Continuous-Time Distribution Matching for Few-Step Diffusion Distillation

Tao Liu[1,*], **Hao Yan**[2,*], **Mengting Chen**[2,†], **Taihang Hu**[2], **Zhengrong Yue**[2], **Zihao Pan**[2]
Jinsong Lan[2], **Xiaoyong Zhu**[2], **Ming-Ming Cheng**[1,‡], **Bo Zheng**[2,‡], **Yaxing Wang**[3,‡]
[1]VCIP, College of Computer Science, Nankai University [2]Alibaba Group
[3]College of Artificial Intelligence, Jilin University

https://byliutao.github.io/cdm_page/
https://github.com/byliutao/cdm



Figure 1: **CDM enables high-fidelity few-step text-to-image generation.** We compare our *Continuous-Time Distribution Matching* (CDM) against DMD2, both distilled from Longcat-Image (1024×1024) and evaluated at 4 NFE with identical prompts and seeds. Without relying on any GAN or reward-model auxiliary objectives, CDM produces sharper textures, richer fine-grained details, and overall higher visual fidelity, while DMD2 suffers from noticeable over-smoothing and detail loss. *(Best viewed zoomed-in.)*

Abstract

Step distillation has become a leading technique for accelerating diffusion models, among which Distribution Matching Distillation (DMD) and Consistency Distillation are two representative paradigms. While consistency methods enforce self-consistency along the full PF-ODE trajectory to steer it toward the clean data manifold, vanilla DMD relies on sparse supervision at a few predefined discrete timesteps. In practical DMD training, mode-seeking optimization operates with approximate scores and sparse discrete supervision, and the resulting models can exhibit visual artifacts and over-smoothed outputs. These empirical limitations motivate improving the supervision structure, while prior methods introduce auxiliary objectives—such as GANs or reward models—to improve visual fidelity. In this work, *we introduce **C**ontinuous-Time **D**istribution **M**atching (**CDM**), migrating the DMD framework from discrete anchoring to continuous optimization for the first time.* CDM achieves this through two continuous-time designs. First, we

*Equal contribution.
†Project leader.
‡Co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

replace the fixed discrete schedule with a dynamic continuous schedule of random length, so that distribution matching is enforced at arbitrary points along sampling trajectories rather than only at a few fixed anchors. Second, we propose a continuous-time alignment objective that performs active off-trajectory matching on latents extrapolated via the student's velocity field, improving generalization and preserving fine visual details. Extensive experiments on different architectures, including SD3-Medium and Longcat-Image, demonstrate that CDM provides highly competitive visual fidelity for few-step image generation without relying on complex auxiliary objectives.

1 Introduction

The remarkable capabilities of diffusion and flow-matching models [9, 14, 24, 27, 45, 51] have revolutionized text-to-image generation in recent years, setting new benchmarks for high-fidelity visual synthesis. Despite their exceptional generation quality, these models fundamentally rely on an iterative sampling process. This sequential procedure, typically demanding tens to hundreds of network evaluations, imposes a severe computational bottleneck that ultimately limits their real-world deployment. Accelerating this generation process without sacrificing sample quality has therefore become a central research challenge.

To bridge this gap, a variety of diffusion distillation paradigms have emerged [28, 31, 36, 47, 48, 52]. While early efforts reduced sampling to a few steps, the resulting models often struggle to balance inference speed with faithful text-image alignment. Among the diverse technical routes aimed at few-step synthesis, score-based distribution matching—prominently represented by Diff-Instruct [32] and Distribution Matching Distillation [62]—has emerged as a leading framework. By mathematically matching the student's output distribution with the pre-trained teacher's target distribution, these methods have demonstrated state-of-the-art performance in accelerating generative models.

Despite its success, existing DMD methods [1, 25, 61] inherit a structural limitation from their backward simulation strategy. To keep the simulated training trajectory consistent with the few-step inference procedure, they restrict the simulated timesteps to a fixed set of discrete anchors that matches the inference schedule. Unlike Consistency Distillation [29, 31, 52], which naturally optimizes trajectories within a continuous space, this strict confinement to sparse discrete schedules severely limits DMD. The lack of intermediate supervision can leave the student's velocity field insufficiently constrained between anchors. The underlying reverse KL objective is mode-seeking [30, 60], but this property alone does not imply oversmoothing or reduced per-sample fidelity. In practical DMD training, the student is optimized using approximate teacher and fake-model scores under sparse discrete supervision; models trained in this setting can exhibit oversmoothing and visual artifacts, and prior methods introduce auxiliary objectives such as GANs or reward models to improve visual fidelity [1, 61]. Our focus is on improving this supervision structure and the resulting distillation behavior, rather than changing the mode-seeking nature of the underlying reverse KL objective.

However, our preliminary empirical analysis challenges this strict training-inference alignment requirement [18] (Figure 2). We investigate an alternative formulation where the model is optimized via backward simulation using uniformly sampled continuous timesteps $t \in (0, 1]$ with random length at each training iteration, decoupling it from the fixed inference schedule. By simply randomizing the training timestep at each iteration, the student is trained over the full continuous time space rather than a few fixed points, and receives teacher gradients from a much wider range of trajectories. Empirically, this simple change not only preserves distillation performance, but yields consistent improvements: the dynamically scheduled model attains higher HPSv3 scores with finer details and fewer artifacts than its strictly aligned counterpart. This suggests that distribution matching is *schedule-independent*—rather than serving as a necessary anchor, the discrete schedule acts as an overly restrictive constraint on the student's achievable quality.

Given that distribution matching benefits from unrestricted continuous timesteps, it is crucial to understand what exactly the model learns from these matching signals. Recent studies [25, 63] decouple DMD training into a CFG Augmentation (CA) loss and a Distribution Matching (DM) loss, treating the latter simply as a "regularizer" for training stability and mitigating artifacts. However, visual evidence in Figure 3 (further supported by the quantitative validation in Appendix Table 5) reveals a fundamentally different paradigm. When student models are distilled solely with the DM

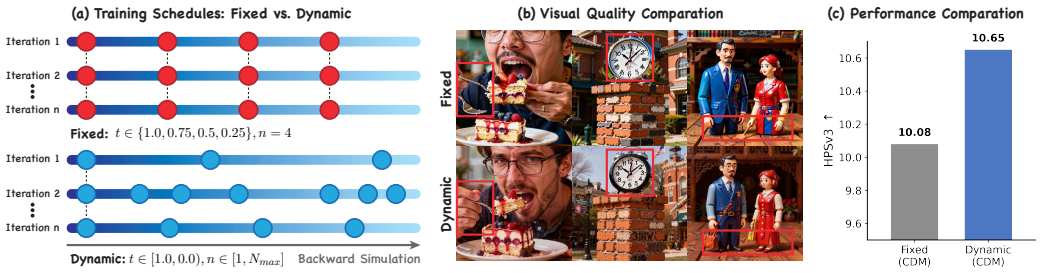

Figure 2: **Empirical evidence of schedule decoupling. (a)** Conventional distillation strictly anchors backward simulation to predefined discrete inference timesteps. In contrast, our dynamic scheduling optimizes over uniformly sampled continuous timesteps $t \in (0, 1]$ at each iteration. **(b)** Visually, the dynamically scheduled model produces finer details and fewer artifacts than the strictly aligned baseline. **(c)** Quantitatively, it also attains a higher HPSv3 score, indicating that exact discrete alignment is not only unnecessary but in fact restrictive—motivating our continuous-time formulation.

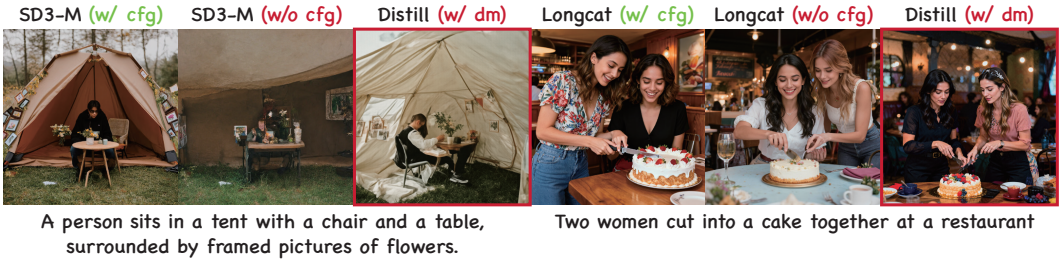

Figure 3: **Visual evidence on the role of the DM loss.** Samples from teacher models (SD3-Medium and Longcat-Image) with and without CFG, compared against student models distilled with the DM loss alone. Students distilled with the DM loss alone closely match their teachers' *CFG-free* samples, indicating that the DM loss is not a mere stabilizer but the key driver that aligns the student to the teacher's CFG-free distribution.

loss, their generated images closely match the samples produced by the teacher *without classifier-free guidance (CFG)*—which we refer to as the teacher's **CFG-free distribution**. This tight correlation indicates that the achievable performance of the DM loss is closely aligned with the teacher's CFG-free distribution. Rather than acting as a passive regularizer, the DM loss plays a substantive role in faithfully capturing this CFG-free distribution throughout the distillation process.

While continuous scheduling provides flexible on-trajectory supervision, few-step generation inevitably introduces severe numerical truncation errors due to large integration step sizes, causing the inference trajectory to drift off the ideal manifold [38, 39]. To directly counter this, we propose a novel *Continuous-Time Distribution Matching (CDM) loss*, which intrinsically incorporates a *velocity-driven extrapolation* mechanism into its matching objective. Instead of restricting supervision to on-trajectory latents, the CDM loss actively probes off-trajectory latents by taking a first-order step along the student's predicted velocity field, and enforces distribution matching upon them. Acting as a powerful spatial alignment objective, it effectively mitigates off-trajectory drift, empowering the student to self-correct integration errors and recover sharp, high-frequency details.

In summary, to the best of our knowledge, we are the first to migrate the DMD distillation framework from discrete schedules to a continuous optimization space. Our contributions are as follows:

- We empirically reveal two key insights in distribution matching: (1) anchoring the training optimization to a fixed set of discrete timesteps is not necessary; and (2) the distribution matching (DM) loss acts not merely as a "regularizer", but drives the student to align with the teacher's CFG-free distribution.
- To fully exploit these findings, we propose the CDM framework. This paradigm unifies a *dynamic continuous scheduling* strategy for flexible on-trajectory supervision, and a novel

off-trajectory CDM loss equipped with velocity-driven extrapolation to actively mitigate numerical integration errors during sampling.

- Extensive experimental results demonstrate that our continuous paradigm yields significant performance gains, establishing new state-of-the-art results for few-step image generation across different models (*e.g.,* SD3-Medium and Longcat-Image) without relying on complex auxiliary modules.

