

Target-Driven Distillation: Consistency Distillation with Target Timestep Selection and Decoupled Guidance

Cunzheng Wang^{1,2*}, Ziyuan Guo², Yuxuan Duan^{2,3}, Huaxia Li², Nemo Chen², Xu Tang², Yao Hu²

¹School of Software Technology, Zhejiang University

²Xiaohongshu

³Shanghai Jiao Tong University

22251351@zju.edu.cn, guoziyuan@xiaohongshu.com, sjtudyx2016@sjtu.edu.cn,
{lihuaxia,linghu,tangshen,xiahou}@xiaohongshu.com

Abstract

Consistency distillation methods have demonstrated significant success in accelerating the generative tasks of diffusion models. However, since previous consistency distillation methods use simple and straightforward strategies in selecting target timesteps, they usually struggle with blurs and detail losses in generated images. To address these limitations, we introduce Target-Driven Distillation (TDD), which (1) adopts a delicate selection strategy of target timesteps, increasing the training efficiency; (2) utilizes decoupled guidances during training, making TDD open to post-tuning on guidance scale during inference periods; (3) can be optionally equipped with non-equidistant sampling and \mathbf{x}_0 clipping, enabling a more flexible and accurate way for image sampling. Experiments verify that TDD achieves state-of-the-art performance in few-step generation, offering a better choice among consistency distillation models.

1 Introduction

Diffusion models (Sohl-Dickstein et al. 2015; Song and Ermon 2019) have demonstrated exceptional performance in image generation, producing diverse and high-quality images. Unlike previous models like GANs (Goodfellow et al. 2014; Karras, Laine, and Aila 2019) or VAEs (Kingma and Welling 2013; Sohn, Lee, and Yan 2015), diffusion models are good at modeling complex image distributions and conditioning on non-label conditions such as free-form text prompts. However, since diffusion models adopt iterative denoising processes, they usually take substantial time to generate images. To address such a challenge, consistency distillation methods (Song et al. 2023; Luo et al. 2023a,b; Kim et al. 2023; Wang et al. 2024) have been proposed as effective strategies to accelerate generation while maintaining image quality. These methods distill pre-trained diffusion models following the *self-consistency property* i.e. the predicted results from any two neighboring timesteps towards the same target timestep are regularized to be the same. Depending on the choice of target timesteps, recent consistency distillation methods can be categorized as *single-target distillation* and *multi-target distillation*, illustrated in Figure 1.

*Corresponding author.

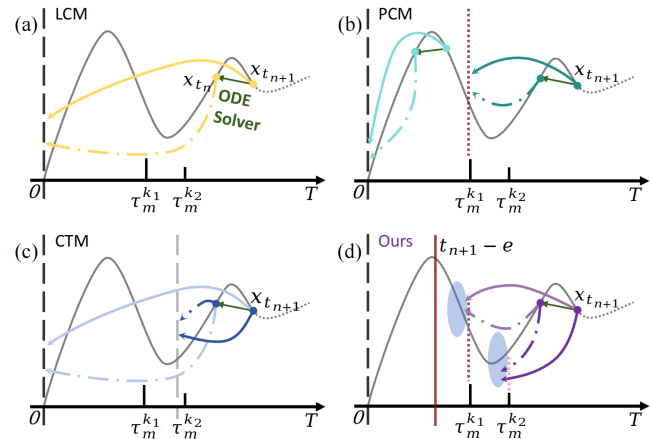


Figure 1: Comparison of different distillation methods. $\tau_m^{k_1}$ and $\tau_m^{k_2}$ represent a target timestep when divided into k_1 and k_2 , respectively. LCM (a) and PCM (b) are examples of single-target distillation, where a certain \mathbf{x}_{t_n} corresponds to only one target timestep. In contrast, CTM (c) and ours (d) are multi-target distillation methods, where a certain \mathbf{x}_{t_n} can correspond to multiple target timesteps.

Single-target distillation methods follow a one-to-one mapping when choosing target timesteps, that is, they always choose *the same* target timestep each time they reach a certain timestep along the trajectory of PF-ODE (Song et al. 2020). One straightforward choice is to map any timestep to the final timestep at 0 (Song et al. 2023; Luo et al. 2023a). However, these methods usually suffer from the accumulated error of long-distance predictions. Another choice is evenly partitioning the full trajectory into several sub-trajectories and to map a timestep to the initial of the sub-trajectory it belongs to (Wang et al. 2024). Although the error can be reduced by shortening the predicting distances when training, the image quality will be suboptimal when adopting a schedule with a different number of sub-trajectories during inference periods.

On the other hand, multi-target distillation methods follow a one-to-multiple mapping, that is, *possibly different* target timesteps may be chosen each time they come to a certain timestep. A typical choice is to map the current timestep

to a random target timestep ahead (Kim et al. 2023; Zheng et al. 2024). Theoretically, these methods are trained to predict from any to any timestep, thus may generally achieve good performance under different schedules. Yet, practically most of these predictions are redundant since we will never go through them under common denoising schedules. Hence, multi-target distillation methods usually require a large time budget to train.

