et al., 2020b), the encoder structure is formulated using a set of continuous-time random variables defined by a fixed forward diffusion process[1],
$$\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}_t,$$
initialized by the data variable, $\mathbf{x}_0 \sim p_{\text{data}}$. A reverse-time process (Anderson, 1982) 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 Song et al. (2020b), 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 (Efron, 2011), 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.

In practice, the denoiser $\mathbb{E}[\mathbf{x}|\mathbf{x}_t]$ is approximated using a neural network D_ϕ, obtained by minimizing the Denoising Score Matching (DSM) (Vincent, 2011; Song et al., 2020b) loss $\mathbb{E}_{\mathbf{x}_0,t,p_{0t}(\mathbf{x}|\mathbf{x}_0)}[\|\mathbf{x}_0 - D_\phi(\mathbf{x},t)\|_2^2]$, 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
$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \frac{\mathbf{x}_t - D_\phi(\mathbf{x}_t,t)}{t}.$$
Sampling from DM involves solving the PF ODE, equivalent to computing the integral
$$\int_T^0 \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\,\mathrm{d}t = \int_T^0 \frac{\mathbf{x}_t - D_\phi(\mathbf{x}_t,t)}{t}\,\mathrm{d}t \iff \mathbf{x}_0 = \mathbf{x}_T + \int_T^0 \frac{\mathbf{x}_t - D_\phi(\mathbf{x}_t,t)}{t}\,\mathrm{d}t, \qquad (1)$$
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 $\texttt{Solver}(\mathbf{x}_T, T, 0; \phi)$ (with an initial value of $\mathbf{x}_T$ at time T and ending at time 0), can be directly applied to solve Eq. (1) (Song et al., 2020b). For instance, DDIM (Song et al., 2020a) corresponds to a 1st-order Euler solver, while EDM (Karras et al., 2022) introduces a 2nd-order Heun solver. Despite recent advancements in numerical solvers (Lu et al., 2022b; Zhang & Chen, 2022), further improvements may be challenging due to the inherent discretization error present in all solvers (De Bortoli et al., 2021), ultimately limiting the sample quality obtained with few NFEs.

Distillation Sampling Distillation models (Salimans & Ho, 2021; Meng et al., 2023) successfully amortize the sampling cost by directly estimating the integral of Eq. (1) with a single neural network evaluation. However, their multistep sampling approach (Song et al., 2023) 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. We refer to Appendix A for a thorough literature review.

---

[1] This paper can be extended to VPSDE encoding (Song et al., 2020b) with re-scaling (Kim et al., 2022a).

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

To address the challenges in both score-based and distillation samplings, we introduce the Consistency Trajectory Model (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.

3.1 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 initial time t with an initial condition $\mathbf{x}_t$ to final time $s \leq t$:

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

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:*

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

Here, g satisfies:

- *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 $\mathbf{x}_t$.*

- *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 its limit: $g(\mathbf{x}_t, t, t) := \mathbb{E}[\mathbf{x}|\mathbf{x}_t]$.*

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

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

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

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 as follows:

Figure 4: Learning objectives of Score-based ($t = s$ line), distillation ($s = 0$ line), and CTM (upper triangle).

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

We remark that this parametrization satisfies the initial condition $G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t) = \mathbf{x}_t$ for free, leading to improved training stability[2]. Appendix C.1 offers further insights into this parametrization.
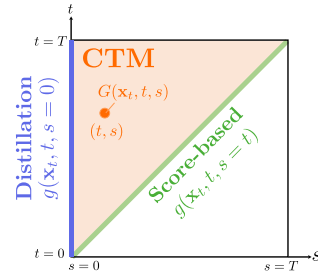
---

[2]Ensuring the initial condition's satisfaction is crucial for stable training. Directly estimating G with a network leads to rapid divergence, causing instability as the network output may deviate arbitrarily from the initial condition.

3.2 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:

$$G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx \texttt{Solver}(\mathbf{x}_t, t, s; \boldsymbol{\phi}). \tag{3}$$

In a scenario with no ODE discretization and no score approximation errors, $\texttt{Solver}$ perfectly reconstructs the PF ODE trajectory, and comparing the prediction and reconstruction in Eq. (3) leads $G_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t, s) = G(\mathbf{x}_t, t, s)$ at optimal $\boldsymbol{\theta}^*$, given sufficient network flexibility. The same conclusion holds by matching the local consistency:

$$G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx G_{\texttt{sg}(\boldsymbol{\theta})}(\texttt{Solver}(\mathbf{x}_t, t, t - \Delta t; \boldsymbol{\phi}), t - \Delta t, s), \tag{4}$$

where $\texttt{sg}(\cdot)$ is stop-gradient. With the initial condition ($\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t) = \mathbf{x}_t$) satisfied, matching Eq. (4) avoids collapsing to the trivial solution (Proposition 4 in Appendix B.1).

To estimate the entire solution trajectory with higher precision, we introduce *soft matching*, illustrated in Figure 5, ensuring consistency between prediction from $\mathbf{x}_t$ and the prediction from $\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi})$ for any $u \in [s, t)$:

$$G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s) \approx G_{\texttt{sg}(\boldsymbol{\theta})}(\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi}), u, s). \tag{5}$$

This soft matching spans two frameworks:

- As $u = s$, Eq. (3) enforces *global consistency matching*, i.e., a reconstruction loss.
- As $u = t - \Delta t$, Eq. (4) is *local consistency matching*. Additionally, if $s = 0$, it recovers CM's distillation loss.
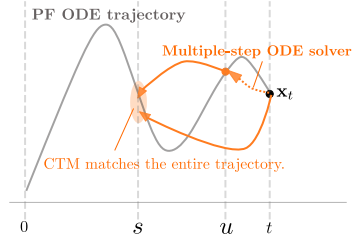

Figure 5: An illustration of CTM prediction and target at time s with an initial value $\mathbf{x}_t$.

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. (5), we could use either distance in pixel space or in feature space. However, the pixel distance may overemphasize the distance at large s due to the diffusion scale, requiring time-weighting adjustments. Furthermore, feature distance requires a time-conditional feature extractor, which can be expensive to train. Hence, we propose to use a feature distance d in clean data space by comparing

$$\mathbf{x}_{\text{est}}(\mathbf{x}_t, t, s) := G_{\texttt{sg}(\boldsymbol{\theta})}\Big(G_{\boldsymbol{\theta}}(\mathbf{x}_t, t, s), s, 0\Big)$$

$$\mathbf{x}_{\text{target}}(\mathbf{x}_t, t, u, s) := G_{\texttt{sg}(\boldsymbol{\theta})}\Big(G_{\texttt{sg}(\boldsymbol{\theta})}(\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi}), u, s), s, 0\Big). \tag{6}$$

CTM loss is defined as

$$\mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) := \mathbb{E}_{t \in [0,T]} \mathbb{E}_{s \in [0,t]} \mathbb{E}_{u \in [s,t)} \mathbb{E}_{\mathbf{x}_0, p_{0t}(\mathbf{x}|\mathbf{x}_0)} \Big[d\big(\mathbf{x}_{\text{target}}(\mathbf{x}, t, u, s), \mathbf{x}_{\text{est}}(\mathbf{x}, t, s)\big) \Big], \tag{7}$$

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 3 and 5) for details.

3.3 TRAINING CONSISTENCY TRAJECTORY MODELS

Training CTM with Eq. (7) 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 the DSM loss[3]

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

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

---

[3]We opt for the conventional DSM loss instead of minimizing $\mathbb{E}[\|D_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t)\|_2^2]$ with teacher score supervision, as the DSM loss's optimum can recover the true denoiser $g_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t, t) = \mathbb{E}[\mathbf{x}|\mathbf{x}_t]$, whereas teacher score supervision can only reach $D_{\boldsymbol{\phi}}$ as its optimum.

On the other hand, CTM, distilling from the teacher model, is constrained by the teacher's D_ϕ 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

Algorithm 1 CTM Training
___
1: **repeat**
2:     Sample $\mathbf{x}_0$ from data distribution
3:     Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
4:     Sample $t \in [0, T]$, $s \in [0, t]$, $u \in [s, t]$
5:     Calculate $\mathbf{x}_t = \mathbf{x}_0 + t\boldsymbol{\epsilon}$
6:     Calculate $\texttt{Solver}(\mathbf{x}_t, t, u; \boldsymbol{\phi})$
7:     Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\partial}{\partial \boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta})$
8:     Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \frac{\partial}{\partial \boldsymbol{\eta}}\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta})$
9: **until** converged
___

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) := \mathcal{L}_{\text{CTM}}(\boldsymbol{\theta}; \boldsymbol{\phi}) + \lambda_{\text{DSM}}\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta}), \tag{8}$$

in a single training framework, by optimizing $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta})$. Here, $\lambda_{\text{DSM}} \geq 0$ and $\lambda_{\text{GAN}} \geq 0$ are the weighting functions, see Algorithm 1.

4 SAMPLING WITH CTM

CTM enables score evaluation through $g_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, t)$, supporting standard score-based sampling with ODE/SDE solvers. In high-dimensional image synthesis, as shown in Figure 6's left two columns, CTM performs comparably to EDM using Heun's method as a PF ODE solver.