2 Related Work

Diffusion Distillation. While diffusion models [14, 45, 53] have achieved unprecedented success in visual generation tasks, their iterative sampling process poses a significant computational bottleneck. To accelerate inference, numerous distillation paradigms have been proposed. Progressive distillation [36, 46, 47] accelerates sampling by iteratively training a student to compress two teacher steps into one, progressively halving the required function evaluations. Consistency models [19, 29, 31, 41, 52, 57, 64] take a different approach by enforcing a self-consistency property: learning a direct mapping from any point along the probability flow ODE trajectory to the trajectory's origin on the data manifold, enabling few-step generation. Alternatively, adversarial distillation methods [22, 48] leverage a discriminator to align the few-step student's output directly with the real data distribution. Recent hybrid approaches further combine these paradigms: SANA-Sprint [3] and SwiftVideo [55] unify continuous-time consistency distillation with adversarial distribution alignment or trajectory distribution alignment, while TwinFlow [5] pairs consistency modeling with self-adversarial distribution matching to enable high-fidelity one-step generation.

Score-based Distillation. Score-based distillation originated in text-to-3D generation, where SDS [42] and VSD [58] leveraged pretrained diffusion scores to optimize 3D representations, establishing the conceptual foundation of distribution matching for distillation. Extending this paradigm to 2D image generation, Diff-Instruct [32] and DMD [62] formulated KL-based distribution matching frameworks for distilling diffusion models into few-step generators, with DMD2 [61] further improving stability via adversarial losses. Subsequent theoretical analyses [25, 63] decoupled the score distillation objective, revealing that CFG augmentation drives few-step conversion while the distribution matching term serves as a stabilizing regularizer. More recently, the DMD framework has been extended along multiple axes: scaling to large flow-based models [11], incorporating RL-based or GAN-based refinement [1, 17, 44], combining with consistency distillation or progressive distillation [10, 44, 59], introducing scale-wise distillation [2, 54], score identity distillation [65], or cache-aware distillation [21, 37]. Despite these advances, all existing DMD-based methods evaluate the DM loss exclusively at sparse discrete timesteps, leaving the continuous trajectory unoptimized. To address these limitations, we propose Continuous-Time Distribution Matching (CDM), which introduces a dynamic continuous schedule together with a velocity-driven off-trajectory alignment objective, shifting the optimization to the continuous-time domain. Notably, a concurrent work [43] shares a similar off-trajectory insight, likewise rolling out the model's own velocity for a single step to construct off-trajectory states; it then re-noises these states and regresses them toward ground-truth samples for post-training alignment, whereas we apply distribution matching directly on the extrapolated latents in the data-free distillation setting.

3 Method

We present Continuous-Time Distribution Matching (CDM), a unified distillation framework that lifts the discrete-time DMD paradigm into a fully continuous-time formulation for high-fidelity few-step generation. We first formalize the decoupled Distribution Matching Distillation (DMD) baseline (Section 3.1). Building on this, we relax the fixed inference schedule into a dynamic continuous schedule and theoretically examine its implications for distribution matching (Section 3.2). Finally, in Section 3.3 we complement these with the CDM loss, which extends supervision from on-trajectory anchors to off-trajectory latents via a velocity-driven extrapolation, regularizing the student's velocity field across the continuous time domain. The unified training pipeline is illustrated in Figure 4.

3.1 Preliminaries: Decoupled Distribution Matching

The goal of our distillation framework is to train a student flow model $\mathcal{D}_\theta$ capable of generating high-quality samples in N discrete steps, by distilling knowledge from a pre-trained teacher model $\mathcal{D}_\phi$ that typically requires $T \gg N$ steps. Here, $\mathcal{D}(\mathbf{x}_t, t, \mathbf{c})$ denotes the model prediction that estimates the clean data from the noisy latent $\mathbf{x}_t$ at timestep $t \in (0, 1]$, conditioned on $\mathbf{c}$. Formally, assuming the underlying neural network v_θ is trained to predict the velocity field, the clean data estimate $\mathcal{D}_\theta$ is explicitly parameterized as:

$$\mathcal{D}_\theta(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{x}_t - t v_\theta(\mathbf{x}_t, t, \mathbf{c}). \tag{1}$$

Building upon DMD [62], DMD2 [61], and Decoupled DMD (D-DMD) [25], we employ a backward simulation strategy to construct the sampling trajectory. Specifically, starting from random noise $\mathbf{x}_{t_1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we generate the trajectory by numerically integrating the probability flow ODE along the student's predefined discrete time schedule $\{t_1, \ldots, t_N\}$. During this process, we extract an intermediate latent state $\mathbf{x}_{t_i}$, where the index $i \sim \mathcal{U}\{1, \ldots, N\}$ is uniformly sampled. The distillation objective is decoupled into two orthogonal components: a CFG Augmentation (CA) term and a Distribution Matching (DM) term:

$$\mathcal{L}_{\text{DMD}} = \mathcal{L}_{\text{CA}} + \mathcal{L}_{\text{DM}}. \tag{2}$$

CA Loss ($\mathcal{L}_{\text{CA}}$). To enforce text-image alignment, the latent $\mathbf{x}_{t_i}$ is passed through the student model to yield the clean data estimate $\mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$. This estimate is subsequently perturbed with noise to a random continuous timestep $\tau \in (0, 1]$ to form $\mathbf{z}_\tau$. Following DMD [62], we introduce a dynamic weighting factor $w_\tau = \|\mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \mathbf{c}) - \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})\|_1^{-1}$ to normalize the gradient's magnitude. The CA loss is then defined as:

$$\mathcal{L}_{\text{CA}} = \frac{1}{2} \left\| \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}) - \text{sg}\left[\mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}) + \underbrace{w_\tau \alpha \left(\mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \mathbf{c}) - \mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \varnothing) \right)}_{\Delta_{\text{ca}}^{\text{real}} \text{ (CFG Augmentation)}} \right] \right\|_2^2, \tag{3}$$

where $\mathbf{c}$ is the conditioning text, α is the guidance scale, and $\text{sg}[\cdot]$ is the stop-gradient operator.

DM Loss ($\mathcal{L}_{\text{DM}}$). To align the student's marginal distribution with the real data manifold, we similarly reuse the student's clean data estimate $\mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$. This estimate is independently perturbed with noise to another random continuous timestep $\tilde{\tau} \in (0, 1]$ to form $\mathbf{z}_{\tilde{\tau}}$. Using a frozen real teacher $\mathcal{D}_\phi$ and an online-updated fake teacher $\mathcal{D}_\psi$ (which parameterizes the student's score), the DM loss is defined as:

$$\mathcal{L}_{\text{DM}} = \frac{1}{2} \left\| \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}) - \text{sg}\left[\mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}) + \underbrace{w_{\tilde{\tau}} (\mathcal{D}_\phi(\mathbf{z}_{\tilde{\tau}}, \tilde{\tau}, \mathbf{c}) - \mathcal{D}_\psi(\mathbf{z}_{\tilde{\tau}}, \tilde{\tau}, \mathbf{c}))}_{\Delta_{\text{dm}}^{\text{real-fake}} \text{ (Distribution Matching)}} \right] \right\|_2^2, \tag{4}$$

where $\mathcal{D}_\phi$ and $\mathcal{D}_\psi$ denote the frozen real teacher and the online-updated fake teacher (which parameterizes the student's generated distribution), respectively.

3.2 Dynamic Time Schedule

In vanilla DMD2 [61] paradigm, the backward simulation strategy relies on a fixed, predefined set of discrete timesteps matching the target inference schedule, denoted as $\mathcal{S}_{\text{infer}} = \{t_1, \ldots, t_N\}$. To maintain strict training-inference consistency, prior methods force the backward simulation during training to exclusively operate on these exact points.

However, we propose to break this rigid constraint by introducing a continuous dynamic time schedule. In each training iteration, the backward simulation length N is no longer fixed but randomly sampled ($N \sim \mathcal{U}\{1, N_{\max}\}$). We then randomly generate a strictly decreasing continuous time sequence $1 = t_1 > t_2 > \ldots > t_N > 0$, where 1 represents pure noise and 0 represents the clean image. This dynamic schedule brings two independent benefits. First, the random simulation length N exposes the student to varying numbers of inference steps at training time and lets the teacher provide gradient signals over a more diverse distribution of intermediate latents $\mathbf{x}_{t_i}$. Second, the student's anchors t_i are no longer confined to the fixed discrete set $\mathcal{S}_{\text{infer}}$; instead, they are drawn from

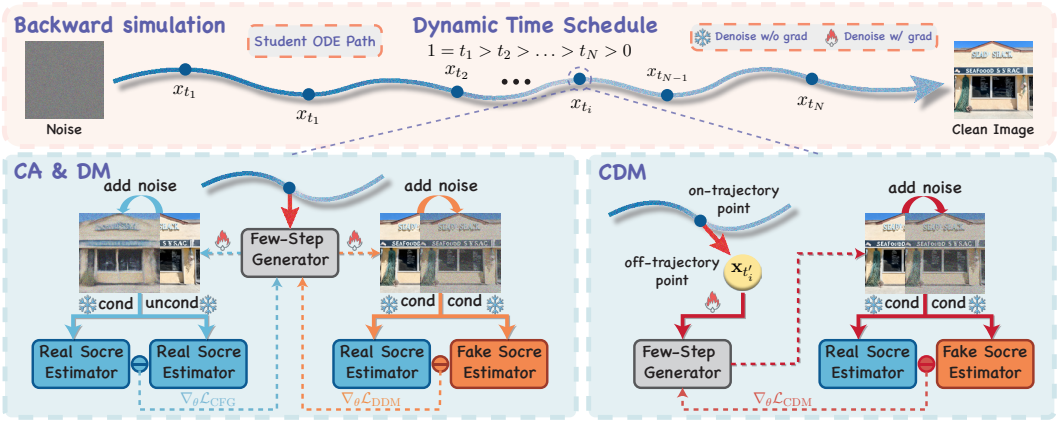

Figure 4: **Overview of Continuous-Time Distribution Matching (CDM). Top:** Our approach employs a dynamic continuous time schedule during backward simulation, sampling intermediate anchors uniformly from $(0, 1]$. **Bottom Left:** CFG augmentation (CA) and distribution matching (DM) operate on this dynamic schedule to align text-image conditions and data distributions at on-trajectory anchors. **Bottom Right:** To address inter-anchor inconsistency, the proposed CDM objective explicitly extrapolates off-trajectory latents ($\mathbf{x}_{t_i'}$) using the predicted velocity.

the same continuous domain $(0, 1]$ as the teacher's perturbation timesteps τ and $\tilde{\tau}$, which remain independently sampled. This eliminates the mismatch between the discrete student anchors and the continuous teacher supervision in vanilla DMD.

To provide a theoretical motivation for our dynamic time schedule, we examine the optimization from a score-matching perspective by applying Tweedie's formula [8] (see Appendix D for detailed derivations). Let $p_{\text{real}}(\mathbf{z}_t|\mathbf{c})$ denote the marginal distribution of the real data at a continuous noise level t, and $p_{\text{fake}}(\mathbf{z}_t|\mathbf{c})$ represent the fake target distribution.