To mitigate the aforementioned issues, we propose **Target-Driven Distillation (TDD)**, a multi-target approach that emphasizes delicately selected target timesteps during distillation processes. Our method involves three key designs: **Firstly**, for any timestep, it selects a nearby timestep ahead that belongs to one of a predefined set of few-step equidistant denoising schedules (e.g. 4–8 steps), which eliminates long-distance predictions while only focusing on the timesteps we will probably pass through during inference periods under different schedules. Also, TDD incorporates a stochastic offset that further pushes the selected timestep ahead towards the final target timestep, in order to accommodate nondeterministic sampling such as γ -sampling (Kim et al. 2023). **Secondly**, while distilling classifier-free guidance (CFG) (Ho and Salimans 2022) into the distilled models, to align with the standard training process using CFG, TDD additionally replaces a portion of the text conditions with unconditional (*i.e.* empty) prompts. With such a design, TDD is open to a proposed inference-time tuning technique on a guidance scale, allowing user-specified balances between the accuracy and the richness of image contents conditioned on text prompts. **Finally**, TDD is optionally equipped with a non-equidistant sampling method doing short-distance predictions at initial steps and long-distance ones at later steps, which helps to improve image quality. Additionally, TDD adopts \mathbf{x}_0 clipping to prevent out-of-bound predictions and address the overexposure issue.

Our contributions are summarized as follows:

- We provide a taxonomy on consistency distillation models, classifying previous works as *single-target* and *multi-target* distillation methods.
- We propose Target-Driven Distillation, which highlights target timestep selection and decoupled guidance during distillation processes.
- We present extensive experiments to validate the effectiveness of our proposed distillation method.

2 Related Work

Diffusion models (Sohl-Dickstein et al. 2015; Song and Ermon 2019; Karras et al. 2022) have demonstrated significant advantages in high-quality image synthesis (Ramesh et al. 2022; Rombach et al. 2022; Dhariwal and Nichol 2021), image editing (Meng et al. 2021; Saharia et al. 2022a; Balaji et al. 2022), and specialized tasks such as layout generation (Zheng et al. 2023; Wu et al. 2024a) and customized video generation (Wu et al. 2024b). However, their multi-step iterative process incurs significant computational costs, hindering real-time applications. Beyond developing faster samplers (Song, Meng, and Ermon 2020; Lu et al. 2022a,b; Zhang and Chen 2022), there is growing interest in model

distillation approaches (Sauer et al. 2023; Liu, Gong, and Liu 2022; Sauer et al. 2024; Yin et al. 2024). Among them, consistency distillation models have been proven to be particularly effective in accelerating generation processes while preserving output similarity between the original and distilled models.

Song et al. introduced the concept of consistency models, emphasizing the importance of achieving self-consistency across arbitrary pairs of points on the same probability flow ordinary differential equation (PF-ODE) trajectory (Song et al. 2020). This approach is particularly effective when distilled from a teacher model or when incorporating modules like LCM-LoRA (Luo et al. 2023b), which can achieve few-step generation with minimal retraining resources.

However, a key limitation of these models is increasing learning difficulty when mapping points further from timestep 0, leading to suboptimal performance when mapping from pure noise in a single step. Phased Consistency Models (PCM) (Wang et al. 2024) address this issue by dividing the ODE trajectory into multiple sub-trajectories, reducing learning difficulty by mapping each point within a sub-trajectory to its initial point. However, in these methods, each point is mapped to a unique target timestep during distillation, resulting in suboptimal inference when predicting another target timestep from this point.

Recent advancements, such as Consistency Trajectory Models (CTM) (Kim et al. 2023) and Trajectory Consistency Distillation (TCD) (Zheng et al. 2024), aim to overcome this issue by enabling consistency models to perform anytime-to-anytime jumps, allowing all points between timestep 0 and the current timestep to be used as possible target timesteps. However, the inclusion of numerous unused target timesteps reduces training efficiency and makes the model less sensitive to fewer-step denoising timesteps.

3 Method

In this section, we will first deliver some preliminaries in section 3.1, followed by detailed descriptions of our proposed Target-Driven Distillation in sections 3.2 to 3.4.

3.1 Preliminaries

Diffusion Model Diffusion models constitute a category of generative models that draw inspiration from thermodynamics and stochastic processes, encompassing both a forward process and a reverse process. The forward process is modeled as a stochastic differential equation (SDE) (Song et al. 2020; Karras et al. 2022). Let $p_{\text{data}}(\mathbf{x})$ denotes the data distribution and $p_t(\mathbf{x})$ the distribution of \mathbf{x} at time t . For a given set $\{\mathbf{x}_t | t \in [0, T]\}$, the stochastic trajectory is described by:

$$d\mathbf{x}_t = f(\mathbf{x}_t, t) dt + g(t) d\mathbf{w}_t, \quad (1)$$

where \mathbf{w}_t represents standard Brownian motion, $f(\mathbf{x}_t, t)$ is the drift coefficient for deterministic changes, and $g(t)$ is the diffusion coefficient for stochastic variations. At $t = 0$, we have $p_0(\mathbf{x}) \equiv p_{\text{data}}(\mathbf{x})$.

Any diffusion process described by an SDE can be represented by a deterministic process described by an ODE that



Figure 2: Visual comparison among different methods. In addition, we have released a detailed comparison between our method and TCD. Our method demonstrates advantages in both image complexity and clarity.

shares identical marginal distributions, referred to as a Probability Flow ODE (PF-ODE). The PF-ODE is formulated as:

$$dx = \left[f(x, t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t(x) \right] dt, \quad (2)$$

where $\nabla_x \log p_t(x)$ represents the gradient of the log-density of the data distribution $p_t(x)$, known as the score function. Empirically, we approximate this score function with a score model $s_\phi(x, t)$ trained via score matching techniques. Although there are numerous methods (Song, Meng, and Ermon 2020; Lu et al. 2022a,b; Karras et al. 2022) available to solve ODE trajectories, they still necessitate a large number of sampling steps to attain high-quality generation results.