CTM additionally enables time traversal along the solution trajectory, allowing for the newly introduced γ-*sampling* method, refer to Algorithm 3 and Figure 7. 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, γ-sampling denoises $\mathbf{x}_{t_0}$ to time $\sqrt{1 - \gamma^2}t_1$ with $G_{\boldsymbol{\theta}}(\mathbf{x}_{t_0}, t_0, \sqrt{1 - \gamma^2}t_1)$, and perturb this denoised sample with forward diffusion to the noise level at time t_1. It iterates this back-and-forth traversal until reaching to time $t_N = 0$.
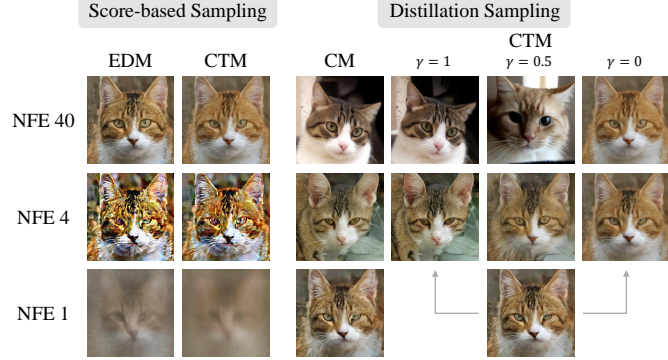

Figure 6: Comparison of score-based models (EDM), distillation models (CM), and CTM with various sampling methods and NFE trained on AFHQ-cat (Choi et al., 2020) 256×256.

Our γ-sampling is a new distillation sampler that unifies previously proposed sampling techniques, including distillation sampling and score-based sampling.

- Figure 7-(a): When $\gamma = 1$, it coincides to the multistep sampling introduced in CM, which is fully stochastic and results in semantic variation when NFE changes, e.g., compare samples of NFE 4 and 40 with the deterministic sample of NFE 1 in the third column of Figure 6. With the fixed $\mathbf{x}_T$, CTM reproduces CM's samples in the fourth column of Figure 6.

- Figure 7-(c): When $\gamma = 0$, it becomes the deterministic distillation sampling that estimates the solution of the PF ODE. A key distinction between the γ-sampling and score-based sampling is that CTM avoids sampling errors by directly estimating Eq. (2). However, score-based samplers like DDIM (1st-order Euler solver) or EDM (2nd-order Heun solver) are susceptible to discretization errors from Taylor approximation, especially with small NFE. (the leftmost column of Figure 6). Deterministic nature as $\gamma = 0$ ensures the sample semantic preserved across NFE changes, visualized in the rightmost column of Figure 6.

- Figure 7-(b): When $0 < \gamma < 1$, it generalizes the EDM's stochastic sampler (Algorithm 2). Appendix B.3 shows that γ-sampling's sample variances scale proportionally with γ^2.
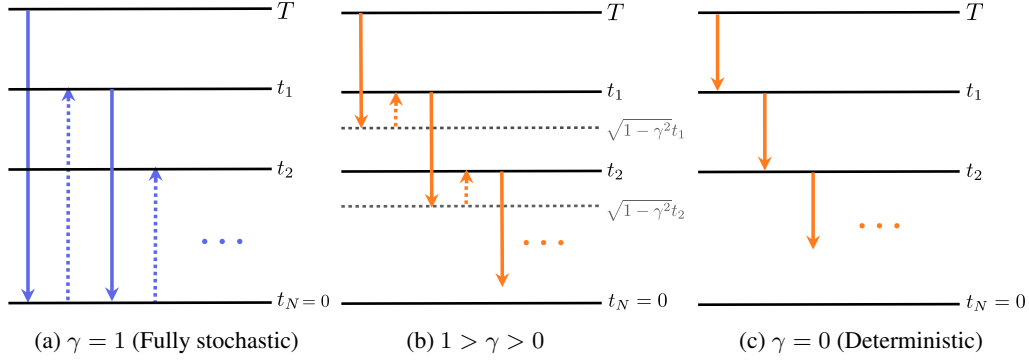
(a) $\gamma = 1$ (Fully stochastic) (b) $1 > \gamma > 0$ (c) $\gamma = 0$ (Deterministic)

Figure 7: Illustration of γ-sampling with varying γ value. It denoises with the network evaluation and iteratively diffuses the sample in reverse by $(t_n \xrightarrow{\text{Denoise}} \sqrt{1 - \gamma^2} t_{n+1} \xrightarrow{\text{Noisify}} t_{n+1})_{n=0}^{N-1}$.
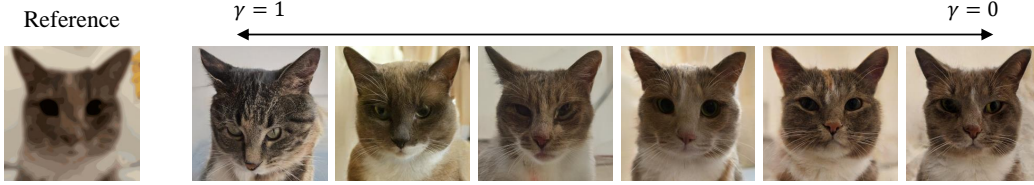

Figure 8: γ controls sample variance in stroke-based generation (see Appendix C.3).

The optimal choice of γ depends on practical usage and empirical configuration (Karras et al., 2022; Xu et al., 2023). Figure 8 demonstrates γ-sampling in stroke-based generation (Meng et al., 2021), revealing that the sampler with $\gamma = 1$ leads to significant semantic deviations from the reference stroke, while smaller γ values yield closer semantic alignment and maintain high fidelity. In contrast, Figure 9 showcases γ's impact on generation performance. In Figure 9-(a), γ has less influence with small NFE, but the setup with $\gamma \approx 0$ is the only one that resembles the performance of the Heun's solver as NFE increases. Additionally, CM's multistep sampler ($\gamma = 1$) significantly degrades sample quality as NFE increases. This quality deterioration concerning γ becomes more pronounced with higher NFEs, shown in Figure 9-(b), potentially attributed to error accumulation during the iterative long "jumps" for denoising. We explain this phenomenon using a 2-step γ-sampling example in the following theorem, see Theorem 8 for a generalized result for N-steps.

Theorem 2 ((Informal) 2-steps γ-sampling)**.** *Let* $t \in (0, T)$ *and* $\gamma \in [0, 1]$*. Denote* $p_{\theta*, 2}$ *as the density obtained from the γ-sampler with the optimal CTM, following the transition sequence* $T \to \sqrt{1 - \gamma^2} t \to t \to 0$*, starting from* p_T*. Then* $D_{TV}(p_{data}, p_{\theta*, 2}) = \mathcal{O}\left(\sqrt{T - \sqrt{1 - \gamma^2} t + t}\right)$*.*

When it becomes N-steps, $\gamma = 1$-sampling iteratively conducts long jumps from t_n to 0 for each step n, which aggregates the error to be $\mathcal{O}(\sqrt{T + t_1 + \cdots + t_N})$. In contrast, such time overlap between jumps does not occur in $\gamma = 0$-sampling, eliminating the error accumulation, resulting in $\mathcal{O}(\sqrt{T})$ error, see Appendix C.2. In summary, CTM addresses challenges associated with large NFE in distillation models with $\gamma = 0$ and removes the discretization error in score-based models.
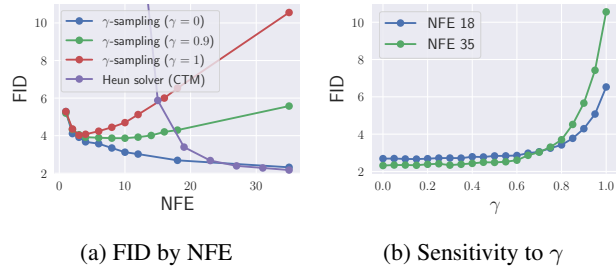

(a) FID by NFE (b) Sensitivity to γ

Figure 9: (a) CTM enables score-based sampling and distillation γ-sampling on CIFAR-10. (b) The FID degrade highlights the importance of trajectory learning.

Table 1: Performance comparisons on CIFAR-10.