For the CFG Augmentation (CA) loss, the gradient mathematically defines the direction of an implicit classifier $\log p_{\text{real}}(\mathbf{c}|\mathbf{z}_\tau)$ [63], effectively pushing the student's generation toward regions of higher text-image alignment:

$$\nabla_\theta \mathcal{L}_{\text{CA}} = -w_\tau \alpha \frac{\tau^2}{1-\tau} \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta} \right)^T \nabla_{\mathbf{z}_\tau} \log p_{\text{real}}(\mathbf{c}|\mathbf{z}_\tau). \tag{5}$$

For the Distribution Matching (DM) loss, the formulation reveals its analytical connection to the Kullback-Leibler divergence. Specifically, optimizing the DM loss corresponds to minimizing the KL divergence $D_{\text{KL}}(p^{\tilde{\tau}}_{\text{gen}} \| p^{\tilde{\tau}}_{\text{real}})$ between the student's generative distribution and the real data distribution at time $\tilde{\tau}$:

$$\nabla_\theta \mathcal{L}_{\text{DM}} = -w_{\tilde{\tau}} \frac{\tilde{\tau}^2}{1-\tilde{\tau}} \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta} \right)^T \left(\nabla_{\mathbf{z}_{\tilde{\tau}}} \log p_{\text{real}}(\mathbf{z}_{\tilde{\tau}}|\mathbf{c}) - \nabla_{\mathbf{z}_{\tilde{\tau}}} \log p_{\text{fake}}(\mathbf{z}_{\tilde{\tau}}|\mathbf{c}) \right). \tag{6}$$

Crucially, the student's input timestep t_i and the teacher's perturbation timesteps $\tau, \tilde{\tau}$ are independently sampled from the same continuous distribution over $(0, 1]$. In expectation, this mechanism encourages both the CA and DM gradients in Equations (5) and (6) to regularize the student's velocity field across the continuous time domain, rather than overfitting to sparse discrete anchors. While this continuous formulation provides a theoretical intuition for a smoother velocity field, we empirically validate its generalization benefits in Figure 2 and our experiments (Section 4).

3.3 Continuous-Time Distribution Matching (CDM)

The dynamic continuous schedule introduced in Section 3.2 provides supervision at randomly sampled anchors visited by backward simulation and can in principle cover any point at $(0, 1]$ given enough iterations. The supervision is applied to one anchor at a time: at each t_i, the loss only constrains the student's prediction $\mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$ to match the target distribution at that single point. It

does not constrain the student's velocity v_θ to remain consistent across adjacent time steps. Few-step inference, however, depends on this property: each Euler step from t_j to t_{j-1} introduces an error of order $\mathcal{O}((\Delta t)^2 \sup_\tau \|dv_\theta/d\tau\|)$, where the last term measures how rapidly v_θ changes between adjacent time steps (see Appendix E for a detailed derivation). Supervising each anchor in isolation gives no direct control over this term.

To reduce this inter-anchor inconsistency, we introduce the CDM loss, which adds supervision on intermediate latents between adjacent anchors. Given an on-trajectory latent $\mathbf{x}_{t_i}$ and its predicted velocity $v_{t_i} = v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$, we sample a paired anchor $t_i' \sim \mathcal{U}(0, 1]$ independent of the integration schedule and perform a first-order Euler extrapolation:

$$\mathbf{x}_{t_i'} = \mathbf{x}_{t_i} + (t_i' - t_i)\, v_{t_i}. \tag{7}$$

Because the underlying probability flow ODE trajectory is curved, a large stride $|t_i' - t_i|$ along the linearized velocity v_{t_i} produces an intermediate latent $\mathbf{x}_{t_i'}$ that lies between (or beyond) the discrete anchors and is not visited by standard backward simulation.

To supervise $\mathbf{x}_{t_i'}$, we construct the target latent directly from the local clean data estimate predicted at the extrapolated point. Specifically, we pass $\mathbf{x}_{t_i'}$ through the student model to obtain the local prediction $\hat{\mathbf{x}}_0^{(i')} = \mathcal{D}_\theta(\mathbf{x}_{t_i'}, t_i', \mathbf{c})$, and re-noise it to a continuous time $\hat{\tau} \sim \mathcal{U}(0, 1]$:

$$\mathbf{z}_{\hat{\tau}} = (1 - \hat{\tau})\, \mathrm{sg}[\hat{\mathbf{x}}_0^{(i')}] + \hat{\tau}\boldsymbol{\epsilon}_{\hat{\tau}}, \quad \boldsymbol{\epsilon}_{\hat{\tau}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{8}$$

By anchoring the reference target to the local estimate $\hat{\mathbf{x}}_0^{(i')}$, we establish a self-consistency constraint for the student's vector field. Due to the Euler extrapolation, $\mathbf{x}_{t_i'}$ naturally drifts off the ideal sampling trajectory. Re-noising this drifted prediction yielding $\mathbf{z}_{\hat{\tau}}$ allows the frozen teacher to evaluate the local score matching error. This localized supervision essentially penalizes invalid velocity predictions outside the main trajectory, promoting a smoother and more regularized flow for few-step integration.

The CDM loss is then defined on the extrapolated input and the $\hat{\mathbf{x}}_0^{(i')}$-anchored target:

$$\mathcal{L}_{\mathrm{CDM}} = \frac{1}{2} \left\| \mathcal{D}_\theta(\mathbf{x}_{t_i'}, t_i', \mathbf{c}) - \mathrm{sg}\Big[\mathcal{D}_\theta(\mathbf{x}_{t_i'}, t_i', \mathbf{c}) + w_{\hat{\tau}} \underbrace{\big(\mathcal{D}_\phi(\mathbf{z}_{\hat{\tau}}, \hat{\tau}, \mathbf{c}) - \mathcal{D}_\psi(\mathbf{z}_{\hat{\tau}}, \hat{\tau}, \mathbf{c})\big)}_{\Delta_{\mathrm{cdm}}^{\mathrm{real-fake}}} \Big] \right\|_2^2. \tag{9}$$

By matching the student's prediction at the off-trajectory latent $\mathbf{x}_{t_i'}$ to the target distribution, $\mathcal{L}_{\mathrm{CDM}}$ constrains v_θ across the continuous interval, reducing the inter-anchor inconsistency.

Full Training Objective. Our comprehensive training objective unifies these mathematical components into a single sum:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CA}} + \mathcal{L}_{\mathrm{DM}} + \mathcal{L}_{\mathrm{CDM}}. \tag{10}$$

4 Experiments

4.1 Experimental Setup

Experiment Setting. We conduct our main experiments on SD3-Medium [9] at a resolution of 1024×1024. For evaluation, we employ Aesthetic Score (AES) [49], PickScore [20], HPS v3 [35], and CLIP Score (ViT-H-14) [12] on 2K prompts sampled from the test split of the PickScore dataset [20]. We additionally report fine-grained prompt adherence on DPG-Bench (DPG) [16] using 1K prompts. For a comprehensive evaluation, we compare our method against several leading few-step generation baselines, including Hyper-SD [44], Flash [1], TDM [33], DMD2 [61], and D-DMD [25]. Furthermore, to demonstrate the broad applicability of our approach, we also extend our experiments to Longcat-Image [56]. Detailed experiment configurations are provided in Appendix C.

4.2 Main Results

Quantitative Results. As shown in Table 1, CDM achieves competitive performance against existing few-step baselines on both SD3-Medium and Longcat-Image with only 4 NFE. On SD3-Medium,



Figure 5: **Qualitative comparison on SD3-Medium.** CDM produces more photorealistic results with richer details than competing methods. All results are generated using the same initial noise and random seed for fair comparison.

Table 1: **Quantitative comparison of different methods on SD3-Medium and Longcat-Image.** For each backbone, the best and second-best results are highlighted in **bold** and underline, respectively; the base model serves as a reference and is excluded from the ranking. Methods marked with $*$ denote our reproduced results. Image-Free indicates methods that do not rely on real images during distillation; Continuous indicates methods whose supervision is applied at arbitrary continuous timesteps.

Method	NFE	Aesthetic↑	DPGBench↑	PickScore↑	HPSv3↑	CLIPScore↑	Image-Free	Continuous
SD3-Medium								
Base [9]	100	5.885	85.04	21.73	8.189	28.60	-	-
Hyper-SD [44]	8	5.180	80.43	20.82	6.054	27.93	✗	✓
Flash [1]	4	5.968	80.47	21.69	8.282	**28.18**	✗	✗
TDM [33]	4	6.013	83.12	21.61	8.468	27.63	✓	✗
DMD2* [61]	4	6.038	83.96	21.58	8.419	27.56	✓	✗
D-DMD* [25]	4	6.038	84.52	21.85	9.176	27.69	✓	✗
CDM (Ours)	4	**6.075**	**85.26**	**21.95**	**9.561**	27.98	✓	✓
Longcat-Image								
Base [56]	100	5.926	87.08	21.65	9.450	26.78	-	-
DMD2* [61]	4	5.800	87.12	21.07	8.803	**26.99**	✓	✗
D-DMD* [25]	4	5.782	88.04	21.23	9.629	26.57	✓	✗
CDM (Ours)	4	**5.919**	**88.35**	**21.53**	**10.65**	26.72	✓	✓

CDM obtains the best Aesthetic (6.075), DPGBench (85.26), PickScore (21.95), and HPSv3 (9.561), while maintaining a highly competitive CLIPScore. Among image-free methods, CDM compares favorably with D-DMD [25], achieving consistent improvements across all metrics, notably improving HPSv3 from 9.176 to 9.561. A similar trend holds on Longcat-Image, where CDM attains the best results on Aesthetic, DPGBench, PickScore, and HPSv3. Interestingly, our 4-NFE student matches or even surpasses the 100-NFE pretrained teacher on a range of metrics (*e.g.*, DPGBench and HPSv3) on both backbones, suggesting that the proposed continuous-time optimization framework provides supervision signals that go beyond merely replicating the teacher's outputs. More quantitative comparisons, including training and inference efficiency, are provided in Appendix G.

Qualitative Comparison. Figure 5 presents a side-by-side qualitative comparison against representative few-step baselines as well as the 100-NFE teacher. Across diverse prompts, CDM consistently yields sharper textures and fine-grained details (e.g., background elements and material reflections), and stronger semantic adherence to multi-entity compositional prompts, while competing baselines

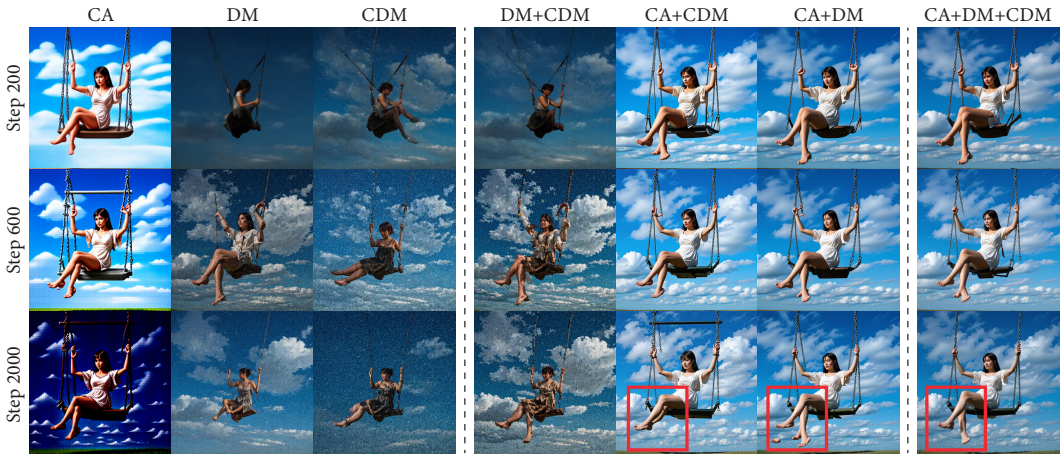

Figure 6: **Qualitative ablation of loss components across training steps. Left:** Individual losses (CA, DM, CDM) in isolation. **Right:** Pairwise and full combinations. Partial combinations suffer from brightness collapse or degraded local fidelity at later stages, whereas our full objective (CA+DM+CDM) effectively preserves both global semantic coherence and local details.