Consistency Distillation To render a unified representation across all consistency distillation methods, we define the teacher model as ϕ , the consistency function with the student model as f_θ , the conditional prompt as c , and the ODE solver $\Phi(\dots; \phi)$ predicting from a certain timestep t_{n+1} to its previous timestep t_n following an equidistant schedule from T to 0. With a certain point $x_{t_{n+1}}$ on the trajectory at timestep t_{n+1} , and its previous point $\hat{x}_{t_n}^\phi$ predicted by $\Phi(\dots; \phi)$, the core consistency loss can be formulated as

$$\mathcal{L}_{\text{CMs}} := \left\| f_\theta(x_{t_{n+1}}, t_{n+1}, \tau) - f_{\theta^-}(\hat{x}_{t_n}^\phi, t_n, \tau) \right\|_2^2, \quad (3)$$

where f_{θ^-} is the consistency function with a target model updated with the exponential moving average (EMA) from the student model, and τ refers to the *target timestep*.

Among the mainstream distillation methods, the choices of τ are the most critical differences (see Figure 1). Single-target distillation methods select the same τ each time when predicting from a certain t_{n+1} . For example, CM (Song et al. 2023) sets $\tau = 0$ for any timestep t_{n+1} , while PCM (Wang et al. 2024) segments the full trajectory into \mathcal{K} (e.g. 4) phased sub-trajectories, and chooses the next ending point:

$$\tau = \max \left\{ \tau \in \left\{ 0, \frac{T}{\mathcal{K}}, \frac{2T}{\mathcal{K}}, \dots, \frac{(\mathcal{K}-1)T}{\mathcal{K}} \right\} \mid \tau < t_n \right\}. \quad (4)$$

On the other hand, multi-target distillation methods may select different values for τ each time predicting from t_{n+1} . For instance, CTM (Kim et al. 2023) selects a random τ within the interval $[0, t_n)$.

Our TDD, different from previous approaches that rely on simple trajectory segmentation or selecting from all possible τ , employs a strategic selection of τ , detailed in section 3.2. According to our taxonomy, TDD is a multi-target distillation method, but we aim to minimize training on redundant predictions unnecessary for inference.

3.2 Target Timestep Selection

First, TDD pre-determines a set of equidistant denoising schedules, whose numbers of denoising steps range from \mathcal{K}_{\min} to \mathcal{K}_{\max} , that we may adopt during inference periods. In the full trajectory of a PF-ODE from T to 0, for each $k \in [\mathcal{K}_{\min}, \mathcal{K}_{\max}]$, the corresponding schedule include timesteps $\{\tau_m^k\}_{m=0}^{k-1}$ where $\tau_m^k = \frac{mT}{k}$. Then, we can define

the union of all the timesteps of these schedules as

$$\mathcal{T} = \bigcup_{k=\mathcal{K}_{\min}}^{\mathcal{K}_{\max}} \{\tau_m^k\}_{m=0}^{k-1}, \quad (5)$$

which includes all the possible timesteps that we may choose as target timesteps. Note that Equation (5) is a generalized formulation, where $\mathcal{K}_{\min} = \mathcal{K}_{\max} = 1$ for CM, $1 < \mathcal{K}_{\min} = \mathcal{K}_{\max} < N$ for PCM, and $\mathcal{K}_{\min} = \mathcal{K}_{\max} = N$ for CTM where N is the total number of predictions within the equidistant schedule used by the ODE solver Φ . As for TDD, we cover commonly used few-step denoising schedules, with typical values for $\mathcal{K}_{\min} = 4$ and $\mathcal{K}_{\max} = 8$.

Based on the condition, we establish the consistency function as:

$$\mathbf{f} : (\mathbf{x}_t, t, \tau) \mapsto \mathbf{x}_\tau, \quad (6)$$

where $t \in [0, T]$ and $\tau \in \mathcal{T}$, and we expect that the predicted results to the specific target timestep τ will be consistent.

Training Although \mathcal{T} is a selected set of timesteps, predicting to an arbitrary timestep within \mathcal{T} introduces redundancy, as the model does not need to learn long-distance predictions from a large t to a small τ . Therefore, we introduce an additional constraint $e = \frac{T}{\mathcal{K}_{\min}}$. This constraint further narrows possible choices at timestep t , reducing the learning difficulty. Formally, we uniformly select

$$\tau_m \sim \mathcal{U}(\{\tau \in \mathcal{T} | t - e \leq \tau \leq t\}). \quad (7)$$

Besides, γ -sampling (Kim et al. 2023) is commonly used in few-step generation to introduce randomness and stabilize outputs. To accommodate this design, we introduce an additional hyperparameter $\eta \in [0, 1]$. The final consistency target timesteps are selected by

$$\tilde{\tau}_m \sim \mathcal{U}([(1 - \eta)\tau_m, \tau_m]). \quad (8)$$

Define the solution using the teacher model T_ϕ from $\mathbf{x}_{t_{n+1}}$ to \mathbf{x}_{t_n} with PF-ODE solver as:

$$\hat{\mathbf{x}}_{t_n}^\phi = \Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_n; T_\phi), \quad (9)$$

where $\Phi(\cdot \cdot \cdot; T_\phi)$ is update function and $\hat{\mathbf{x}}_{t_n}^\phi$ is an accurate estimate of \mathbf{x}_{t_n} from $\mathbf{x}_{t_{n+1}}$. The loss function of TDD can be defined as:

$$\mathcal{L}_{\text{TDD}}(\theta, \theta^-; \phi) := E[\sigma(t_n, \tilde{\tau}_m) \left\| \mathbf{f}_\theta(\mathbf{x}_{t_{n+1}}, t_{n+1}, \tilde{\tau}_m) - \mathbf{f}_{\theta^-}(\hat{\mathbf{x}}_{t_n}^\phi, t_n, \tilde{\tau}_m) \right\|_2^2], \quad (10)$$

where the expectation is over $\mathbf{x} \sim p_{\text{data}}$, $n \sim \mathcal{U}[1, N - 1]$, $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 I)$ and $\hat{\mathbf{x}}_{t_n}^\phi$ is defined by Equation (9). Namely, $\mathcal{U}[1, N - 1]$ denotes a uniform distribution over 1 to $N - 1$. $\sigma(\cdot, \cdot)$ is a positive weighting function, following CM, we set $\sigma(t_n, \tilde{\tau}_m) \equiv 1$. For a detailed description of our algorithm, please refer to Algorithm 1.

3.3 Decoupled Guidance

Distillation with Decoupled Guidance Classifier-Free Guidance (CFG) allows a model to precisely control the

Algorithm 1: Target-Driven Distillation

Input: dataset \mathcal{D} , learning rate δ , the update function of ODE solver $\Phi(\cdot \cdot \cdot; \cdot)$, EMA rate μ , noise schedule α_t, σ_t , number of ODE steps N , fixed guidance scale ω' , empty prompt ratio ρ , $\mathcal{K}_{\min}, \mathcal{K}_{\max}$.

Parameter: initial model parameter θ

Output:

- 1: $\mathcal{T} \leftarrow \emptyset$
 - 2: **for** $k \in \{\mathcal{K}_{\min}, \mathcal{K}_{\min} + 1, \dots, \mathcal{K}_{\max}\}$ **do**
 - 3: Set and add time steps $\tau_m^k \in \{\tau_0^k, \tau_1^k, \dots, \tau_{k-1}^k\}$ to \mathcal{T}
 - 4: **end for**
 - 5: let $e = \frac{T}{\mathcal{K}_{\min}}$
 - 6: **repeat**
 - 7: Sample $(z, c) \sim \mathcal{D}, \tau_m \sim \mathcal{U}(\{\tau \in \mathcal{T} | t - e \leq \tau \leq t\})$
 - 8: Sample $n \sim \mathcal{U}[1, N - 1], \tilde{\tau}_m \sim \mathcal{U}([(1 - \eta)\tau_m, \tau_m])$
 - 9: Sample $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\alpha_{t_{n+1}} \mathbf{x}, \sigma_{t_{n+1}}^2 \mathbf{I})$
 - 10: **if** probability $> \rho$ **then**
 - 11: $\hat{\mathbf{x}}_{t_n}^{\phi, w'} \leftarrow (1 + \omega') \Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_n; T_\phi)$
 $- \omega' \Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_n; T_\phi)$
 - 12: **else**
 - 13: $\hat{\mathbf{x}}_{t_n}^{\phi, w'} \leftarrow \Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}, t_n; T_\phi)$
 - 14: **end if**
 - 15: $\mathcal{L}_{\text{TDD}}^{w'} := \left\| \mathbf{f}_\theta(\mathbf{x}_{t_{n+1}}, t_{n+1}, \tilde{\tau}_m) - \mathbf{f}_{\theta^-}(\hat{\mathbf{x}}_{t_n}^{\phi, w'}, t_n, \tilde{\tau}_m) \right\|_2^2$
 - 16: $\theta \leftarrow \theta - \delta \nabla_\theta \mathcal{L}(\theta, \theta^-; \phi)$
 - 17: $\theta^- \leftarrow \text{sg}(\mu \theta^- + (1 - \mu) \theta)$
 - 18: **until** convergence
-

generation results without relying on an external classifier during the generation process, effectively modulating the influence of conditional signals. In current consistency distillation methods, to ensure the stability of the training process, it is common to use the sample $\hat{\mathbf{x}}_{t_n}^{\phi, w'}$ generated by the teacher model with classifier-free guidance as a reference in the optimization process for the student model's generated samples. We believe that w' solely represents the diversity constraint in the distillation process, controlling the complexity and generalization of the learning target. This allows for faster learning with fewer parameters. Therefore, w' should be treated separately from the CFG scale w used during inference in consistency models. Therefore, regardless of whether $w' > 0$, following Ho and Salimans, it is essential to include both unconditional and conditional training samples in the training process. Based on this, we replace a portion of the condition with an empty prompt without applying CFG enhancement. For conditions that are not empty, the loss function of TDD can be updated as follows:

$$\mathcal{L}_{\text{TDD}}^{w'} := \left\| \mathbf{f}_\theta(\mathbf{x}_{t_{n+1}}, t_{n+1}, \tilde{\tau}_m) - \mathbf{f}_{\theta^-}(\hat{\mathbf{x}}_{t_n}^{\phi, w'}, t_n, \tilde{\tau}_m) \right\|_2^2. \quad (11)$$

Guidance Scale Tuning Define $\epsilon_\theta(\mathbf{x}_t)$ as a model distilled with decoupled guidances, at each inference step, the noise predicted by the model can be expressed as:

$$\hat{\epsilon}_w = (1 + w) \epsilon_\theta(\mathbf{x}_t^{w'}, t, c) - w \epsilon_\theta(\mathbf{x}_t^{w'}, t), \quad (12)$$