Model	NFE	Unconditional		Conditional
		FID↓	NLL↓	FID↓
GAN Models				
BigGAN (Brock et al., 2018)	1	8.51	✗	-
StyleGAN-Ada (Karras et al., 2020)	1	2.92	✗	2.42
StyleGAN-D2D (Kang et al., 2021)	1	-	✗	2.26
StyleGAN-XL (Sauer et al., 2022)	1	-	✗	1.85
Diffusion Models – Score-based Sampling				
DDPM (Ho et al., 2020)	1000	3.17	3.75	-
DDIM (Song et al., 2020a)	100	4.16	-	-
	10	13.36	-	-
Score SDE (Song et al., 2020a)	2000	2.20	3.45	-
VDM (Kingma et al., 2021)	1000	7.41	2.49	-
LSGM (Vahdat et al., 2021)	138	2.10	3.43	-
EDM (Karras et al., 2022)	35	2.01	2.56	1.82
Diffusion Models – Distillation Sampling				
KD (Luhman & Luhman, 2021)	1	9.36	✗	-
DFNO (Zheng et al., 2023)	1	5.92	✗	-
Rectified Flow (Liu et al., 2022)	1	4.85	✗	-
PD (Salimans & Ho, 2021)	1	9.12	✗	-
CD (official report) (Song et al., 2023)	1	3.55	✗	-
CD (retrained)	1	10.53	✗	-
CD + GAN (Lu et al., 2023)	1	2.65	✗	-
CTM (ours)	1	1.98	2.43	1.73
PD (Salimans & Ho, 2021)	2	4.51	-	-
CD (Song et al., 2023)	2	2.93	-	-
CTM (ours)	2	1.87	2.43	1.63

Table 2: Performance comparisons on ImageNet 64×64.

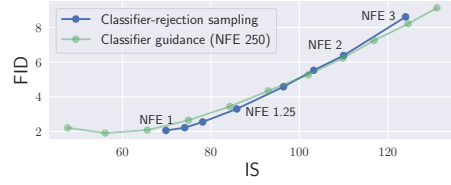| Model	NFE	FID↓	IS↑
ADM (Dhariwal & Nichol, 2021)	250	2.07	-
EDM (Karras et al., 2022)	79	2.44	48.88
BigGAN-deep (Brock et al., 2018)	1	4.06	-
StyleGAN-XL (Sauer et al., 2022)	1	2.09	**82.35**
Diffusion Models – Distillation Sampling			
PD (Salimans & Ho, 2021)	1	15.39	-
BOOT (Gu et al., 2023)	1	16.3	-
CD (Song et al., 2023)	1	6.20	40.08
CTM (ours)	1	2.06	69.83
PD (Salimans & Ho, 2021)	2	8.95	-
CD (Song et al., 2023)	2	4.70	-
CTM (ours)	2	**1.90**	63.90

Figure 10: FID-IS curve on ImageNet.

(a) EDM (79 NFE) (b) CTM w/o GAN (1 NFE) (c) CTM w/ GAN (1 NFE)

Figure 11: Samples generated by (a) EDM, (b) CTM without GAN ($\lambda_{\text{GAN}} = 0$), and (c) CTM with GAN ($\lambda_{\text{GAN}} = 1$). CTM distills knowledge from EDM (teacher) and employs adversarial training for further refining fine-grained details. More generated samples are demonstrated in Appendix E.

(d) Without classifier-rejection sampling (NFE 1) (e) With classifier-rejection sampling (avg. NFE 2)

Figure 12: Random samples (Siberian Husky) (d) with and (e) without classifier-free sampling.

5 EXPERIMENTS

5.1 STUDENT (CTM) BEATS TEACHER (DM) – QUANTITATIVE ANALYSIS

We evaluate CTM on CIFAR-10 and ImageNet 64×64, using the pre-trained diffusion checkpoints from EDM for CIFAR-10 and CM for ImageNet as the teacher models. We adopt EDM's training configuration for $\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta})$ and employ StyleGAN-XL's (Sauer et al., 2022) discriminator for $\mathcal{L}_{\text{GAN}}(\boldsymbol{\theta}, \boldsymbol{\eta})$. During training, we employ adaptive weights λ_{DSM} and λ_{GAN}, inspired by VQGAN (Esser et al., 2021) to balance DSM and GAN losses with the CTM loss. For both datasets, we utilize the DDPM architecture. For CIFAR-10, we take EDM's implementation; and for ImageNet, CM's implementation is used. On top of these architectures, we incorporate s-information via auxiliary temporal embedding with positional embedding (Vaswani et al., 2017), and add this embedding to the t-embedding. This training setup (Appendix D), along with the deterministic sampling ($\gamma = 0$), allows CTM's generation to outperform teacher models with NFE 1 and achieve SOTA FIDs with NFE 2.

CIFAR-10 CTM's NFE 1 generation excels both EDM and StyleGAN-XL with FID of 1.73 on conditional CIFAR-10, and CTM achieves the SOTA FID of 1.63 with 2 NFEs, surpassing all generative models. These results are obtained with the implementation based on the official PyTorch code of CM. However, retraining CM with this official PyTorch code yields FID of 10.53 (unconditional), higher than the reported FID of 3.55. Additionally, CTM's ability to approximate scores using $g_\theta(\mathbf{x}_t, t, t)$ enables evaluating Negative Log-Likelihood (NLL) (Song et al., 2021; Kim et al., 2022b), establishing a new SOTA NLL. This improvement can be attributed, in part, to CTM's reconstruction loss when $u = s$, and improved alignment with the oracle process (Lai et al., 2023a).

ImageNet CTM's generation surpasses both teacher EDM and StyleGAN-XL with NFE 1, outperforming previous models with no guidance (Dhariwal & Nichol, 2021), see Figure 11 for the comparison of CTM with the teacher model. Notably, all results in Tables 1 and 2 are achieved within 30K-100K training iterations, requiring only 10% of the iterations compared to CM and EDM.

Classifier-Rejection Sampling CTM's fast sampling enables classifier-rejection sampling. In the evaluation, for each class, we select the top 50 samples out of $\frac{50}{1-r}$ samples based on predicted class probability, where r is the rejection ratio. This sampler, combined with NFE 1 sampling, consumes an average of NFE $\frac{1}{1-r}$. In Figure 10, CTM, employing cost-effective classifier-rejection sampling, shows a FID-IS trade-off comparable to classifier-guided results (Ho & Salimans, 2021) achieved with high NFEs of 250. Additionally, Figure 12 confirms that samples rejected by the classifier exhibit superior quality and maintain class consistency, in agreement with the findings of Ho & Salimans (2021). We employ the classifier at resolution of 64×64 provided by Dhariwal & Nichol (2021).

5.2 QUALITATIVE ANALYSIS

CTM Loss Figure 13 highlights the advantages of employing the proposed soft consistency matching in Eq. (5) during CTM training. It outperforms the local consistency matching (Eq. (4)). Additionally, it demonstrates comparable performance to the global consistency matching (Eq. 3) with NFE 1, superior performance with large NFE. Furthermore, soft matching is computationally efficient, enhancing the scalability of CTM.

(a) NFE 1   (b) NFE 18

Figure 13: Comparison of local, global, and the proposed soft consistency matching.

DSM Loss Figure 14 illustrates two benefits of incorporating $\mathcal{L}_{\mathrm{DSM}}$ with $\mathcal{L}_{\mathrm{CTM}}$. It preserves sample quality for small NFE unless DSM scale outweighs CTM. For large NFE sampling, it significantly improves sample quality due to accurate score estimation. Throughout the paper, we maintain $\lambda_{\mathrm{DSM}} = 1$ based on insights from Figure 14, unless otherwise specified.

(a) NFE 1   (b) NFE 18

Figure 14: The effect of DSM loss.

GAN Loss Analogous to the DSM loss, Figure 15 illustrates the advantages of incorporating the GAN loss for both small and large NFE sample quality. Figure 11 demonstrates that CTM can produce samples resembling those of EDM (teacher), with GAN refining local details. Throughout the paper, we adopt the warm-up strategy for GAN training: deactivate GAN training with $\lambda_{\mathrm{GAN}} = 0$ for
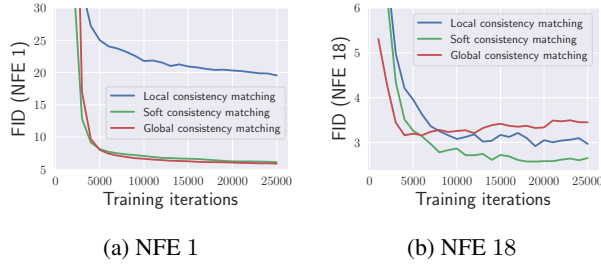

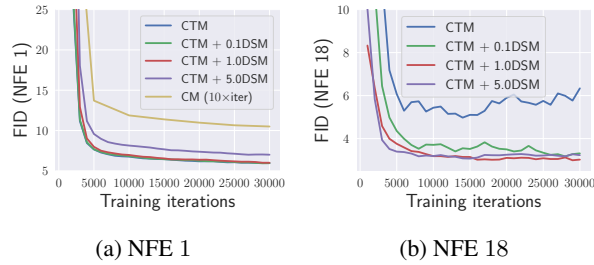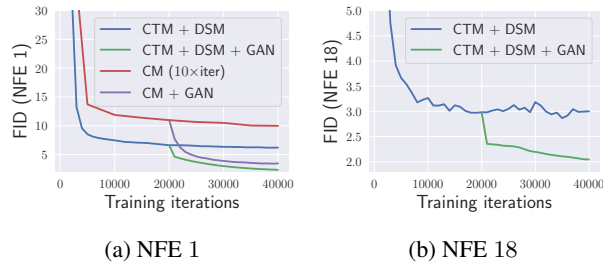(a) NFE 1   (b) NFE 18

Figure 15: The effect of GAN loss.