Table 2: **Ablation study on SD3-Medium at 4 NFE. Left:** Effect of individual loss components. **Right:** Analysis of core mechanism designs (time schedule, perturbation strategy, and target latent construction). The full CDM design achieves the best performance balance.

Configuration	AES↑	DPG↑	PICK↑	HPSv3↑	CLIP↑
(a) Single Loss Ablation					
w/o $\mathcal{L}_{\text{CA}}$	5.861	72.87	21.05	8.128	24.78
w/o $\mathcal{L}_{\text{DM}}$	6.016	84.57	21.75	8.954	27.66
w/o $\mathcal{L}_{\text{CDM}}$	6.067	85.12	21.85	9.153	27.91
(b) Dual Loss Ablation					
w/o $\mathcal{L}_{\text{DM}}$&$\mathcal{L}_{\text{CDM}}$	4.634	3.45	17.50	-10.15	14.60
w/o $\mathcal{L}_{\text{CA}}$&$\mathcal{L}_{\text{CDM}}$	5.787	70.60	20.82	7.258	25.31
w/o $\mathcal{L}_{\text{CA}}$&$\mathcal{L}_{\text{DM}}$	5.778	72.38	20.80	7.331	24.78
Full CDM	**6.075**	**85.26**	**21.95**	**9.561**	**27.98**

Model Variant	AES↑	DPG↑	PICK↑	HPSv3↑	CLIP↑
(a) Time Schedule					
w/ Fixed Schedule	6.051	83.84	21.89	9.482	27.75
(b) Off-trajectory Perturbation					
w/o Perturbation (on-traj)	6.027	84.43	21.94	9.374	27.90
w/ Gaussian Perturb.	6.040	84.65	21.92	9.516	27.88
(c) Target Latent Construction					
w/ Full-trajectory $\hat{\mathbf{x}}_0$ target	6.026	85.14	21.92	9.346	27.97
Full CDM	**6.075**	**85.26**	**21.95**	**9.561**	**27.98**

often exhibit blurry high-frequency content or missing attributes. Notably, despite operating with only 4 NFE, CDM matches or even visually surpasses the 100-NFE teacher in perceptual sharpness and aesthetics on many cases, corroborating the quantitative trends in Table 1. Additional visual results on both SD3-Medium and Longcat-Image are provided in Appendix J.

4.3 Ablation Study

Loss Components. We ablate each loss component in Table 2 and Figure 6. Relying solely on $\mathcal{L}_{\text{CA}}$ leads to structural collapse, whereas using only $\mathcal{L}_{\text{DM}}$ or $\mathcal{L}_{\text{CDM}}$ recovers visual quality but struggles with prompt adherence (*e.g.*, noticeably lower CLIP scores). Pairing $\mathcal{L}_{\text{CA}}$ with either distribution-matching loss bridges this gap, significantly improving both alignment and aesthetics. Ultimately, our full objective ($\mathcal{L}_{\text{CA}} + \mathcal{L}_{\text{DM}} + \mathcal{L}_{\text{CDM}}$) achieves the best performance across all metrics, with HPSv3 peaking at 9.561. This confirms their complementary roles: $\mathcal{L}_{\text{CA}}$ anchors structure and semantic alignment, while $\mathcal{L}_{\text{DM}}$ and $\mathcal{L}_{\text{CDM}}$ provide essential on- and off-trajectory distributional supervision, respectively.

Core Mechanism Design. To validate the critical design choices in Continuous-Time Distribution Matching, we perform an in-depth ablation on its three core mathematical components in the right panel of Table 2: the dynamic time schedule, the off-trajectory perturbation strategy, and the target latent construction. First, reverting our *dynamic schedule* to a standard fixed schedule leads to a considerable drop in generation fidelity, confirming that unifying the simulated trajectory and continuous distribution matching limits structural discrepancy. Second, replacing our *velocity-driven extrapolation* with a Gaussian noise baseline—implemented by first predicting the clean data $\mathbf{x}_0$ from $\mathbf{x}_t$ and re-adding noise to yield $\mathbf{x}'_t$—or removing perturbations entirely, degrades overall performance. This indicates that conventional re-noising fails to capture meaningful off-trajectory states,

whereas our velocity-based extrapolation accurately simulates the exact truncation drift encountered during large-step inference. Finally, utilizing the full-trajectory *final generation* $\hat{\mathbf{x}}_0$ rather than the intermediate target $\hat{\mathbf{x}}_0^{(i')}$ as the teacher's input yields suboptimal results. This validates that anchoring the supervision to localized predictions provides a more direct and effective signal for error correction than relying on the extended backward simulation.

5 Conclusion

In this paper, we present Continuous-Time Distribution Matching (CDM), a novel distillation framework for high-quality few-step diffusion generation. Existing discrete-time methods optimize on fixed timesteps, leading to accumulated discretization errors and detail degradation during few-step inference. To bridge this gap, CDM shifts the optimization into a continuous-time space, leveraging dynamic scheduling and an off-trajectory alignment objective ($\mathcal{L}_{\text{CDM}}$) to explicitly simulate and correct truncation drifts back to the target data manifold. Extensive experiments on SD3-Medium and Longcat-Image demonstrate that CDM effectively recovers sharp textures and semantic adherence, achieving state-of-the-art 4-step generation. Notably, it accomplishes this purely through robust continuous supervision, bypassing the need for adversarial training or costly external reward models. We hope this work paves the way for more accessible diffusion distillation and inspires future extensions to complex visual synthesis.

Acknowledgments and Disclosure of Funding

This research was supported by the General Program of the National Natural Science Foundation of China (Grant No. 62676167). We gratefully acknowledge this funding, which made the work presented in this paper possible. We would like to express our sincere gratitude to Yaxing Wang and Mengting Chen for their invaluable guidance and continuous support throughout this project. Their insightful advice on research directions, constructive feedback on our methods and experiments, and generous help in shaping this paper have been essential to its completion.

References

[1] Clement Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15686–15695, 2025.

[2] Feiyang Chen, Hongpeng Pan, Haonan Xu, Xinyu Duan, Yang Yang, and Zhefeng Wang. Cross-resolution distribution matching for diffusion distillation. *arXiv preprint arXiv:2603.06136*, 2026.

[3] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation, 2025.

[4] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation, 2025.

[5] Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. Twinflow: Realizing one-step generation on large models with self-adversarial flows. *arXiv preprint arXiv:2512.05150*, 2025.

[6] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[10] Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals. *arXiv preprint arXiv:2510.27684*, 2025.

[11] Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. *arXiv preprint arXiv:2506.00523*, 2025.

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[16] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[17] Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Zhen Li, Bo Zhang, et al. Distribution matching distillation meets reinforcement learning. *arXiv preprint arXiv:2511.13649*, 2025.

[18] 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.

[19] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[21] Haoyu Li, Tingyan Wen, Lin Qi, Zhe Wu, Yihuang Chen, Xing Zhou, Lifei Zhu, Xueqian Wang, and Kai Zhang. 1. x-distill: Breaking the diversity, quality, and efficiency barrier in distribution matching distillation. *arXiv preprint arXiv:2604.04018*, 2026.

[22] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Hongsheng Li, et al. Decoupled dmd: Cfg augmentation as the spear, distribution matching as the shield. *arXiv preprint arXiv:2511.22677*, 2025.

[26] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Yanzuo Lu, Yuxi Ren, Xin Xia, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Andy J Ma, Xiaohua Xie, and Jian-Huang Lai. Adversarial distribution matching for diffusion distillation towards efficient image and video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16818–16829, 2025.

[31] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[32] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

[33] Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17719–17728, 2025.

[34] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Sundeep Ranganath. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[35] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15086–15095, 2025.

[36] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14297–14306, 2023.

[37] Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation. *arXiv preprint arXiv:2601.09881*, 2026.

[38] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *International Conference on Machine Learning*, pages 26245–26265. PMLR, 2023.

[40] Gaurav Parmar, Richard Zhang, and Phillip Isola. On aliased resizing and surreptitious sabotage of the FID score. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

[41] Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. Facm: Flow-anchored consistency models. *arXiv preprint arXiv:2507.03738*, 2025.

[42] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[43] You Qin, Linqing Wang, Hao Fei, Roger Zimmermann, Liefeng Bo, Qinglin Lu, and Chunyu Wang. Soar: Self-correction for optimal alignment and refinement in diffusion models, 2026.

[44] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in neural information processing systems*, 37:117340–117362, 2024.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[46] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[47] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[48] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[49] Christoph Schuhmann. LAION-Aesthetics. https://laion.ai/blog/laion-aesthetics/, Aug 2022. Blog post.

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[52] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32211–32252, 2023.

[53] 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*, 2021.

[54] Nikita Starodubcev, Ilya Drobyshevskiy, Denis Kuznedelev, Artem Babenko, and Dmitry Baranchuk. Scale-wise distillation of diffusion models. *arXiv preprint arXiv:2503.16397*, 2025.

[55] Yanxiao Sun, Jiafu Wu, Yun Cao, Chengming Xu, Yabiao Wang, Weijian Cao, Donghao Luo, Chengjie Wang, and Yanwei Fu. Swiftvideo: A unified framework for few-step video generation through trajectory-distribution alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 9233–9241, 2026.

[56] Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

[57] Fu-Yun Wang, Zhaoyang Huang, Alexander W Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in neural information processing systems*, 37:83951–84009, 2024.

[58] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

[59] Hongyang Wei, Hongbo Liu, Zidong Wang, Yi Peng, Baixin Xu, Size Wu, Xuying Zhang, Xianglong He, Zexiang Liu, Peiyu Wang, et al. Skywork unipic 3.0: Unified multi-image composition via sequence modeling. *arXiv preprint arXiv:2601.15664*, 2026.

[60] Sirui Xie, Zhisheng Xiao, Diederik P Kingma, Tingbo Hou, Ying N Wu, Kevin Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. Em distillation for one-step diffusion models. *Advances in Neural Information Processing Systems*, 37:45073–45104, 2024.

[61] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[62] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[63] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023.

[64] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024.

[65] Mingyuan Zhou, Yi Gu, and Zhendong Wang. Few-step diffusion via score identity distillation. *arXiv preprint arXiv:2505.12674*, 2025.