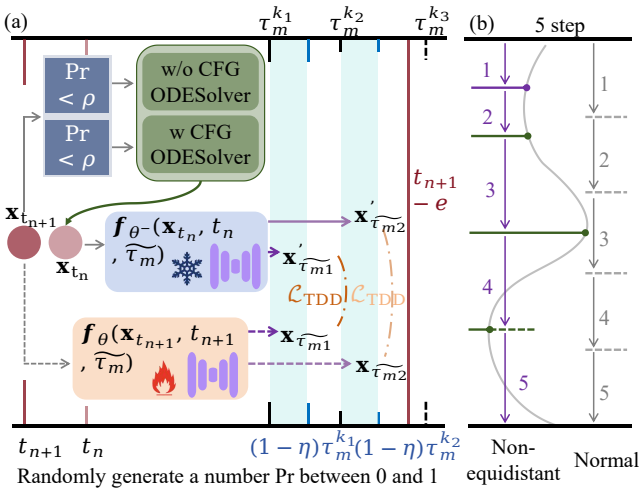


Figure 3: Illustration of TDD training and sampling processes. Fig (a) shows the distillation process, where τ^k represents equidistant timestep within segments. Fig (b) compares non-equidistant sampling with standard sampling for 5-step inference.

where $\mathbf{x}_t^{w'}$ is the input state of the consistency model at time step t distilled with the diversity guidance scale w' .

Although setting a high value for w' (e.g., $w' > 7$) can enhance certain aspects of the generated images, it simultaneously results in significantly reduced image complexity and excessively high contrast.

Is there a way to address this issue without retraining, allowing us to reacquire the ability to sample with a small scale of CFG (Ho and Salimans 2022) similar to the original model? By incorporating the unconditional into the training, we can get $\epsilon_\theta(\mathbf{x}_t^{w'}, t, c) \propto (1 + w')\epsilon_\phi(\mathbf{x}_t, t, c) - w'\epsilon_\phi(\mathbf{x}_t, t)$ and $\epsilon_\theta(\mathbf{x}_t^{w'}, t) \propto \epsilon_\phi(\mathbf{x}_t, t)$, where $\epsilon_\phi(\cdot)$ is the master model.

Denote $(1 + w)\epsilon_\phi(\mathbf{x}_t, t, c) - w\epsilon_\phi(\mathbf{x}_t, t)$ as ϵ_w , representing the noise inferred by the teacher model when using the normal scale w at each time step. We finally obtain:

$$\epsilon_w \approx [\hat{\epsilon}_w + w'\epsilon_\theta(\mathbf{x}_t, t)]/(1 + w'). \quad (13)$$

This equation suggests that regulating $\hat{\epsilon}_w$ with the normal w can approximate the output of the original model ϵ_w . For a more detailed derivation, please refer to the appendix.

In the aforementioned formula, the distillation diversity constraint w' is known and fixed. The parameter w can be inferred based on the standard teacher model's scale. Furthermore, for the current consistency model, even though it has not yet learned the unguided path, this formula can still be approximately utilized for inference, as no other learning has been conducted. This can be expressed as follows:

$$\epsilon'_w \approx [\hat{\epsilon}'_w + \bar{w}'\epsilon'_\theta(\mathbf{x}_t, t)]/(1 + \bar{w}'). \quad (14)$$

where $\epsilon'_\theta(\cdot)$ is current consistency model and $\bar{w}' = (w'_{\min} + w'_{\max})/2$.

3.4 Sample

Since we have distilled the model on multiple equidistant schedules (section 3.2), we can extend the conventionally

equidistant inference strategy to a non-equidistant one. By reducing the sampling interval during high-noise periods, we can mitigate the generation difficulty and achieve better synthesis results. As shown in Figure 3 (b), we ensure that the inference process passes through the target timesteps corresponding to \mathcal{K}_{\min} . As the inference steps increase, additional target timesteps are gradually inserted between these key timesteps. For example, with $\mathcal{K}_{\min} = 4$, as the number of steps increases, we insert 8-step (\mathcal{K}_{\max}) target timesteps within each adjacent 4-step target interval to enhance generation quality.

In addition, the γ -sampler proposed in CTM (Kim et al. 2023) alternates forward and backward jumps along the solution trajectory to solve \mathbf{x}_0 , allowing control over the randomness ratio through γ , which can enhance generation quality to some extent. Predicting \mathbf{x}_s from \mathbf{x}_t can be represented as follows:

$$\mathbf{x}_s = \frac{\alpha_s}{\alpha_t} \mathbf{x}_t - \sigma_s (e^{h_s} - 1) \epsilon_\theta(\mathbf{x}_t, t), \quad (15)$$

where $h_s = \lambda_s - \lambda_t$ and λ represents the log signal-to-noise ratio, $\lambda = \log(\alpha/\sigma)$. However, in few-step sampling with a high CFG scale, the noise distribution after the first step significantly deviates from the expected distribution. To address this issue, we adopt an \mathbf{x}_0 formulation. To approximate $x_\theta(x_t, t) \approx x_0 = (\mathbf{x}_t - \sigma_t \epsilon)/(\alpha_t)$, we can derive

$$\mathbf{x}_s = \frac{\sigma_s}{\sigma_t} \mathbf{x}_t - \alpha_s (e^{-h_s} - 1) x_\theta(\mathbf{x}_t, t). \quad (16)$$