9

warm-up iterations and then activate GAN training with $\lambda_{\text{GAN}} = 1$, in line with the recommendation from VQGAN (Esser et al., 2021). This warm-up strategy is applied by default unless otherwise specified.

Training Without Pre-trained DM Leveraging our score learning capability, we replace the pre-trained score approximation, $D_\phi(\mathbf{x}_t, t)$, with CTM's approximation, $g_\theta(\mathbf{x}_t, t, t)$, allowing us to obtain the corresponding empirical PF ODE $d\mathbf{x}_t = \frac{\mathbf{x}_t - g_\theta(\mathbf{x}_t, t, t)}{t}$. Consequently, we can construct a pretrained-free target, $\hat{\mathbf{x}}_{\text{target}} := G_{\text{sg}(\theta)}(G_{\text{sg}(\theta)}(\text{Solver}(\mathbf{x}_t, t, u; \text{sg}(\theta)), u, s), s, 0)$, to replace $\mathbf{x}_{\text{target}}$ in computing the CTM loss $\mathcal{L}_{\text{CTM}}$. In contrast to $\mathbf{x}_{\text{target}}$ (Eq. (6)), $\hat{\mathbf{x}}_{\text{target}}$ eliminates the need for pre-trained DMs. When incorporated with DSM and GAN losses, it achieves a NFE 1 FID of 2.39 on unconditional CIFAR-10, a performance on par with pre-trained DMs. We highlight that the CTM loss can be used with or without the pre-trained DM, which is contrastive to CM with the Consistency Training loss for pretrained-free training that uses an ad-hoc noise injection technique.

6 CONCLUSION

CTM, a novel generative model, addresses issues in established models. With a unique training approach accessing intermediate PF ODE solutions, it enables 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

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

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

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.

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

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

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.

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)*, pp. 1–5. IEEE, 2023.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8188–8197, 2020.

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.

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.

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

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.

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

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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 {\&} Generative Modeling*, 2023.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

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.

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.

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.

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

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*, 2022a.

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, 2022b.

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*, pp. 11201–11228. PMLR, 2022c.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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. 2023a.

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*, 2023b.

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.

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.

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*, pp. 14429–14460. PMLR, 2022a.

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, 2022b.

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.

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

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.

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*, pp. 14297–14306, 2023.

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.

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

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

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*, pp. 10684–10695, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

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, 2022b.

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)*, pp. 1–5. IEEE, 2023.

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

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

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.

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

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, 2020b.

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.

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

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

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

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*, pp. 10347–10357. PMLR, 2021.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *arXiv preprint arXiv:2306.14878*, 2023.

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

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*, pp. 586–595, 2018.

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*, pp. 42390–42402. PMLR, 2023.

CONTENTS

A RELATED WORKS

Diffusion Models DMs excel in high-fidelity synthetic image and audio generation (Dhariwal & Nichol, 2021; Saharia et al., 2022b; Rombach et al., 2022), as well as in applications like media editing, restoration (Meng et al., 2021; Cheuk et al., 2023; Kawar et al., 2022; Saito et al., 2023; Hernandez-Olivan et al., 2023; Murata et al., 2023). Recent research aims to enhance DMs in sample quality (Kim et al., 2022b;a), density estimation (Song et al., 2021; Lu et al., 2022a), and especially, sampling speed (Song et al., 2020a).

Fast Sampling of DMs The SDE framework underlying DMs (Song et al., 2020b) has driven research into various numerical methods for accelerating DM sampling, exemplified by works such as (Song et al., 2020a; Zhang & Chen, 2022; Lu et al., 2022b). Notably, (Lu et al., 2022b) reduced the ODE solver steps to as few as 10-15. Other approaches involve learning the solution operator of ODEs (Zheng et al., 2023), discovering optimal transport paths for sampling (Liu et al., 2022), or employing distillation techniques (Luhman & Luhman, 2021; Salimans & Ho, 2021; Berthelot et al., 2023; Shao et al., 2023). However, previous distillation models may experience slow convergence or extended runtime. Gu et al. (2023) introduced a bootstrapping approach for data-free distillation. Furthermore, Song et al. (2023) 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 (Lu et al., 2023). 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 (Kim et al., 2022c), identified discrepancies between learned scores and ground truth scores. Recent developments have introduced various consistency concepts, showing their ability to enhance sample quality (Daras et al., 2023; Li et al., 2023), accelerate sampling speed (Song et al., 2023), and improve density estimation in diffusion modeling (Lai et al., 2023a). Notably, Lai et al. (2023b) 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, variance bounds for γ-sampling, and a more general form of accumulated errors induced by γ-sampling (cf. Theorem 2).

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.

$$\frac{\mathrm{d}\mathbf{x}_u}{\mathrm{d}u} = \frac{\mathbf{x}_u - D_{\boldsymbol{\phi}}(\mathbf{x}_u, u)}{u}, \quad u \in [0, T]. \tag{9}$$

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. (7) to the discrete time counterpart:

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

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 3. *Define $\Delta_N t := \max\limits_{n \in [\![1,N]\!]} \{|t_{n+1} - t_n|\}$. Assume that $G_{\boldsymbol{\theta}}$ is uniform Lipschitz in $\mathbf{x}$ 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 $\mathcal{L}_{CTM}^N(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$, 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. (4) 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

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

where

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

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

Proposition 4. *If there is a $\boldsymbol{\theta}_N$ so that $\tilde{\mathcal{L}}_{CTM}^N(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$, 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 3, 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 5, 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:

$$\mathbf{x}_{\text{target}}(\mathbf{x}_{t_n}, t_n, t_{n-1}, t_m) \equiv G(\mathbf{x}_{t_n}, t_n, t_m), \quad \text{for all } n \in [\![1,N]\!], m \in [\![0,n]\!], N \in \mathbb{N}. \tag{10}$$

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. (2).

Proposition 5. *Suppose that*

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

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

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

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

If for any N, there is a $\boldsymbol{\theta}_N$ such that $\mathcal{L}_{CTM}^N(\boldsymbol{\theta}_N; \boldsymbol{\phi}) = 0$. 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. (10) is satisfied, then $\|p_{\boldsymbol{\theta}_N}(\cdot) - p_{data}(\cdot)\|_\infty \to 0$ as $N \to \infty$.

B.2 NON-INTERSECTING TRAJECTORY OF THE OPTIMAL CTM

CTM learns distinct trajectories originating from various initial points $\mathbf{x}t$ and times t. In the following proposition, we demonstrate that the distinct trajectories derived by the optimal CTM, which effectively distills information from its teacher model ($G_{\boldsymbol{\theta}^*}(\cdot, t, s) \equiv G(\cdot, t, s; \boldsymbol{\phi})$ for any $t, s \in [0, T]$), do not intersect.

Proposition 6. *Suppose that a well-trained $\boldsymbol{\theta}^*$ such that $G_{\boldsymbol{\theta}^*}(\cdot, t, s) \equiv G(\cdot, t, s; \boldsymbol{\phi})$ for any $t, s \in [0, T]$, and that $D_{\boldsymbol{\phi}}(\cdot, t)$ is Lipschitz, i.e., there is a constant $L_{\boldsymbol{\phi}} > 0$ so that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ and $t \in [0, T]$*

$$\|D_{\boldsymbol{\phi}}(\mathbf{x}, t) - D_{\boldsymbol{\phi}}(\mathbf{y}, t)\|_2 \leq L_{\boldsymbol{\phi}} \|\mathbf{x} - \mathbf{y}\|_2 \,.$$

Then for any $s \in [0, t]$, the mapping $G_{\boldsymbol{\theta}^*}(\cdot, t, s) \colon \mathbb{R}^D \to \mathbb{R}^D$ is bi-Lipschitz. Namely, for any $\mathbf{x}_t, \mathbf{y}_t \in \mathbb{R}^D$

$$e^{-L_{\boldsymbol{\phi}}(t-s)} \|\mathbf{x}_t - \mathbf{y}_t\|_2 \leq \|G_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t, s) - G_{\boldsymbol{\theta}^*}(\mathbf{y}_t, t, s)\|_2 \leq e^{L_{\boldsymbol{\phi}}(t-s)} \|\mathbf{x}_t - \mathbf{y}_t\|_2 \,. \quad (11)$$

This implies that $\mathbf{x}_t \neq \mathbf{y}_t$, $G_{\boldsymbol{\theta}^*}(\mathbf{x}_t; t, s) \neq G_{\boldsymbol{\theta}^*}(\mathbf{y}_t; t, s)$ for all $s \in [0, t]$.

Specifically, the mapping from an initial value to its corresponding solution trajectory, denoted as $\mathbf{x}_t \mapsto G_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t, \cdot)$, is injective. Conceptually, this ensures that if we use guidance at intermediate times to shift a point to another guided-target trajectory, the guidance will continue to affect the outcome at $t = 0$.