[66] Kai Zou, Zhiwei Zhao, Bin Liu, and Nenghai Yu. Advancing aesthetic image generation via composition transfer. *International Journal of Computer Vision*, 134:252, 2026. Dataset: text-to-image-2M, https://huggingface.co/datasets/jackyhate/text-to-image-2M.

A Limitations

While CDM achieves strong few-step generation quality, it has several limitations that we leave for future work. First, although our dynamic continuous schedule and CDM loss do not introduce any additional cost at inference time, they do increase per-iteration training cost: the dynamic schedule samples a variable simulation length $N \sim \mathcal{U}\{1, N_{\max}\}$ that prolongs the average backward simulation, and the CDM loss requires an extra forward pass of the real and fake teachers on the extrapolated off-trajectory latent $\mathbf{x}_{t'_i}$ (see Appendix G for detailed results). Second, as a distillation framework, CDM is fundamentally upper-bounded by the teacher: the DM and CDM losses both rely on the teacher's score as the supervision signal, so concepts or compositions that the teacher itself handles poorly are unlikely to be recovered through distillation alone, as suggested by the CFG-free analysis in Figure 3. Third, our empirical study centers on text-to-image generation such as SD3-Medium and Longcat-Image; the class-conditional ImageNet 256×256 study in Appendix I.2 extends the evaluation to matched budgets of 2,000 and 4,000 student updates. The longer-budget results show that D-DMD substantially narrows the FID gap, so the relative advantage depends on the training budget. These two budgets do not establish convergence; a fuller study with convergence curves, per-task tuning, and additional class-conditional baselines is left for future work. We will also explore extending CDM to text-and-image-to-image (TI2I) editing and to video diffusion models, where trajectory length and temporal consistency play a larger role, in future studies.

B Broader Impact

CDM offers a more efficient distillation recipe for diffusion models, reducing the inference cost of high-quality text-to-image generation by an order of magnitude and thereby improving the accessibility of these models on commodity hardware. Since our work only distills a pre-trained teacher and does not introduce new generative capabilities or training data, its potential risks—such as the misuse of generated imagery for misinformation or copyright infringement—are largely inherited from the underlying teacher model rather than amplified by our method. We encourage practitioners deploying CDM-distilled models to combine them with established safeguards such as NSFW filtering, invisible watermarking, and content provenance standards (e.g., C2PA).

C Experiment Details

C.1 Training Hyperparameters

C.1.1 SD3-Medium

We apply our framework to distill the pre-trained SD3-Medium [9] into a 4-step student model. The optimization relies on a unified objective comprising CFG Augmentation (CA), Distribution Matching (DM), and Continuous-Time Distribution Matching (CDM) regularization, combined with equal weights. For the CA loss, we adopt the same teacher timestep sample strategy as in D-DMD [25]. The training dataset consists of 200K prompts randomly sampled from the training sets of T2I-2M [66], LAION [50], ShareGPT-4o-Image [4], PickScore [20], and OCR [26]. We perform full fine-tuning on the student network using the AdamW optimizer with a batch size of 128. The learning rate is set to 1×10^{-5} for the student generator and 5×10^{-6} for the fake teacher. The weight decay is set to 0.001, and β values are $(0.9, 0.999)$. Following the Two Time-scale Update Rule (TTUR), the fake teacher is updated 2 times per student generator update. The CFG guidance scale (α) is maintained at 7.0, and the dynamic schedule length ($N_{\max}$) is set to 28. The entire distillation process runs for 4K iterations on 16 A100 GPUs, which takes approximately 24 hours.

C.1.2 LongCat

The distillation of our LongCat [56] 4-step generator shares the same basic training settings as SD3-Medium. To handle its large model size efficiently, we employ LoRA [15] fine-tuning with rank 64 and alpha 128. The training utilizes a batch size of 64 and requires 2K iterations on 16 A100 GPUs, taking approximately 24 hours to achieve optimal convergence.

C.2 Baseline Implementation Details

All baselines use SD3-Medium [9] as the teacher backbone, with the number of function evaluations (NFE) and inference-time hyperparameters kept consistent with each method's official configuration. For TDM [33] (https://github.com/Luo-Yihong/TDM), Hyper-SD [44] (https://huggingface.co/ByteDance/Hyper-SD), and Flash-Diffusion [1] (https://github.com/gojasper/flash-diffusion), we directly use their official checkpoints and recommended generation settings. For DMD2 [61] and D-DMD [25], since no official SD3-Medium checkpoints are publicly available, we re-implement them within our unified framework under the exact same setting as our CDM and notably without the GAN-based adversarial loss.

D A Score-Matching Perspective on $\mathcal{L}_{\text{CA}}$ and $\mathcal{L}_{\text{DM}}$

Tweedie's formula [8] provides the bridge between a denoiser's posterior-mean prediction and the underlying score function. We first derive its form under the flow-matching interpolation used throughout this paper (Appendix D.1), and then apply it to interpret the gradients of $\mathcal{L}_{\text{CA}}$ and $\mathcal{L}_{\text{DM}}$ in Equations (3) and (4) (Appendices D.2 and D.3). The two derivations together justify the claim in Section 3.2 that, under the dynamic continuous schedule, both losses regularize the student's velocity field uniformly over $(0, 1]$.

Throughout this section, $p_{\text{real}}(\mathbf{z}_\tau|\mathbf{c})$ denotes the marginal distribution of the real data (modeled by the frozen teacher $\mathcal{D}_\phi$) at continuous noise level τ, and $p_{\text{fake}}(\mathbf{z}_\tau|\mathbf{c})$ denotes the corresponding distribution implicitly defined by the online-updated fake teacher $\mathcal{D}_\psi$.

D.1 Tweedie's Formula under the Flow-Matching Interpolation

For completeness, we derive Tweedie's formula under the flow-matching interpolation.

Proposition 1 (Tweedie's Formula for Flow Matching)**.** *Consider the flow-matching forward process that interpolates between clean data $\mathbf{x}_0 \sim p_{\text{data}}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:*

$$\mathbf{z}_\tau = (1-\tau)\mathbf{x}_0 + \tau\boldsymbol{\epsilon}, \quad \tau \in [0, 1]. \tag{11}$$

Then the posterior mean of the clean data given the noisy observation satisfies

$$\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau] = \frac{\mathbf{z}_\tau + \tau^2 \nabla_{\mathbf{z}_\tau} \log p(\mathbf{z}_\tau)}{1-\tau}. \tag{12}$$

A denoiser $\mathcal{D}(\mathbf{z}_\tau, \tau)$ trained under the mean-squared-error objective converges to the posterior mean $\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau]$, so Equation (12) also yields the score-prediction identity used in the main text.

Proof: Given a clean sample $\mathbf{x}_0$, the conditional distribution of $\mathbf{z}_\tau$ is Gaussian:

$$p(\mathbf{z}_\tau|\mathbf{x}_0) = \mathcal{N}\big(\mathbf{z}_\tau;\, (1-\tau)\mathbf{x}_0,\, \tau^2\mathbf{I}\big). \tag{13}$$

The marginal distribution of $\mathbf{z}_\tau$ is obtained by integrating over the data distribution:

$$p(\mathbf{z}_\tau) = \int p(\mathbf{x}_0)\, \mathcal{N}\big(\mathbf{z}_\tau;\, (1-\tau)\mathbf{x}_0,\, \tau^2\mathbf{I}\big)\, d\mathbf{x}_0. \tag{14}$$

Differentiating $\log p(\mathbf{z}_\tau)$ with respect to $\mathbf{z}_\tau$ gives

$$\begin{aligned} \nabla_{\mathbf{z}_\tau} \log p(\mathbf{z}_\tau) &= \frac{\nabla_{\mathbf{z}_\tau} p(\mathbf{z}_\tau)}{p(\mathbf{z}_\tau)} \\ &= \frac{1}{p(\mathbf{z}_\tau)} \int p(\mathbf{x}_0)\, \nabla_{\mathbf{z}_\tau} \mathcal{N}\big(\mathbf{z}_\tau;\, (1-\tau)\mathbf{x}_0,\, \tau^2\mathbf{I}\big)\, d\mathbf{x}_0. \end{aligned} \tag{15}$$

Using the identity for the derivative of a Gaussian density,

$$\nabla_{\mathbf{z}_\tau} \mathcal{N}(\mathbf{z}_\tau;\, \boldsymbol{\mu},\, \tau^2\mathbf{I}) = \frac{\boldsymbol{\mu} - \mathbf{z}_\tau}{\tau^2}\, \mathcal{N}(\mathbf{z}_\tau;\, \boldsymbol{\mu},\, \tau^2\mathbf{I}),$$

with $\boldsymbol{\mu} = (1-\tau)\mathbf{x}_0$, we obtain

$$\begin{aligned}\nabla_{\mathbf{z}_\tau} \log p(\mathbf{z}_\tau) &= \frac{1}{p(\mathbf{z}_\tau)} \int p(\mathbf{x}_0)\, \frac{(1-\tau)\mathbf{x}_0 - \mathbf{z}_\tau}{\tau^2}\, \mathcal{N}\big(\mathbf{z}_\tau;\, (1-\tau)\mathbf{x}_0,\, \tau^2\mathbf{I}\big)\, d\mathbf{x}_0 \\ &= \frac{(1-\tau)\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau] - \mathbf{z}_\tau}{\tau^2}, \end{aligned} \tag{16}$$

where the second equality follows from the definition of the posterior mean,

$$\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau] = \int \mathbf{x}_0\, p(\mathbf{x}_0|\mathbf{z}_\tau)\, d\mathbf{x}_0.$$

Solving for $\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau]$ yields

$$(1-\tau)\,\mathbb{E}[\mathbf{x}_0|\mathbf{z}_\tau] = \mathbf{z}_\tau + \tau^2 \nabla_{\mathbf{z}_\tau} \log p(\mathbf{z}_\tau), \tag{17}$$

from which Equation (12) follows. □

D.2 Score-Matching View of the CA Gradient

Treating the stop-gradient target in Equation (3) as a constant, the chain rule gives

$$\nabla_\theta \mathcal{L}_{\mathrm{CA}} = -w_\tau \alpha \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta}\right)^\top \left(\mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \mathbf{c}) - \mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \varnothing)\right). \tag{18}$$

Applying Proposition 1 to both teacher predictions and using $p_{\mathrm{real}}(\mathbf{z}_\tau|\varnothing) = p_{\mathrm{real}}(\mathbf{z}_\tau)$, the prediction difference reduces to

$$\mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \mathbf{c}) - \mathcal{D}_\phi(\mathbf{z}_\tau, \tau, \varnothing) = \frac{\tau^2}{1-\tau}\left(\nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{z}_\tau|\mathbf{c}) - \nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{z}_\tau)\right). \tag{19}$$

Bayes' rule $\log p(\mathbf{c}|\mathbf{z}_\tau) = \log p(\mathbf{z}_\tau|\mathbf{c}) - \log p(\mathbf{z}_\tau) + \log p(\mathbf{c})$, differentiated with respect to $\mathbf{z}_\tau$, eliminates the data-independent prior and gives the implicit-classifier identity

$$\nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{z}_\tau|\mathbf{c}) - \nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{z}_\tau) = \nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{c}|\mathbf{z}_\tau). \tag{20}$$

Substituting back into Equation (18),

$$\nabla_\theta \mathcal{L}_{\mathrm{CA}} = -w_\tau \alpha \frac{\tau^2}{1-\tau} \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta}\right)^\top \nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{c}|\mathbf{z}_\tau), \tag{21}$$

which matches Equation (5) in the main text. Hence $\mathcal{L}_{\mathrm{CA}}$ steers θ along the implicit-classifier gradient $\nabla_{\mathbf{z}_\tau} \log p_{\mathrm{real}}(\mathbf{c}|\mathbf{z}_\tau)$, which corresponds to maximizing text–image alignment.