Following prior works (Saharia et al. 2022b; Lu et al. 2022b), we apply the clipping method \mathcal{C} , which clips each latent variable to the specific percentile of its absolute value and normalizes it to prevent saturation of the latent variables. Let $\hat{\mathbf{x}}_0 = \mathcal{C}(x_\theta(\mathbf{x}_t, t))$, we ultimately obtain:

$$\hat{\mathbf{x}}_s = \frac{\sigma_s}{\sigma_t} \mathbf{x}_t - \alpha_s (e^{-h_s} - 1) \hat{\mathbf{x}}_0. \quad (17)$$

When using γ -sampling, the transition from timestep t to the next timestep p can be expressed as

$$\mathbf{x}_p = \frac{\alpha_p}{\alpha_s} \hat{\mathbf{x}}_s + \sqrt{1 - \frac{\alpha_p^2}{\alpha_s^2}} \mathbf{z}, \quad \mathbf{z} \in \mathcal{N}(0, I), \quad (18)$$

where $s = (1 - \gamma)p$.

4 Experiments

4.1 Dataset

We use a subset of the Laion-5B (Schuhmann et al. 2022) dataset for training. All images in the dataset have an aesthetic score above 5.5, totaling approximately 260 million. Additionally, we evaluate the performance using the COCO-2014 (Lin et al. 2014) validation set, split into 30k (COCO-30K) and 2k (COCO-2K) captions for assessing different metrics. We also benchmark performance using the PartiPrompts dataset (Yu et al. 2022), which contains over 1600 prompts across diverse categories and challenge aspects.

4.2 Backbone

We employ SDXL (Podell et al. 2023) as the backbone for our experiments, specifically we train a LoRA (Hu et al. 2021) through distillation.

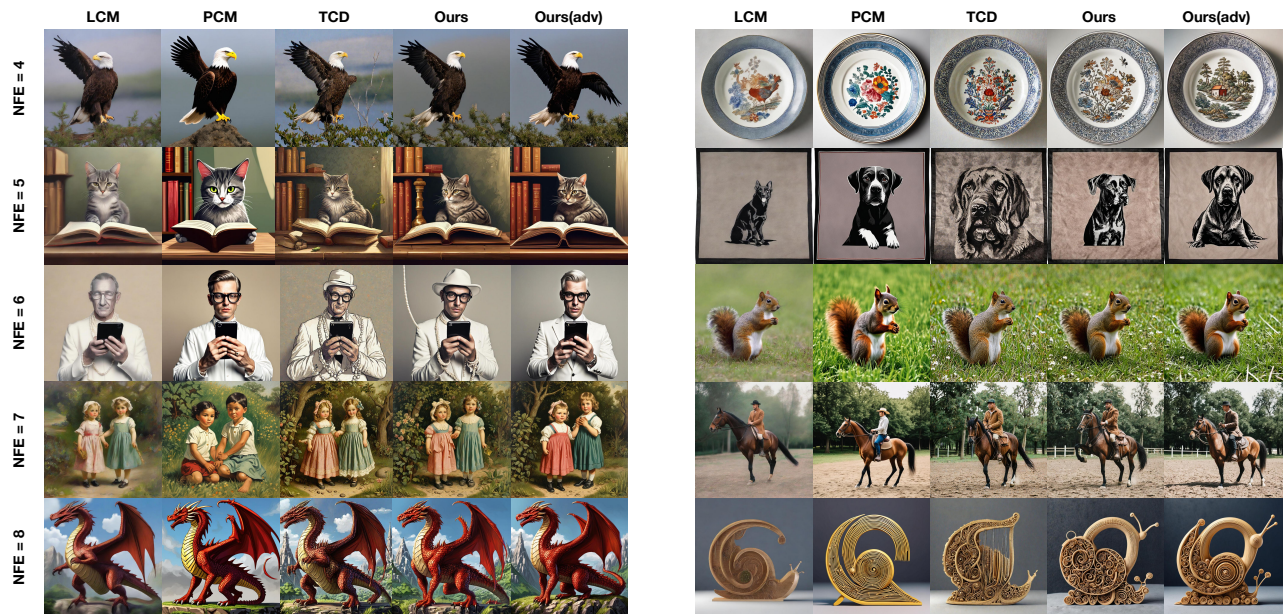


Figure 4: Qualitative comparison of different methods under NFE from 4 to 8 steps.

4.3 Metrics

We evaluate image generation quality using Frechet Inception Distance (FID) (Heusel et al. 2017) and assess image content richness with the Image Complexity Score (IC) (Feng et al. 2022). Additionally, we use Pickscore (Kirstain et al. 2023) to measure human preference. FID and IC are tested on the COCO-30K dataset, while PickScore is evaluated on COCO-2K and PartiPrompts.

4.4 Performance Comparison

In this section, we present a comprehensive performance evaluation of our proposed method against several baselines, including LCM, PCM, and TCD, across the COCO-30K, PartiPrompts, and COCO-2K datasets, as detailed in Table 1. We introduce two methods, “Ours” and “Ours*”, representing normal sampling and non-equidistant sampling (4-step and 8-step as still normal sampling), respectively. Additionally, “Ours(adv)” incorporates PCM’s (Wang et al. 2024) adversarial process during distillation, demonstrating that our method can effectively integrate adversarial training.