B.3 VARIANCE BOUNDS OF γ-SAMPLING

Suppose the sampling timesteps are $T = t_0 > t_1 > \cdots > t_N = 0$. In Proposition 7, we analyze the variance of

$$X_{n+1} := G_{\boldsymbol{\theta}}(X_n, t_n, \sqrt{1 - \gamma^2} t_{n+1}) + Z_n,$$

resulting from n-step γ-sampling, initiated at

$$X_1 := G_{\boldsymbol{\theta}}(\mathbf{x}_{t_0}, t_0, \sqrt{1 - \gamma^2} t_1) + \gamma Z_0, \quad \text{where } Z_n \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \gamma^2 t_{n+1}^2)\mathbf{I}).$$

Here, 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 $t, s \in [0, T]$, for simplicity.

Proposition 7. *We have*

$$\zeta^{-1}(t_n, t_{n+1}, \gamma) Var(X_n) + \gamma^2 t_{n+1}^2 \leq Var(X_{n+1}) \leq \zeta(t_n, t_{n+1}, \gamma) Var(X_n) + \gamma^2 t_{n+1}^2,$$

where $\zeta(t_n, t_{n+1}, \gamma) = \exp\left(2L_{\boldsymbol{\phi}}(t_n - \sqrt{1 - \gamma^2} t_{n+1})\right)$ and $L_{\boldsymbol{\phi}}$ is a Lipschitz constant of $D_{\boldsymbol{\phi}}(\cdot, t)$.

In line with our intuition, CM's multistep sampling ($\gamma = 1$) yields a broader range of $Var(X_{n+1})$ compared to $\gamma = 0$, resulting in diverging semantic meaning with increasing sampling NFE.

B.4 ACCUMULATED ERRORS IN THE GENERAL FORM OF γ-SAMPLING.

We can extend Theorem 2 for two steps γ-sampling for the case of multisteps.

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 8 (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, $\mathcal{T}_{t \to s} \colon \mathbb{R}^D \to \mathbb{R}^D$ denotes the oracle transition mapping from t to s, determined by Eq. (2). 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 $\mathcal{T}_{t \to s}^{\boldsymbol{\theta}^*} \sharp p_t$, where $\mathcal{T}_{t \to s}^{\boldsymbol{\theta}^*}$ denotes the transition mapping associated with the optimal CTM trained from Eq. (8).

B.5 Transition Densities with the Optimal CTM

In this section, for simplicity, we assume the optimal CTM, $G_{\boldsymbol{\theta}^*} \equiv G$ with a well-learned $\boldsymbol{\theta}^*$, which recovers the true G-function. We establish that the density propagated by this optimal CTM from any time t to a subsequent time s aligns with the predefined density determined by the fixed forward process.

We now present the proposition ensuring alignment of the transited density.

Proposition 9. *Let $\{p_t\}_{t=0}^T$ be densities defined by the diffusion process Eq. (2), where $p_0 := p_{data}$. Denote $\mathcal{T}_{t \to s}(\cdot) := G(\cdot, t, s) \colon \mathbb{R}^D \to \mathbb{R}^D$ for any $t \geq s$. Suppose that the score $\nabla \log p_t$ satisfies that there is a function $L(t) \geq 0$ so that $\int_0^T |L(t)|\, \mathrm{d}t < \infty$ and*

   (i) Linear growth: $\|\nabla \log p_t(\mathbf{x})\|_2 \leq L(t)(1 + \|\mathbf{x}\|_2)$, for all $\mathbf{x} \in \mathbb{R}^D$

   (ii) Lipschitz: $\|\nabla \log p_t(\mathbf{x}) - \nabla \log p_t(\mathbf{y})\|_2 \leq L(t)\|\mathbf{x} - \mathbf{y}\|_2$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$.

Then for any $t \in [0, T]$ and $s \in [0, t]$, $p_s = \mathcal{T}_{t \to s} \sharp p_t$.

This theorem guarantees that by learning the optimal CTM, which possesses complete trajectory information, we can retrieve all true densities at any time using CTM.

C Algorithmic Details

C.1 Motivation of Parametrization

Our parametrization of $G_{\boldsymbol{\theta}}$ is affected from the discretized ODE solvers. For instance, the one-step Euler solver has the solution of

$$\mathbf{x}_s^{\text{Euler}} = \mathbf{x}_t - (t - s)\frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}|\mathbf{x}_t]}{t} = \frac{s}{t}\mathbf{x}_t + \left(1 - \frac{s}{t}\right)\mathbb{E}[\mathbf{x}|\mathbf{x}_t].$$

The one-step Heun solver is

$$\begin{aligned}
\mathbf{x}_s^{\text{Heun}} &= \mathbf{x}_t - \frac{t-s}{2}\left(\frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}|\mathbf{x}_t]}{t} + \frac{\mathbf{x}_s^{\text{Euler}} - \mathbb{E}[\mathbf{x}|\mathbf{x}_s^{\text{Euler}}]}{s}\right) \\
&= \mathbf{x}_t - \frac{t-s}{2}\left(\frac{\mathbf{x}_t}{t} + \frac{\mathbf{x}_s^{\text{Euler}}}{s}\right) + \frac{t-s}{2}\left(\frac{\mathbb{E}[\mathbf{x}|\mathbf{x}_t]}{t} + \frac{\mathbb{E}[\mathbf{x}|\mathbf{x}_s^{\text{Euler}}]}{s}\right) \\
&= \frac{s}{t}\mathbf{x}_t + \left(1 - \frac{s}{t}\right)\left(\left(1 - \frac{t}{2s}\right)\mathbb{E}[\mathbf{x}|\mathbf{x}_t] + \frac{t}{2s}\mathbb{E}[\mathbf{x}|\mathbf{x}_s^{\text{Euler}}]\right).
\end{aligned}$$

Again, the solver scales $\mathbf{x}_t$ with $\frac{s}{t}$ and multiply $1 - \frac{s}{t}$ to the second term. Therefore, our $G(\mathbf{x}_t, t, s) = \frac{s}{t}\mathbf{x}_t + (1 - \frac{s}{t})g(\mathbf{x}_t, t, s)$ is a natural way to represent the ODE solution.

For future research, we establish conditions enabling access to both integral and integrand expressions. Consider a continuous real-valued function $a(t, s)$. We aim to identify necessary conditions on $a(t, s)$ for the expression of G as:

$$G(\mathbf{x}_t, t, s) = a(t, s)\mathbf{x}_t + \left(1 - a(t, s)\right)h(\mathbf{x}_t, t, s),$$

for a vector-value function $h(\mathbf{x}_t, t, s)$ and that h satisfies:

- $\lim_{s \to t} h(\mathbf{x}_t, t, s)$ exists;
- it can be expressed algebraically with $\mathbb{E}.[\mathbf{x}|\mathbf{x}_t]$.

Starting with the definition of G, we can obtain

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

$$= a(t, s)\mathbf{x}_t + \big(1 - a(t, s)\big) \underbrace{\Big[\mathbf{x}_t + \frac{1}{1 - a(t, s)} \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}|\mathbf{x}_u]}{u} \, \mathrm{d}u\Big]}_{h(\mathbf{x}_t, t, s)}.$$

Suppose that there is a continuous function $c(t)$ so that

$$\lim_{s \to t} \frac{s - t}{1 - a(t, s)} = c(t),$$

then

$$\lim_{s \to t} h(\mathbf{x}_t, t, s) = \mathbf{x}_t + \lim_{s \to t} \Big[\frac{1}{1 - a(t, s)} \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}|\mathbf{x}_u]}{u} \, \mathrm{d}u\Big]$$

$$= \mathbf{x}_t + \lim_{s \to t} \Big[\frac{s - t}{1 - a(t, s)} \frac{\mathbf{x}_{t^*} - \mathbb{E}[\mathbf{x}|\mathbf{x}_{t^*}]}{t^*}\Big], \quad \text{for some } t^* \in [s, t]$$

$$= \mathbf{x}_t + c(t)\Big(\frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}|\mathbf{x}_t]}{t}\Big)$$

$$= \Big(\frac{1 + c(t)}{t}\Big)\mathbf{x}_t - \frac{c(t)}{t}\mathbb{E}[\mathbf{x}|\mathbf{x}_t].$$

The second equality follows from the mean value theorem (We omit the continuity argument details for Markov filtrations). Therefore, we obtain the desired property 2). We summarize the necessary conditions on $a(s, t)$ as:

$$\text{There is some continuous function } c(t) \text{ in } t \text{ so that } \lim_{s \to t} \frac{s - t}{1 - a(t, s)} = c(t). \tag{12}$$

We now explain the above observation with an example by considering EDM-type parametrization. Consider $c_{\text{skip}} = c_{\text{skip}}(t, s) := \sqrt{\frac{(s - \sigma_{\min})^2 + \sigma_{\text{data}}^2}{(t - \sigma_{\min})^2 + \sigma_{\text{data}}^2}}$ and $c_{\text{out}} = c_{\text{out}}(t, s) := \big(1 - \frac{s}{t}\big)$. Then $G(\mathbf{x}_t, t, s)$ can be expressed as

$$G(\mathbf{x}_t, t, s) = c_{\text{skip}}(t, s)\mathbf{x}_t + c_{\text{out}}(t, s)h(\mathbf{x}_t, t, s),$$

where h is defined as

$$h(\mathbf{x}_t, t, s) = \frac{1}{c_{\text{out}}}\Big[(1 - c_{\text{skip}})\mathbf{x}_t + \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}|\mathbf{x}_u]}{u} \, \mathrm{d}u\Big].$$