D.3 Score-Matching View of the DM Gradient

The DM loss in Equation (4) aligns the student-induced distribution p_{fake} with the real-data distribution p_{real} by minimizing, in expectation, the reverse Kullback–Leibler divergence

$$\nabla_\theta D_{\mathrm{KL}}\big(p^{\tilde\tau}_{\mathrm{fake}} \,\|\, p^{\tilde\tau}_{\mathrm{real}}\big) = \mathbb{E}_{\mathbf{z}_{\tilde\tau}}\left[\left(\nabla_{\mathbf{z}_{\tilde\tau}} \log p_{\mathrm{fake}}(\mathbf{z}_{\tilde\tau}|\mathbf{c}) - \nabla_{\mathbf{z}_{\tilde\tau}} \log p_{\mathrm{real}}(\mathbf{z}_{\tilde\tau}|\mathbf{c})\right) \frac{\partial \mathbf{z}_{\tilde\tau}}{\partial \theta}\right], \tag{22}$$

at any continuous noise level $\tilde\tau \in (0, 1]$. Following DMD [62], p_{fake} is tracked online by the fake teacher $\mathcal{D}_\psi$, which provides a tractable estimate of $\nabla_{\mathbf{z}_{\tilde\tau}} \log p_{\mathrm{fake}}$ on the fly.

We now show that the gradient of $\mathcal{L}_{\mathrm{DM}}$ realizes Equation (22) in score form. Treating the stop-gradient target in Equation (4) as a constant, the chain rule gives

$$\nabla_\theta \mathcal{L}_{\mathrm{DM}} = -w_{\tilde\tau} \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta}\right)^\top \left(\mathcal{D}_\phi(\mathbf{z}_{\tilde\tau}, \tilde\tau, \mathbf{c}) - \mathcal{D}_\psi(\mathbf{z}_{\tilde\tau}, \tilde\tau, \mathbf{c})\right). \tag{23}$$

Applying Proposition 1 to $\mathcal{D}_\phi$ and $\mathcal{D}_\psi$ converts the prediction difference into a difference of conditional scores:

$$\mathcal{D}_\phi(\mathbf{z}_{\tilde\tau}, \tilde\tau, \mathbf{c}) - \mathcal{D}_\psi(\mathbf{z}_{\tilde\tau}, \tilde\tau, \mathbf{c}) = \frac{\tilde\tau^2}{1-\tilde\tau}\left(\nabla_{\mathbf{z}_{\tilde\tau}} \log p_{\mathrm{real}}(\mathbf{z}_{\tilde\tau}|\mathbf{c}) - \nabla_{\mathbf{z}_{\tilde\tau}} \log p_{\mathrm{fake}}(\mathbf{z}_{\tilde\tau}|\mathbf{c})\right). \tag{24}$$

Substituting into Equation (23),

$$\nabla_\theta \mathcal{L}_{\mathrm{DM}} = -w_{\tilde{\tau}} \frac{\tilde{\tau}^2}{1-\tilde{\tau}} \left(\frac{\partial \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})}{\partial \theta} \right)^\top \left(\nabla_{\mathbf{z}_{\tilde{\tau}}} \log p_{\mathrm{real}}(\mathbf{z}_{\tilde{\tau}}|\mathbf{c}) - \nabla_{\mathbf{z}_{\tilde{\tau}}} \log p_{\mathrm{fake}}(\mathbf{z}_{\tilde{\tau}}|\mathbf{c}) \right), \quad (25)$$

which matches Equation (6) in the main text. The bracketed score difference shares the structure of the integrand in Equation (22) and, in expectation, drives p_{fake} toward p_{real} at noise level $\tilde{\tau}$.

Together, Equations (21) and (25) share the same prefactor $\frac{\tau^2}{1-\tau} (\partial \mathcal{D}_\theta / \partial \theta)^\top$ and act on a score-valued target. Because the dynamic continuous schedule (Section 3.2) draws the student anchor t_i and the teacher perturbation timesteps $\tau, \tilde{\tau}$ independently from the same continuous distribution on $(0, 1]$, the resulting supervision is applied uniformly over the entire time domain rather than only at sparse discrete anchors.

E Local and Global Truncation Error of Euler Sampling

This appendix derives the local and global truncation error bounds for explicit Euler sampling to formally show that the error is controlled by M_2 (the supremum of the velocity field's material derivative), which is precisely the quantity suppressed by the CDM loss.

Setup and Local Error. Consider the probability-flow ODE $d\mathbf{x}_\tau / d\tau = v_\theta(\mathbf{x}_\tau, \tau, \mathbf{c})$ for $\tau \in [\epsilon, 1]$. Discretizing the interval with a schedule $1 = t_1 > t_2 > \cdots > t_N = \epsilon$ and step sizes $\Delta t = h_j := t_j - t_{j+1} > 0$, a single explicit Euler step from t_j down to t_{j+1} is:

$$\tilde{\mathbf{x}}_{t_{j+1}} = \mathbf{x}_{t_j} - h_j \, v_\theta(\mathbf{x}_{t_j}, t_j, \mathbf{c}). \quad (26)$$

Assuming v_θ is sufficiently smooth and L-Lipschitz in $\mathbf{x}$, we follow the standard local-error convention by comparing a single Euler step against the exact ODE solution passing through the same starting point $\mathbf{x}_{t_j}$ at $\tau = t_j$. Expanding the true solution around $\tau = t_j$ via Taylor's theorem with Lagrange remainder gives:

$$\mathbf{x}_{t_{j+1}} = \mathbf{x}_{t_j} - h_j \, \dot{\mathbf{x}}_{t_j} + \tfrac{1}{2} h_j^2 \, \ddot{\mathbf{x}}_{\xi_j}, \qquad \xi_j \in (t_{j+1}, t_j). \quad (27)$$

Substituting $\dot{\mathbf{x}}_\tau = v_\theta(\mathbf{x}_\tau, \tau, \mathbf{c})$ and subtracting Equation (26) yields the local truncation error:

$$\left\| \mathbf{x}_{t_{j+1}} - \tilde{\mathbf{x}}_{t_{j+1}} \right\| = \tfrac{1}{2} h_j^2 \left\| \ddot{\mathbf{x}}_{\xi_j} \right\| \le \tfrac{1}{2} h_j^2 \cdot M_2^{(j)} = \mathcal{O}\left((\Delta t)^2 \sup_{\tau \in [t_{j+1}, t_j]} \left\| \frac{dv_\theta}{d\tau} \right\| \right), \quad (28)$$

where $M_2^{(j)}$ bounds the material derivative $dv_\theta / d\tau$ (the total variation of v_θ along the trajectory) over the step:

$$M_2^{(j)} := \sup_{\tau \in [t_{j+1}, t_j]} \left\| \frac{dv_\theta}{d\tau} \right\| = \sup_{\tau \in [t_{j+1}, t_j]} \left\| \partial_\tau v_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}) + J_{\mathbf{x}} v_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}) \, v_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}) \right\|. \quad (29)$$

Global Error. Accumulating this local error over N steps down to $\tau = \epsilon$ yields the global error bound:

$$\left\| \mathbf{x}_\epsilon - \hat{\mathbf{x}}_\epsilon \right\| \le \frac{e^{L(1-\epsilon)} - 1}{L} \cdot \tfrac{1}{2} \, \bar{h} \, M_2, \qquad \bar{h} := \max_j h_j, \quad M_2 := \max_j M_2^{(j)}. \quad (30)$$

This bound shows that the global error is $\mathcal{O}(\bar{h})$ (since accumulating $\mathcal{O}(1/\bar{h})$ steps of $\mathcal{O}(\bar{h}^2)$ local error loses one order), and for a fixed step budget N, M_2 is the only factor that can be optimized by training.

How CDM Constrains M_2. The CDM loss supervises the student at the extrapolated off-trajectory latent:

$$\mathbf{x}_{t_i'} = \mathbf{x}_{t_i} + \Delta t \, v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}), \quad \text{where } \Delta t = t_i' - t_i. \quad (31)$$

To understand its regularization effect, consider the first-order Taylor expansion of the velocity field at this extrapolated point around $(\mathbf{x}_{t_i}, t_i)$:

$$\begin{aligned} v_\theta(\mathbf{x}_{t_i'}, t_i') &\approx v_\theta(\mathbf{x}_{t_i}, t_i) + \partial_\tau v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}) \Delta t + J_{\mathbf{x}} v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})(\mathbf{x}_{t_i'} - \mathbf{x}_{t_i}) \\ &= v_\theta(\mathbf{x}_{t_i}, t_i) + \Delta t \underbrace{\left(\partial_\tau v_\theta + J_{\mathbf{x}} v_\theta \, v_\theta \right)}_{= dv_\theta / d\tau}. \end{aligned} \quad (32)$$

Rearranging the terms reveals that the material derivative is functionally approximated by the finite difference taking the Euler leap:

$$\frac{dv_\theta}{d\tau} \approx \frac{v_\theta(\mathbf{x}_{t'_i}, t'_i) - v_\theta(\mathbf{x}_{t_i}, t_i)}{\Delta t}. \tag{33}$$

While standard distillation ensures the student matches the teacher at the anchor $\mathbf{x}_{t_i}$ (i.e., $v_\theta(\mathbf{x}_{t_i}, t_i) \approx v_\phi(\mathbf{x}_{t_i}, t_i)$), the CDM loss additionally enforces $v_\theta(\mathbf{x}_{t'_i}, t'_i) \approx v_\phi(\mathbf{x}_{t'_i}, t'_i)$. Together, they force the student's material derivative to mimic that of the teacher $\frac{dv_\theta}{d\tau} \approx \frac{dv_\phi}{d\tau}$. Since the pre-trained teacher naturally presents a smooth velocity field with bounded variation, CDM effectively transfers this smoothness to the student, implicitly regularizing M_2 and preventing the sporadic high-frequency oscillation typically observed in discretely trained models.