As shown in Figure 4, we qualitatively compare Ours and Ours(adv) with other methods across different inference steps. Our model outperforms others in image quality and text-image alignment. Quantitative results in Section 3.4 show that Ours achieves the best FID. While our method may not always achieve the top Image Complexity (IC), it avoids generating less detailed or cluttered images, unlike TCD, as seen in Figure 4. However, the high image complexity observed in TCD may also be attributed to certain visual artifacts and high-frequency noise, which we will elaborate on in the appendix. Furthermore, evaluations using PickScore on the COCO-2K and PartiPrompt datasets show that Ours and Ours* rank first or second in most cases. Overall, our method demonstrates superior performance and



a good balanced across metrics.

4.5 Ablation Study

Effect of Target Timestep Selection To demonstrate the advantages of Target Timestep Selection, we compared the performance of models trained on mappings required for 4-step inference (*i.e.*, PCM) against those trained on mappings required for 4-8 step inference. We maintained consistent settings with a batch size of 128, a learning rate of $5e-06$, and trained for a total of 15,000 steps. As shown in Figure 5, when using deterministic sampling (*i.e.*, $\gamma = 0$), the model trained on mappings for 4-8 step inference showed only a slight advantage at 4 and 5 steps. However, when incorporating randomness into sampling (*i.e.*, $\gamma = 0.2$), the model trained on 4-8 step mappings outperformed the model trained on 4-step mappings across all 4-8 steps. Furthermore, when we extended the mapping range by $\eta = 0.3$ to better accommodate the randomness in sampling (as indicated by the red line in Figure 5), inference with $\gamma = 0.2$ achieved a well-balanced performance across 4-8 steps, avoiding poor performance at 4-5 steps while also maintaining solid performance at 6-8 steps.

Effect of Distillation with Decoupled Guidances and Guidance Scale Tuning

To demonstrate the advantages of distillation with decoupled guidances, we conducted experiments with a batch size of 128, $K_{\min} = 4$, $K_{\max} = 8$, and $\eta = 0.3$, training two models with empty prompt ratios of 0 and 0.2. After 15k steps, we performed inference using a CFG scale of 3, as shown in Figure 6 (a). This approach effectively stabilized image quality and reduced visual artifacts. Additionally, we applied the guidance scale tuning to models like TCD and PCM, which were distilled with higher CFG values (corresponding to original CFGs of 9 and 5.5, respectively, when inferred with CFG 1). Guid-

METHOD	FID ↓					Image Complexity Score ↑				
	COCO-30K					COCO-30K				
	4 steps	5 steps	6 steps	7 steps	8 steps	4 steps	5 steps	6 steps	7 steps	8 steps
LCM	18.424	18.906	19.457	19.929	20.494	0.419	0.417	0.415	0.412	0.409
PCM	22.213	21.921	21.916	21.786	21.772	0.433	0.447	0.458	0.465	0.471
TCD	<u>17.351</u>	<u>17.430</u>	<u>17.535</u>	<u>17.65943</u>	<u>17.771</u>	0.512	0.521	0.527	0.533	0.536
Ours	17.256	17.388	17.334	17.565	17.681	0.474	0.488	0.499	0.508	0.516

METHOD	PickScore ↑									
	PartisPrompts					COCO-2K				
	4 steps	5 steps	6 steps	7 steps	8 steps	4 steps	5 steps	6 steps	7 steps	8 steps
LCM	22.182	22.226	22.246	22.251	22.237	22.143	22.223	22.268	22.296	22.297
PCM	22.200	22.289	22.340	22.357	22.367	22.291	22.433	22.505	22.537	<u>22.543</u>
TCD	22.350	22.402	22.417	22.426	22.426	22.310	22.423	22.490	22.504	22.504
Ours	<u>22.338</u>	<u>22.414</u>	22.445	22.472	22.493	22.396	<u>22.533</u>	<u>22.593</u>	22.640	22.656
Ours(*)	-	22.431	22.472	22.489	-	-	22.577	22.630	22.652	-
Ours(adv)	22.279	22.388	<u>22.450</u>	<u>22.476</u>	<u>22.490</u>	<u>22.322</u>	22.466	22.502	22.511	22.508

Table 1: Quantitative Comparison under different metrics and datasets.

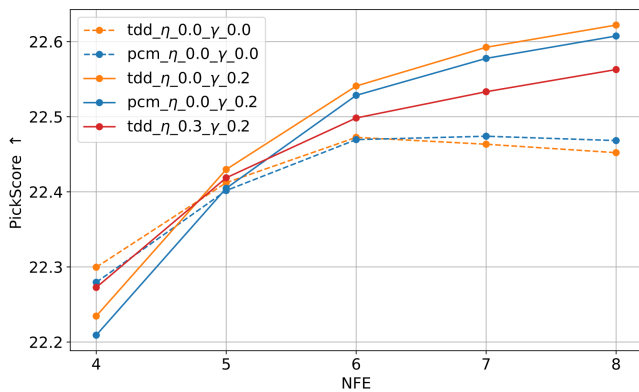


Figure 5: Qualitative comparison between Target-Driven Multi-Target Distillation and Single-Target Distillation.

ance scale tuning successfully converted these models to use normal CFG values during inference, significantly enhancing image content richness by reducing the CFG.

Effect of x_0 clipping Sample In Figure 6, we demonstrate the advantages of x_0 clipping. For some samples inferred with a low guidance scale, such as the one on the far left, certain defects may appear during inference. Increasing the guidance can alleviate these issues to some extent, but it also increases contrast, as seen in the middle image of Figure 6. By applying clipping in the initial steps, we can partially correct these defects without increasing contrast. Additional examples and results from applying clipping beyond the first denoising step are provided in Appendix.