Then, we can verify that c_{skip} satisfies the condition in Eq. (12) and that

$$\mathbb{E}[\mathbf{x}|\mathbf{x}_t] = g(\mathbf{x}_t, t, t) + \frac{\sigma_{\min}^2 + \sigma_{\text{data}}^2 - \sigma_{\min}t}{(t - \sigma_{\min})^2 + \sigma_{\text{data}}^2}\mathbf{x}_t.$$

The DSM loss with this c_{skip} becomes

$$\mathcal{L}_{\text{DM}}(\boldsymbol{\theta}) = \mathbb{E}\Big[\Big\|\mathbf{x}_0 - \Big(g_{\boldsymbol{\theta}}(\mathbf{x}_t, t, t) + \frac{\sigma_{\min}^2 + \sigma_{\text{data}}^2 - \sigma_{\min}t}{(t - \sigma_{\min})^2 + \sigma_{\text{data}}^2}\mathbf{x}_t\Big)\Big\|_2^2\Big]$$

However, empirically, we find that the parametrization of $c_{\text{skip}}(t, s)$ and $c_{\text{out}}(t, s)$ other than the ODE solver-oriented one, i.e., $c_{\text{skip}}(t, s) = \frac{s}{t}$ and $c_{\text{skip}}(t, s) = 1 - \frac{s}{t}$, faces training instability. Therefore, we set $G(\mathbf{x}_t, t, s) = \frac{s}{t}\mathbf{x}_t + (1 - \frac{s}{t})g(\mathbf{x}_t, t, s)$ as our default design and estimate g-function with the neural network.

C.2 Characteristics of γ-sampling

Connection with SDE When $G_{\boldsymbol{\theta}} = G$, a single step of γ-sampling is expressed as:

$$\mathbf{x}_{t_{n+1}}^{\gamma} = \mathbf{x}_{t_n} + G(\mathbf{x}_{t_n}, t_n, \sqrt{1-\gamma^2}t_{n+1}) + \gamma t_{n+1}\boldsymbol{\epsilon}$$

$$= \mathbf{x}_{t_n} - \left(\underbrace{\int_{t_n}^{t_{n+1}} u\nabla\log p_u(\mathbf{x}_u)\,\mathrm{d}u}_{\text{past information}} + \underbrace{\int_{t_{n+1}}^{\sqrt{1-\gamma^2}t_{n+1}} u\nabla\log p_u(\mathbf{x}_u)\,\mathrm{d}u}_{\text{future information}} \right) + \gamma t_{n+1}\boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. This formulation cannot be interpreted as a differential form (Øksendal, 2003) because it look-ahead future information (from t_{n+1} to $\sqrt{1-\gamma^2}t_{n+1}$) to generate the sample $\mathbf{x}_{t_{n+1}}^{\gamma}$ at time t_{n+1}. This suggests that there is no Itô's SDE that corresponds to our γ-sampler pathwisely, opening up new possibilities for the development of a new family of diffusion samplers.

Connection with EDM's stochastic sampler We conduct a direct comparison between EDM's stochastic sampler and CTM's γ-sampling. We denote $\texttt{Heun}(\mathbf{x}_t, t, s)$ as Heun's solver initiated at time t and point $\mathbf{x}_t$ and ending at time s. It's worth noting that EDM's sampler inherently experiences discretization errors stemming from the use of Heun's solver, while CTM is immune to such errors.

The primary distinction between EDM's stochastic sampling in Algorithm 2 and CTM's γ-sampling in Algorithm 3 is the order of the forward (diffuse) and backward (denoise) steps. However, through the iterative process of forward-backward time traveling, these two distinct samplers become indistinguishable. Aside from the order of forward-backward steps, the two algorithms essentially align if we opt to synchronize the CTM's time $(t_n^{\text{CTM}}, \tilde{t}_n^{\text{CTM}})$ to with the EDM's time $(\hat{t}_n^{\text{EDM}}, t_{n+1}^{\text{EDM}})$, respectively, and their γs accordingly.

Algorithm 2 EDM's sampler

1: Start from $\mathbf{x}_{t_0} \sim \pi$
2: **for** $n = 0$ to $N - 1$ **do**
3: $\quad \hat{t}_n \leftarrow (1+\gamma)t_n$
4: $\quad$ Diffuse $\mathbf{x}_{\hat{t}_n} \leftarrow \mathbf{x}_{t_n} + \sqrt{\hat{t}_n^2 - t_n^2}\boldsymbol{\epsilon}$
5: $\quad$ Denoise $\mathbf{x}_{t_{n+1}} \leftarrow \texttt{Heun}(\mathbf{x}_{\hat{t}_n}, \hat{t}_n, t_{n+1})$
6: **end for**
7: **Return** $\mathbf{x}_{t_N}$

Algorithm 3 CTM's γ-sampling

1: Start from $\mathbf{x}_{t_0} \sim \pi$
2: **for** $n = 0$ to $N - 1$ **do**
3: $\quad \tilde{t}_{n+1} \leftarrow \sqrt{1-\gamma^2}t_{n+1}$
4: $\quad$ Denoise $\mathbf{x}_{\tilde{t}_{n+1}} \leftarrow G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}, t_n, \tilde{t}_{n+1})$
5: $\quad$ Diffuse $\mathbf{x}_{t_{n+1}} \leftarrow \mathbf{x}_{\hat{t}_{n+1}} + \gamma t_{n+1}\boldsymbol{\epsilon}$
6: **end for**
7: **Return** $\mathbf{x}_{t_N}$

Algorithm 4 Loss-based Trajectory Optimization

1: $\mathbf{x}_{ref}$ is given
2: Diffuse $\mathbf{x}_{t_0} \leftarrow \mathbf{x}_{ref} + t_0\boldsymbol{\epsilon}$
3: **for** $n = 1$ to N **do**
4: $\quad \tilde{t}_n \leftarrow \sqrt{1-\gamma^2}t_n$
5: $\quad$ Denoise $\mathbf{x}_{\tilde{t}_n} \leftarrow G_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n-1}}, t_{n-1}, \tilde{t}_n)$
6: $\quad$ **for** $m = 1$ to M **do**
7: $\quad\quad$ Sample $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}' \sim \mathcal{N}(0, \mathbf{I})$
8: $\quad\quad$ Apply corrector $\mathbf{x}_{\tilde{t}_n} \leftarrow \mathbf{x}_{\tilde{t}_n} + \frac{\zeta}{2}\left(\nabla\log p_{\tilde{t}_n}(\mathbf{x}_{\tilde{t}_n}) - c_{\tilde{t}_n}\nabla_{\mathbf{x}_{\tilde{t}_n}}L(\mathbf{x}_{\tilde{t}_n}, \mathbf{x}_{ref} + \tilde{t}_n\boldsymbol{\epsilon})\right) + \sqrt{\zeta}\boldsymbol{\epsilon}'$
9: $\quad$ **end for**
10: $\quad$ Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
11: $\quad$ Diffuse $\mathbf{x}_{t_n} \leftarrow \mathbf{x}_{\hat{t}_n} + \gamma t_n\boldsymbol{\epsilon}$
12: **end for**

C.3 Trajectory Control with Guidance

We could apply γ-sampling for application tasks, such as image inpainting or colorization, using the (straightforwardly) generalized algorithm suggested in CM. In this section, however, we propose a loss-based trajectory optimization algorithm in Algorithm 4 for potential application downstream tasks.

Table 3: Implementation details.

Hyperparameter	CIFAR-10			ImageNet 64x64
	Unconditional		Conditional	Conditional
	Training with ϕ	Training from Scratch	Training with ϕ	Training with ϕ
Learning rate	0.0004	0.0004	0.0004	0.000008
Discriminator learning rate	0.002	0.002	0.002	0.002
μ	0.9999	0.999	0.999	0.999
N	18	18	18	40
ODE solver	Heun	Self	Heun	Heun
Max. ODE steps	17	17	17	20
EMA decay rate	0.999	0.999	0.999	0.999
Training iterations	100K	100K	100K	30K
Mixed-Precision (FP16)	True	True	True	True
Batch size	256	128	512	2048
Number of GPUs	4	4	4	8

Algorithm 4 uses the time traversal from t_{n-1} to $\tilde{t}_n$, and apply the loss-embedded corrector (Song et al., 2020b) algorithm to explore $\tilde{t}_n$-manifold. For instance, the loss could be a feature loss between $\mathbf{x}_{\tilde{t}_n}$ and $\mathbf{x}_{\text{ref}} + \tilde{t}_n\epsilon$. With this corrector-based guidance, we could control the sample variance. This loss-embedded corrector could also be interpreted as sampling from a posterior distribution. For Figure 8, we choose $N = 2$ with $(t_0, t_1) = \left((\sigma_{\max}^{1/\rho} + (\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho})0.45)^\rho, (\sigma_{\max}^{1/\rho} + (\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho})0.35)^\rho\right)$, $c_{\tilde{t}_n} \equiv 1$, and $M = 10$.