F Training Algorithm

For completeness, we summarize the full training procedure of CDM in Algorithm 1. The pseudocode reflects the implementation details described in Section 3: the dynamic continuous time schedule (Section 3.2), the decoupled CA and DM losses on backward-simulation anchors (Section 3.1), and the off-trajectory CDM loss anchored to the localized prediction $\hat{\mathbf{x}}_0^{(i')}$ (Section 3.3). All three student-side losses are computed within a single backward simulation per iteration; the intermediate latent $\mathbf{x}_{t_i}$ is extracted from the trajectory at a uniformly sampled anchor t_i. As established in D-DMD [25], this single shared latent $\mathbf{x}_{t_i}$ is used across all objectives to improve training efficiency, and the student parameters θ receive a single combined gradient step. The fake teacher ψ is updated on a separate optimizer using the standard flow-matching objective on the re-noised one-step student prediction $\hat{\mathbf{x}}_0^{(i)}$, sharing the same anchor i with the distillation losses.

Algorithm 1 Training Procedure of CDM

Require: Student θ, real teacher ϕ (frozen), fake teacher ψ, max length $N_{\max}$, guidance scale α, prompt set $\mathcal{C}$

repeat
  Sample prompt $\mathbf{c} \sim \mathcal{C}$, $N \sim \mathcal{U}\{1, \dots, N_{\max}\}$, schedule $1 = t_1 > t_2 > \cdots > t_N > 0$
  // 1. Backward simulation (no gradient)
  $\mathbf{x}_{t_1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; run θ for N Euler steps to obtain $\{\mathbf{x}_{t_n}\}_{n=1}^{N}$
  Sample anchor $i \sim \mathcal{U}\{1, \dots, N\}$
  // 2. Fake teacher update (on the re-noised one-step student prediction $\hat{\mathbf{x}}_0^{(i)}$)
  $\hat{\mathbf{x}}_0^{(i)} \leftarrow \mathcal{D}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$
  Sample $\tau_\psi \sim \mathcal{U}(0, 1]$; $\mathbf{z}_{\tau_\psi} \leftarrow (1 - \tau_\psi)\ \mathrm{sg}[\hat{\mathbf{x}}_0^{(i)}] + \tau_\psi\, \boldsymbol{\epsilon}_\psi$
  $\psi \leftarrow \psi - \eta_\psi\, \nabla_\psi \left\| v_\psi(\mathbf{z}_{\tau_\psi}, \tau_\psi, \mathbf{c}) - (\boldsymbol{\epsilon}_\psi - \mathrm{sg}[\hat{\mathbf{x}}_0^{(i)}]) \right\|_2^2$
  // 3. CA Loss
  Sample $\tau \sim \mathcal{U}(0, 1]$; $\mathbf{z}_\tau \leftarrow (1 - \tau)\ \mathrm{sg}[\hat{\mathbf{x}}_0^{(i)}] + \tau\, \boldsymbol{\epsilon}_\tau$
  Compute $\mathcal{L}_{\mathrm{CA}}$ (Equation (3))
  // 4. DM Loss
  Sample $\tilde{\tau} \sim \mathcal{U}(0, 1]$; $\mathbf{z}_{\tilde{\tau}} \leftarrow (1 - \tilde{\tau})\ \mathrm{sg}[\hat{\mathbf{x}}_0^{(i)}] + \tilde{\tau}\, \boldsymbol{\epsilon}_{\tilde{\tau}}$
  Compute $\mathcal{L}_{\mathrm{DM}}$ (Equation (4))
  // 5. CDM Loss
  Sample $t'_i \sim \mathcal{U}(0, 1]$; $\mathbf{x}_{t'_i} \leftarrow \mathbf{x}_{t_i} + (t'_i - t_i)\, v_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c})$
  $\hat{\mathbf{x}}_0^{(i')} \leftarrow \mathcal{D}_\theta(\mathbf{x}_{t'_i}, t'_i, \mathbf{c})$
  Sample $\hat{\tau} \sim \mathcal{U}(0, 1]$; $\mathbf{z}_{\hat{\tau}} \leftarrow (1 - \hat{\tau})\ \mathrm{sg}[\hat{\mathbf{x}}_0^{(i')}] + \hat{\tau}\, \boldsymbol{\epsilon}_{\hat{\tau}}$
  Compute $\mathcal{L}_{\mathrm{CDM}}$ (Equation (9))
  // 6. Student update
  $\theta \leftarrow \theta - \eta\, \nabla_\theta(\mathcal{L}_{\mathrm{CA}} + \mathcal{L}_{\mathrm{DM}} + \mathcal{L}_{\mathrm{CDM}})$ (Equation (10))
until converged

G More Quantitative Comparisons

To further evaluate model performance and training/inference resource consumption, we select the strongest baseline from Table 1, namely D-DMD [25], and conduct a more in-depth comparison against our CDM along two complementary dimensions: (i) additional quality metrics that are not covered by the main table (OCR accuracy for text rendering and FID for distributional fidelity), and (ii) training and inference efficiency.

Evaluation Protocol. For text rendering evaluation, we calculate OCR accuracy using PaddleOCR [6] on a test set of 1K OCR prompts from FlowGRPO [26]. Additionally, we compute the Fréchet Inception Distance (FID) [13] using 10K prompts from the COCO 2014 validation set [23]. Relative training time is normalized to D-DMD, and inference latency is measured at 1024×1024 resolution with 4 NFE on a single GPU under identical hardware and software conditions across all methods.

Results. As reported in Table 3, CDM delivers the strongest overall quality among all compared methods. Across the seven quality metrics, CDM ranks first on six of them (Aesthetic, DPGBench, PickScore, HPSv3, CLIPScore, and FID), while remaining closely competitive behind the fixed-schedule variant on OCR accuracy. The fixed-schedule variant of CDM consistently ranks second across most quality metrics, indicating that our overall framework is already a strong recipe and that the dynamic time schedule provides a further, consistent improvement on top of it. On the efficiency side, our dynamic continuous-time schedule and CDM loss introduce additional per-iteration training overhead—primarily due to the longer average backward simulation length and the extra forward pass on the extrapolated off-trajectory latent $\mathbf{x}_{t'_i}$—resulting in a relative training time of roughly $1.8\times$ that of D-DMD with comparable peak memory (62.5 vs. 62.2 GB). Note that DMD2 in our unified re-implementation (Appendix C.2) shares exactly the same per-iteration computation structure as D-DMD—the same backward simulation, real/fake teacher evaluations, and fake-teacher training, differing only in how the distillation losses are combined—and therefore has identical training cost, so the overhead of CDM relative to DMD2 matches that relative to D-DMD. Crucially, this overhead is confined to the training phase: at inference time, all compared methods share identical computational cost since they use the same backbone and the same number of function evaluations, so the per-image latency of CDM is on par with D-DMD (246 ms/img). The "Evals" column counts model evaluations per student update (one update spans two training batches under TTUR): the CDM loss adds only three evaluations over D-DMD ($16 \to 19$), while the remaining increase to 40 comes entirely from the longer gradient-free backward simulation of the dynamic schedule; because simulation and teacher forwards are gradient-free, this evaluation-count ratio ($2.5\times$) exceeds the wall-clock ratio ($1.8\times$).

Table 3: **Extended comparison with DMD2 and the strongest baseline D-DMD on SD3-Medium with 4 NFE.** We report the five main quality metrics (Aesthetic, DPGBench, PickScore, HPSv3, CLIPScore) together with two complementary quality metrics (OCR accuracy on 1K FlowGRPO prompts and FID on 10K COCO 2014 val prompts), as well as training memory, relative training time (normalized to D-DMD), model evaluations per student update ("Evals"), and inference latency. The best and second-best results in each column are highlighted in **bold** and underline, respectively.

	Quality (Main Metrics)					Quality (Extra)		Efficiency			
Method	**Aes↑**	**DPG↑**	**Pick↑**	**HPSv3↑**	**CLIP↑**	**OCR↑**	**FID↓**	**Mem (GB)**	**TrainTime**	**Evals**	**Latency (ms)**
DMD2* [61]	6.038	83.96	21.58	8.419	27.56	–	–	62.2	1.0×	16	246
D-DMD* [25]	6.038	84.52	21.85	9.176	27.69	33.47	31.47	62.2	1.0×	16	246
CDM (w/ Fixed Schedule)	6.051	83.84	21.89	9.482	27.75	**37.33**	31.05	62.5	1.1×	19	246
CDM	**6.075**	**85.26**	**21.95**	**9.561**	**27.98**	34.82	**30.30**	62.5	1.8×	40	246

Ruling out additional compute. A natural concern is that CDM's improvement stems from its higher per-iteration cost (Table 3) rather than from continuous-time supervision itself. Table 4 isolates this with two controls that instead give the *discrete* baseline more resources, both evaluated at 4 NFE: (i) densifying D-DMD's 4 anchors to 28 *fixed* discrete anchors ($\mathrm{linspace}(1.0, 0.25, 28)$, matching our $N_{\max} = 28$ and containing all four inference anchors), which multiplies the simulation steps per update by $7\times$ at the same 2,000 updates; and (ii) training D-DMD for $1.8\times$ updates (3,600), matching CDM's total GPU-hours. Neither control closes the gap. Despite $7\times$ more simulation steps, the dense variant is *worse* than 4-anchor D-DMD on four of the five metrics (the exception being prompt-agnostic Aesthetic): densely spaced anchors expose the student only to

small-step on-trajectory states, widening the train–test mismatch against 4-step inference, whereas our $N \sim \mathcal{U}\{1, N_{\max}\}$ schedule preserves large-step coverage and directly supervises large-step drift. The $1.8\times$ run does not improve either: HPSv3 *falls* ($9.211 \rightarrow 8.952$, roughly $7\times$ the cross-seed std), DPGBench, PickScore, and CLIPScore all decline, and only prompt-agnostic Aesthetic rises, indicating that a higher aesthetic score does not translate into better text alignment or overall performance in this setting. CDM's advantage is therefore attributable to continuous-time supervision, not to additional compute.

Table 4: **Continuous-time supervision vs. more compute and denser anchors (SD3-Medium, 4 NFE).** Starting from D-DMD, two controls give the discrete baseline more resources—densifying its 4 anchors to 28 *fixed* discrete anchors ($7\times$ simulation steps per update, same 2,000 updates), and training it for $1.8\times$ updates (GPU-hours matched to CDM)—while CDM is shown at the same $1.8\times$ GPU budget. Each row is averaged over 3 seeds (mean $\pm$ std); as these are re-measured under the multi-seed protocol, absolute values differ slightly from the single-run Table 1. Neither control closes the gap. The best and second-best results in each column are highlighted in **bold** and underline, respectively.