5 Conclusion

Consistency distillation methods have been proven to be effective in accelerating diffusion models' generative tasks. However, previous methods often face blur and detail loss due to simplistic strategies in target timestep selection. We

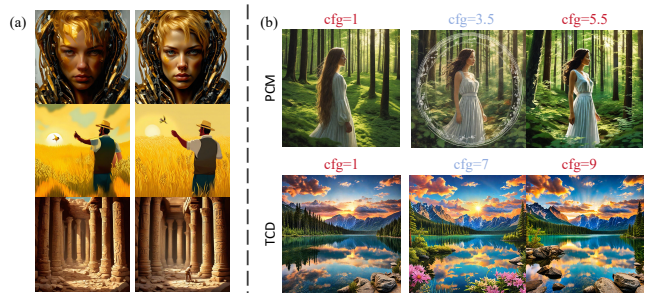


Figure 6: Ablation comparison of distillation with decoupled guidances (a) and guidance scale tuning (b).

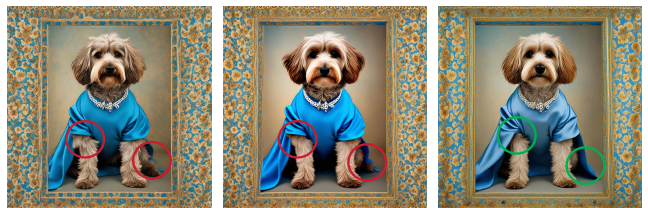


Figure 7: Ablation comparison of x_0 clipping sample. The images on the left are with CFG = 2, the middle with CFG = 3, and the right with CFG = 3 and x_0 clipping applied.

propose Target-Driven Distillation (TDD), which addresses these limitations by (1) employing a refined strategy for selecting target timesteps, thus enhancing training efficiency; (2) using decoupled guidance during training, which allows for post-tuning of the guidance scale during inference; and (3) incorporating optional non-equidistant sampling and x_0 clipping for more flexible and precise image sampling. Experiments demonstrate that TDD achieves state-of-the-art performance in few-step generation, providing a superior option among consistency distillation methods.

References

- Balaji, Y.; Nah, S.; Huang, X.; Vahdat, A.; Song, J.; Zhang, Q.; Kreis, K.; Aittala, M.; Aila, T.; Laine, S.; et al. 2022. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*.
- Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.
- Feng, T.; Zhai, Y.; Yang, J.; Liang, J.; Fan, D.-P.; Zhang, J.; Shao, L.; and Tao, D. 2022. IC9600: a benchmark dataset for automatic image complexity assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 8577–8593.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577.
- Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.
- Kim, D.; Lai, C.-H.; Liao, W.-H.; Murata, N.; Takida, Y.; Uesaka, T.; He, Y.; Mitsufuji, Y.; and Ermon, S. 2023. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kirstain, Y.; Polyak, A.; Singer, U.; Matiana, S.; Penna, J.; and Levy, O. 2023. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 36652–36663.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, 740–755. Springer.
- Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.
- Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022a. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*.
- Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022b. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*.
- Luo, S.; Tan, Y.; Huang, L.; Li, J.; and Zhao, H. 2023a. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.
- Luo, S.; Tan, Y.; Patil, S.; Gu, D.; von Platen, P.; Passos, A.; Huang, L.; Li, J.; and Zhao, H. 2023b. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*.
- Meng, C.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.
- Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.
- Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.
- Saharia, C.; Chan, W.; Chang, H.; Lee, C.; Ho, J.; Salimans, T.; Fleet, D.; and Norouzi, M. 2022a. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, 1–10.
- Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022b. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494.
- Sauer, A.; Boesel, F.; Dockhorn, T.; Blattmann, A.; Esser, P.; and Rombach, R. 2024. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*.
- Sauer, A.; Lorenz, D.; Blattmann, A.; and Rombach, R. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.
- Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294.
- Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.
- Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency models. *arXiv preprint arXiv:2303.01469*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Wang, F.-Y.; Huang, Z.; Bergman, A. W.; Shen, D.; Gao, P.; Lingelbach, M.; Sun, K.; Bian, W.; Song, G.; Liu, Y.; et al. 2024. Phased Consistency Model. *arXiv preprint arXiv:2405.18407*.

Wu, T.; Li, X.; Qi, Z.; Hu, D.; Wang, X.; Shan, Y.; and Li, X. 2024a. SphereDiffusion: Spherical Geometry-Aware Distortion Resilient Diffusion Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6126–6134.

Wu, T.; Zhang, Y.; Wang, X.; Zhou, X.; Zheng, G.; Qi, Z.; Shan, Y.; and Li, X. 2024b. Customcrafter: Customized video generation with preserving motion and concept composition abilities. *arXiv preprint arXiv:2408.13239*.

Yin, T.; Gharbi, M.; Zhang, R.; Shechtman, E.; Durand, F.; Freeman, W. T.; and Park, T. 2024. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6613–6623.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3): 5.

Zhang, Q.; and Chen, Y. 2022. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*.

Zheng, G.; Zhou, X.; Li, X.; Qi, Z.; Shan, Y.; and Li, X. 2023. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22490–22499.

Zheng, J.; Hu, M.; Fan, Z.; Wang, C.; Ding, C.; Tao, D.; and Cham, T.-J. 2024. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*.