D IMPLEMENTATION DETAILS

D.1 TRAINING DETAILS

Following Karras et al. (2022), 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 ϕ would be close to the teacher denoiser $D_{\phi}(\mathbf{x}_t, t)$. 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 (Sauer et al., 2022). For $\mathcal{L}_{\text{CTM}}$ calculation, we use LPIPS (Zhang et al., 2018) 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 $(\sigma_{\max}^{1/\rho} + \xi(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho}))^\rho$. 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 (Tan & Le, 2019) and DeiT-base (Touvron et al., 2021). 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 (Sauer et al., 2022). 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 $\text{sg}(\boldsymbol{\theta})$ by

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

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 ϕ, we set μ to be 0.999 as default. Throughout the experiments, we use $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, and $\sigma_{\text{data}} = 0.5$.

D.2 EVALUATION DETAILS

For likelihood evaluation, we solve the PF ODE, following the practice suggested in Kim et al. (2022b) with the RK45 (Dormand & Prince, 1980) ODE solver of $\texttt{tol} = 1e - 3$ and $t_{\min} = 0.002$.

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. (2022) statistics. For ImageNet, we compute the metrics following Dhariwal & Nichol (2021) and their pre-calculated statistics. For the StyleGAN-XL ImageNet result, we recalculated the metrics based on the statistics released by Dhariwal & Nichol (2021), 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 $(\sigma_{\max}^{1/\rho} + (\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho})\xi)^\rho$. However, we emphasize the time discretization for both training and sampling is a modeler's choice.

E ADDITIONAL GENERATED SAMPLES

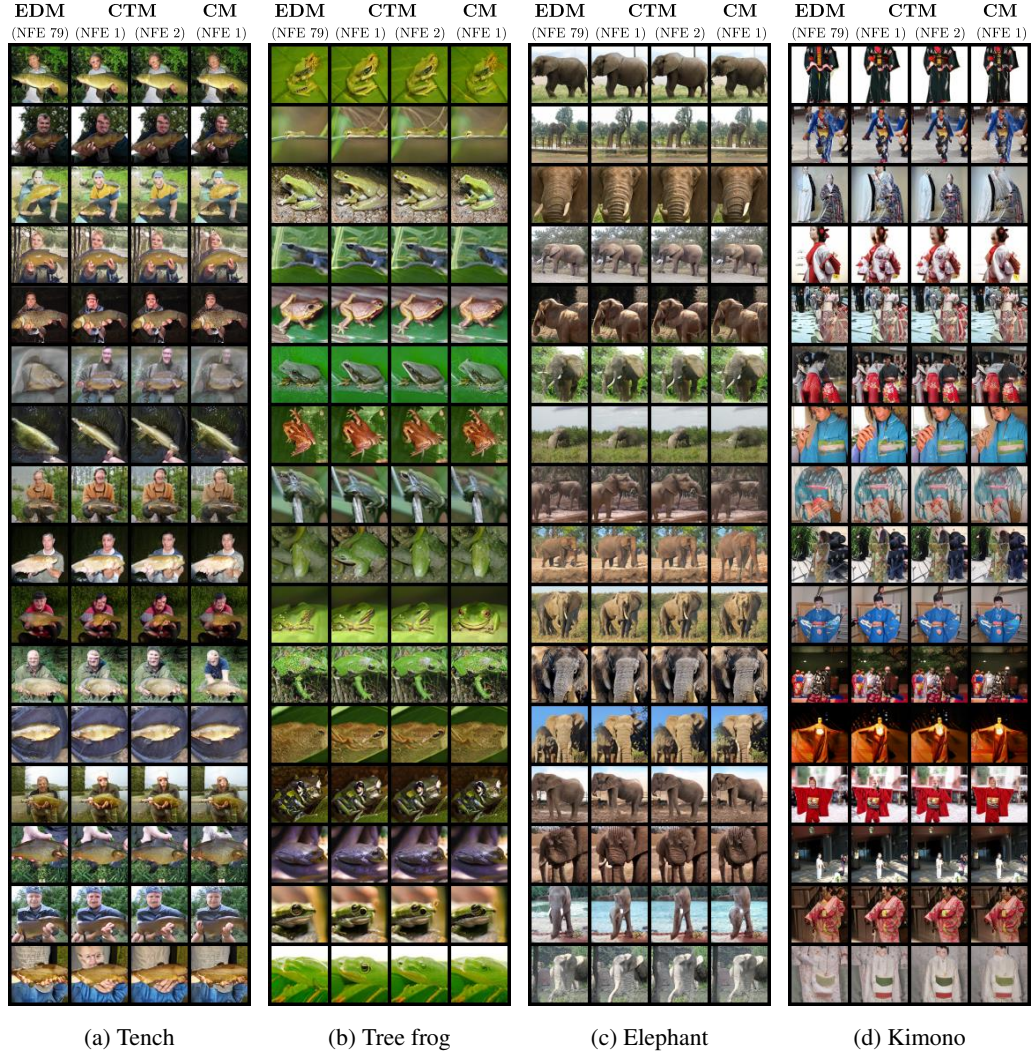(a) Tench (b) Tree frog (c) Elephant (d) Kimono

Figure 16: Uncurated sample comparisons with identical starting points, generated by EDM (FID 2.44) with NFE 79, CTM (FID 2.06) 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).

F   THEORETICAL SUPPORTS AND PROOFS

F.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

$$\lim_{s \to t} g(\mathbf{x}_t, t, s) = \mathbf{x}_t + t \lim_{s \to t} \frac{1}{t - s} \int_t^s \frac{\mathbf{x}_u - \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_u]}{u} \, du$$

$$= \mathbf{x}_t - t \frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]}{t}$$

$$= \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t].$$

∎

F.2   PROOF OF THEOREM 2

Proof of Theorem 2.   Define $\mathcal{T}_{t \to s}$ as the oracle transition mapping from t to s via the diffusion process Eq. (2). Let $\mathcal{T}_{t \to s}^{\boldsymbol{\theta}^*}(\cdot)$ represent the transition mapping from the optimal CTM, and $\mathcal{T}_{t \to s}^{\boldsymbol{\phi}}(\cdot)$ represent the transition mapping from the empirical probability flow ODE. Since all processes start at point T with initial probability distribution p_T and $\mathcal{T}_{t \to s}^{\boldsymbol{\theta}^*}(\cdot) = \mathcal{T}_{t \to s}^{\boldsymbol{\phi}}(\cdot)$, Theorem 2 in (Chen et al., 2022) and $\mathcal{T}_{T \to t} \sharp p_T = p_t$ from Proposition 9 tell us that for $t > s$

$$D_{TV}\left(\mathcal{T}_{t \to s} \sharp p_t, \mathcal{T}_{t \to s}^{\boldsymbol{\theta}^*} \sharp p_t\right) = D_{TV}\left(\mathcal{T}_{t \to s} \sharp p_t, \mathcal{T}_{t \to s}^{\boldsymbol{\phi}} \sharp p_t\right) = \mathcal{O}(t - s). \tag{13}$$

$$D_{TV}\left(\mathcal{T}_{t \to 0} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T, \mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t}^{\boldsymbol{\theta}^*} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t}^{\boldsymbol{\theta}^*} \sharp p_T\right)$$

$$\overset{(a)}{\leq} D_{TV}\left(\mathcal{T}_{t \to 0} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T, \mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T\right)$$

$$+ D_{TV}\left(\mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T, \mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \mathcal{T}_{\sqrt{1-\gamma^2}t \to t}^{\boldsymbol{\theta}^*} \mathcal{T}_{T \to \sqrt{1-\gamma^2}t}^{\boldsymbol{\theta}^*} \sharp p_T\right)$$

$$\overset{(b)}{=} D_{TV}\left(\mathcal{T}_{t \to 0} \mathcal{T}_{T \to t} \sharp p_T, \mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \mathcal{T}_{T \to t} \sharp p_T\right) + D_{TV}\left(\mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T, \mathcal{T}_{T \to \sqrt{1-\gamma^2}t}^{\boldsymbol{\theta}^*} \sharp p_T\right)$$

$$\overset{(c)}{=} D_{TV}\left(\mathcal{T}_{t \to 0} \sharp p_t, \mathcal{T}_{t \to 0}^{\boldsymbol{\theta}^*} \sharp p_t\right) + D_{TV}\left(\mathcal{T}_{T \to \sqrt{1-\gamma^2}t} \sharp p_T, \mathcal{T}_{T \to \sqrt{1-\gamma^2}t}^{\boldsymbol{\theta}^*} \sharp p_T\right)$$

$$\overset{(d)}{=} \mathcal{O}(\sqrt{t}) + \mathcal{O}(\sqrt{T - \sqrt{1-\gamma^2}t}).$$

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 9, and (d) comes from Eq. (13).