Variant (4 NFE)	Budget vs. D-DMD	Aes↑	DPG↑	Pick↑	HPSv3↑	CLIP↑
D-DMD (4 anchors)	1.0× (baseline)	6.037 ± 0.008	$\underline{84.61} \pm 0.08$	$\underline{21.86} \pm 0.01$	$\underline{9.211} \pm 0.035$	$\underline{27.70} \pm 0.02$
D-DMD + dense 28 anchors	7× sim. steps	$\underline{6.088} \pm 0.000$	82.77 ± 0.30	21.10 ± 0.01	8.036 ± 0.046	26.90 ± 0.05
D-DMD @ 3600 updates	1.8× updates	$\mathbf{6.137} \pm 0.006$	83.95 ± 0.24	21.56 ± 0.01	8.952 ± 0.004	26.62 ± 0.05
CDM	1.8× GPU-hrs	6.066 ± 0.013	$\mathbf{85.27} \pm 0.14$	$\mathbf{21.96} \pm 0.01$	$\mathbf{9.438} \pm 0.107$	$\mathbf{28.01} \pm 0.02$

H Quantitative Evaluation of the DM Loss

As discussed in Section 1 and visually demonstrated in Figure 3, relying solely on the Distribution Matching (DM) objective limits the student model to learning a marginal, unguided distribution. To quantitatively validate this, we evaluate the performance of the student models distilled exclusively with the DM loss and compare them against their respective teacher models running with and without Classifier-Free Guidance (CFG).

The results are summarized in Table 5. When disabling CFG for the teacher (Teacher CFG-free), we observe a significant deterioration in both semantic alignment metrics (e.g., DPGBench, HPSv3) and visual fidelity. Crucially, the student model distilled with the DM loss alone almost completely mirrors this performance drop, attaining metric scores that closely track the CFG-free teacher. As shown in Table 5, this highly consistent degradation pattern on both SD3-Medium and Longcat-Image confirms that the DM loss accurately aligns the student with the underlying teacher—but strictly with its weak, unguided marginal distribution.

Table 5: **Quantitative validation of the DM loss's alignment with CFG-free distributions.** The student model distilled exclusively with the DM loss heavily mirrors the performance deterioration of the CFG-free teacher across all metrics, confirming our visual observations in Figure 3.

Method	NFE	Aesthetic↑	DPGBench↑	PickScore↑	HPSv3↑	CLIPScore↑
SD3-Medium						
Teacher (with CFG)	100	5.885	85.04	21.73	8.189	28.60
Teacher (CFG-free)	50	5.681	69.65	20.24	4.693	24.32
Student (DM loss only)	4	5.787	70.60	20.82	7.258	25.31
Longcat-Image						
Teacher (with CFG)	100	5.926	87.08	21.65	9.450	26.78
Teacher (CFG-free)	50	5.952	84.72	21.12	8.253	25.63
Student (DM loss only)	4	6.028	80.53	20.10	8.186	21.69

I CDM under Varying Inference Steps

Although our student is distilled with a target of 4 NFE, the continuous-time training paradigm of CDM does not bind the resulting model to any specific inference schedule. On the one hand,

the dynamic continuous schedule (Section 3.2) randomizes the backward simulation length $N \sim \mathcal{U}\{1, N_{\max}\}$ at every training iteration, so the student is exposed to trajectories of varying lengths rather than a single fixed grid. On the other hand, the $\mathcal{L}_{\text{CDM}}$ loss (Section 3.3) further regularizes the student's velocity field v_θ across the continuous time domain, which directly suppresses the per-step truncation error of order $\mathcal{O}((\Delta t)^2 \sup_\tau \|dv_\theta/d\tau\|)$ that dominates few-step Euler integration. As a result, CDM remains usable across a range of NFEs at test time without any retraining or schedule-specific tuning. Below we verify this quantitatively on text-to-image (Appendix I.1) and further test generalization to class-conditional ImageNet generation (Appendix I.2).

I.1 Text-to-Image (SD3-Medium)

Figure 7 shows samples generated by the same CDM checkpoint under NFE $\in \{3, 4, 6, 8\}$ with identical prompts and seeds, where the model produces coherent and well-aligned images throughout the range and progressively recovers finer details as the NFE increases. To quantify this, we evaluate the same checkpoint-2000 of CDM and D-DMD under NFE $\in \{2, 4, 8\}$ on 2K PickScore prompts, forcing a uniform sigma schedule for both methods at every NFE so that neither benefits from its training-time grid. As reported in Table 6 (mean $\pm$ std over 3 seeds), CDM leads D-DMD on 14 of the 15 (metric $\times$ NFE) cells; the sole exception (Aesthetic at 8 NFE, -0.023) is within generation noise and concerns the prompt-agnostic metric. Varying the *training* seed instead (3 seeds at the 4-NFE working point), CDM's worst seed still exceeds D-DMD's best seed on all five metrics, so the lead is not a seed artifact. The margin is largest exactly where truncation error is greatest (2 NFE) and persists at 8 NFE, confirming that continuous-time supervision generalizes across inference budgets rather than overfitting the 4-step target.

Table 6: **Quantitative robustness across inference budgets (SD3-Medium, 2K PickScore prompts).** The same checkpoint-2000 of D-DMD and CDM at NFE $\in \{2, 4, 8\}$, both forced to a uniform sigma schedule at every NFE. Mean $\pm$ std over 3 seeds; **bold** marks the better method in each cell. CDM leads on 14 of the 15 (metric $\times$ NFE) cells.

NFE	Method	Aes↑	DPG↑	Pick↑	HPSv3↑	CLIP↑
2	D-DMD	5.738 ± 0.005	83.26 ± 0.30	21.17 ± 0.006	7.341 ± 0.012	27.33 ± 0.05
	CDM	$\mathbf{5.862} \pm 0.002$	$\mathbf{83.78} \pm 0.18$	$\mathbf{21.31} \pm 0.010$	$\mathbf{7.604} \pm 0.010$	$\mathbf{27.72} \pm 0.01$
4	D-DMD	6.037 ± 0.008	84.61 ± 0.08	21.86 ± 0.01	9.211 ± 0.035	27.70 ± 0.02
	CDM	$\mathbf{6.066} \pm 0.013$	$\mathbf{85.27} \pm 0.14$	$\mathbf{21.96} \pm 0.01$	$\mathbf{9.438} \pm 0.107$	$\mathbf{28.01} \pm 0.02$
8	D-DMD	$\mathbf{6.076} \pm 0.009$	83.74 ± 0.27	21.83 ± 0.006	9.220 ± 0.032	27.58 ± 0.04
	CDM	6.053 ± 0.009	$\mathbf{84.15} \pm 0.06$	$\mathbf{21.88} \pm 0.014$	$\mathbf{9.710} \pm 0.058$	$\mathbf{27.78} \pm 0.08$

I.2 Class-Conditional ImageNet Generation

To probe generalization beyond text-to-image, we add a class-conditional ImageNet 256×256 study under a matched, moderate-scale protocol. The teacher is SiT-XL/2 [34] (official weights); CDM and D-DMD share the same teacher, hyperparameters, aligned seeds, and global batch size of 128. We evaluate both methods at matched budgets of 2,000 student updates (matching the text-to-image ablation) and 4,000 student updates to examine whether the comparison changes with longer training. Training is image-free—only the 1,000 class labels are used, with real images serving solely as the FID reference statistics. We report FID-50K against the ADM reference statistics [7] computed with clean-fid [40], and Inception Score (IS) via torchvision.

Table 7 shows that CDM has lower FID at every evaluated NFE under both training budgets, but the magnitude of the advantage depends on the budget. At 2,000 updates, CDM reduces 4-NFE FID from 18.479 to 12.629 (approximately 31.7%). Doubling the training budget substantially improves D-DMD: its FID falls from 18.479 to 12.837 at 4 NFE and from 30.048 to 14.273 at 8 NFE. CDM improves more modestly at these NFEs, reaching FID 12.295 and 12.227, respectively; its relative FID advantage therefore narrows to approximately 4.2% at 4 NFE and 14.3% at 8 NFE. The IS comparison is also mixed at the longer budget: D-DMD has a slightly higher IS at 4 NFE (243.90 vs. 240.02), while CDM remains ahead at 2 and 8 NFE. At 2 NFE, longer training improves D-DMD



Figure 7: Generations from the same CDM checkpoint under varying NFE $\in \{3, 4, 6, 8\}$, using identical prompts and random seeds across columns. CDM produces coherent and prompt-aligned images across the full range, with finer details emerging as more inference steps are used.

Table 7: **Class-conditional ImageNet** 256×256 **under two training budgets (SiT-XL/2 teacher).** FID-50K (ADM reference statistics, clean-fid) and Inception Score at NFE $\in \{2, 4, 8\}$ after 2,000 and 4,000 student updates. CDM retains lower FID at every NFE, while longer training substantially narrows the gap. At 4,000 updates and 4 NFE, D-DMD has a slightly higher IS. Best within each training budget and NFE in **bold**.

		2,000 updates		4,000 updates	
Method	**NFE**	**FID-50K**↓	**IS**↑	**FID-50K**↓	**IS**↑
D-DMD	2	103.195	18.01 ± 0.30	86.520	22.17 ± 0.39
CDM	2	**47.975**	**49.12** ± 1.02	**49.974**	**45.06** ± 1.02
D-DMD	4	18.479	148.99 ± 2.31	12.837	**243.90** ± 2.59
CDM	4	**12.629**	**227.31** ± 2.20	**12.295**	240.02 ± 2.39
D-DMD	8	30.048	101.52 ± 1.55	14.273	208.87 ± 2.86
CDM	8	**12.618**	**239.90** ± 2.67	**12.227**	**252.61** ± 3.96

but slightly worsens CDM's FID ($47.975 \rightarrow 49.974$), so CDM does not improve uniformly with training budget.

Across inference budgets, D-DMD's FID worsens from 4 to 8 NFE under both training budgets, although the deterioration is much smaller after 4,000 updates ($12.837 \rightarrow 14.273$) than after 2,000 updates ($18.479 \rightarrow 30.048$); CDM's FID remains similar between 4 and 8 NFE. This pattern is consistent with a limitation of fixed-anchor supervision, but the longer-budget comparison also shows that the initial gap partly reflects the limited training budget. We therefore interpret the results as evidence of an advantage for CDM under the evaluated settings, rather than a budget-independent advantage or proof of convergence. The comparison focuses on NFE $\in \{2, 4, 8\}$; the initial 2,000-

update study did not yield converged one-step models. Full convergence curves, ImageNet-specific hyperparameter tuning, and comparisons against other recent class-conditional few-step methods remain open extensions.

J More Qualitative Results

To complement the main qualitative comparison in Figure 5, we provide additional samples generated by CDM on both backbones. Figure 8 shows results on SD3-Medium, covering a diverse set of prompts that span photorealistic portraits, complex scenes, stylized illustrations, and text-rich compositions. Figure 9 shows the corresponding results on Longcat-Image, demonstrating that the proposed continuous-time distribution matching framework generalizes consistently across different backbones. All images are generated with 4 NFE.



Figure 8: Additional qualitative results of CDM on SD3-Medium at 1024×1024 resolution with 4 NFE, covering diverse prompt categories. Zoom in for best view.



Figure 9: Additional qualitative results of CDM on Longcat-Image at 1024×1024 resolution with 4 NFE, covering diverse prompt categories. Zoom in for best view.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are supported by Section 4, Appendix D, and Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete (and correct) proof in Appendix D, and Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide these in Section 4 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide the data and code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide these in Section 4 and Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide these in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide these in Appendix C and Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and ensure that our research complies with all relevant guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide these in Appendix B.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We provide these in Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cited and stated.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No release of new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The paper does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.