∎

F.3   PROOF OF PROPOSITION 3

Proof of Proposition 3.   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 $\mathcal{L}_{\text{CTM}}^N(\boldsymbol{\theta}; \boldsymbol{\phi}) = 0$, it implies that for any $\mathbf{x}_{t_n}$, $n \in [\![1, N]\!]$, and $m \in [\![1, n]\!]$

$$\mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_m), t_m, 0)\big) = \mathcal{F}\big(G_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}(\mathbf{x}_{t_n}^{\boldsymbol{\phi}}, t_n, t_m), t_m, 0)\big) \tag{14}$$

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

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

Therefore,

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

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

So we can obtain via induction that

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

$\blacksquare$

Indeed, an analogue of Proposition 3 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 $q \geq 1$. We can similarly derive

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

F.4 PROOF OF PROPOSITION 5

Proof of Proposition 5. We first prove that for any $t \in [0, T]$ and $s \leq t$, as $N \to \infty$,

$$\sup_{\mathbf{x} \in \mathbb{R}^D} \|G_{\boldsymbol{\theta}_N}(G_{\boldsymbol{\theta}_N}(\mathbf{x}, t, s), s, 0), G_{\boldsymbol{\theta}_N}(G(\mathbf{x}, t, s; \phi), s, 0)\|_2 \to 0. \tag{15}$$

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 $\Delta_N t \to \infty$.

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

Since both G and $G_{\boldsymbol{\theta}_N}$ are uniform continuous on $\mathbb{R}^D \times [0, T] \times [0, T]$, together with Proposition 3, we obtain Eq. (15) as $\Delta_N t \to \infty$.

In particular, Eq. (15) implies that when $N \to \infty$

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

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

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 $p_{\boldsymbol{\phi}}(\cdot)$. 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 (Boos, 1985; Sweeting, 1986) 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 $p_{\text{data}}(\cdot)$. ∎

F.5 PROOF OF PROPOSITION 6

Lemma 10. *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

$$\mathbf{x}'(\tau) = f(\mathbf{x}(\tau), \tau) \quad on \ [0, T]. \tag{16}$$

Fix a $t \in [0, T]$, the solution operator $\mathcal{T}$ of Eq. (16) with an initial condition $\mathbf{x}_t$ is defined as

$$\mathcal{T}[\mathbf{x}_t](s) := \mathbf{x}_t + \int_t^s f(\mathbf{x}(\tau; \mathbf{x}_t), \tau) \, \mathrm{d}\tau, \quad s \in [t, T]. \tag{17}$$

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 $\mathbf{x}_t, \hat{\mathbf{x}}_t \in \mathbb{R}^D$

$$e^{-L(s-t)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \le \|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 \le e^{L(t-s)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2. \tag{18}$$

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 10. Assumptions (a) and (b) ensure the solution operator in Eq. (17) is well-defined by applying Carathéodory-type global existence theorem (Reid, 1971). 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

$$\begin{cases} \mathbf{y}'(s) = -f(\mathbf{y}(s), s), & s \in [t, s_0] \\ \mathbf{y}(t) = \mathcal{T}[\mathbf{x}_t](s_0) = \mathcal{T}[\hat{\mathbf{x}}_t](s_0) \end{cases} \tag{19}$$

Thus, the uniqueness theorem of solution to Eq. (19) 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 $\mathbf{x}_t, \hat{\mathbf{x}}_t \in \mathbb{R}^D$,

$$\|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 = \|\mathbf{x}(s; \mathbf{x}_t) - \hat{\mathbf{x}}(s; \hat{\mathbf{x}}_t)\|_2$$

$$\le \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \int_t^s \|f(\mathbf{x}(\tau; \mathbf{x}_t), \tau) - f(\hat{\mathbf{x}}(\tau; \hat{\mathbf{x}}_t), \tau)\|_2 \, \mathrm{d}\tau$$

$$\le \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + L \int_t^s \|\mathbf{x}(\tau; \mathbf{x}_t) - \hat{\mathbf{x}}(\tau; \hat{\mathbf{x}}_t)\|_2 \, \mathrm{d}\tau.$$

By applying Gröwnwall's lemma, we obtain

$$\|\mathcal{T}[\mathbf{x}_t](s) - \mathcal{T}[\hat{\mathbf{x}}_t](s)\|_2 = \|\mathbf{x}(s; \mathbf{x}_t) - \hat{\mathbf{x}}(s; \hat{\mathbf{x}}_t)\|_2 \le e^{L(s-t)} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2. \tag{20}$$

On the other hand, consider the reverse time ODE of Eq. (16) 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

$$\mathbf{y}'(u) = h(\mathbf{y}(u), u), \quad u \in [t, s]. \tag{21}$$

Similarly, we define the solution operator to Eq. (21) as

$$\mathcal{S}[\mathbf{y}_t](s) := \mathbf{y}_t + \int_t^s h(\mathbf{y}(u; \mathbf{y}_t), u) \, \mathrm{d}u. \tag{22}$$

Here $\mathbf{y}_t$ denotes the initial value of Eq. (21) 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

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

By applying Gröwnwall's lemma, we obtain

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

Therefore,

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

$\blacksquare$

Proof of Proposition 6. 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 10 to the integral form of $G(\mathbf{x}_t, t, s; \boldsymbol{\phi})$.

$\blacksquare$

F.6 PROOF OF PROPOSITION 7

Lemma 11. *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 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$*

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

Then

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

Proof of Lemma 11.   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 $\text{cov}(h(X), h(Y)) = 0$.

$$2\text{Var}\left(h(X)\right) = \text{Var}\left(h(X) - h(Y)\right)$$

$$= \mathbb{E}\left[(h(X) - h(Y))^2\right] - (\mathbb{E}\left[h(X) - h(Y)\right])^2. \tag{23}$$

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

$$2\text{Var}\left(h(X)\right) = \mathbb{E}\left[(h(X) - h(Y))^2\right] \tag{24}$$

$$\leq L^2 \mathbb{E}\left[(X - Y)^2\right]$$

$$= 2L^2 \text{Var}\left(X\right).$$

The final equality follows the same reasoning as in Eq. (23). Likewise, we can apply the argument from Eq. (24) to show that

$$2\text{Var}\left(h(X)\right) = \mathbb{E}\left[(h(X) - h(Y))^2\right]$$

$$\geq L^{-2} \mathbb{E}\left[(X - Y)^2\right]$$

$$= 2L^{-2} \text{Var}\left(X\right).$$

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 7.   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,

$$\text{Var}\left(X_{n+1}\right) = \text{Var}\left(G_{\boldsymbol{\theta}^*}(X_n, t_n, \sqrt{1 - \gamma^2}t_{n+1})\right) + \text{Var}\left(Z_{n+1}\right)$$

$$= \text{Var}\left(G_{\boldsymbol{\theta}^*}(X_n, t_n, \sqrt{1 - \gamma^2}t_{n+1})\right) + \gamma^2\sigma^2(t_{n+1}). \tag{25}$$

Proposition 6 implies that $G_{\boldsymbol{\theta}^*}(\cdot, t_n, \sqrt{1 - \gamma^2}t_{n+1})$ is bi-Lipschitz and that for any $\mathbf{x}, \mathbf{y}$

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

$$\leq \zeta(t_n, t_{n+1}, \gamma) \left\|\mathbf{x} - \mathbf{y}\right\|_2, \tag{26}$$

where $\zeta(t_n, t_{n+1}, \gamma) = \exp\left(2L_{\boldsymbol{\phi}}(t_n - \sqrt{1 - \gamma^2}t_{n+1})\right)$. Proposition 7 follows immediately from the inequalities (25) and (26). ∎

F.7   Proof of Proposition 9

Proof of Proposition 9. $\{p_t\}_{t=0}^T$ is known to satisfy the Fokker-Planck equation (Øksendal, 2003) (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 (Song et al., 2020b))

$$\frac{\partial p_t}{\partial t} = -\text{div}\left(\mathbf{W}_t p_t\right), \quad \text{in } (0, T) \times \mathbb{R}^D \tag{27}$$

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

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

$$\frac{\partial \mu_t}{\partial t} = -\text{div}\left(\mathbf{W}_t \mu_t\right) \quad \text{in } (0, T) \times \mathbb{R}^D. \tag{28}$$

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 $\mathbf{W}_t := -t\nabla \log p_t : [0, T] \times \mathbb{R}^D \to \mathbb{R}^D$ 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 $\text{Lip}(\mathbf{W}_t, K)$ denotes the Lipschitz constant of $\mathbf{W}_t$ on K.

Thus, Proposition 8.1.8 of (Ambrosio et al., 2005) implies that for p_T-a.e. $\mathbf{x}$, the following reverse time ODE (which is the Eq. (2)) admits a unique solution on $[0, T]$

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t} X_t(\mathbf{x}) = \mathbf{W}_t \left(X_t(\hat{\mathbf{x}}) \right) \\ X_T(\hat{\mathbf{x}}) = \mathbf{x}. \end{cases} \tag{29}$$

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 (Ambrosio et al., 2005)) and the uniqueness of Eq. (29), 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 $s \in [0, t]$.

$\blacksquare$