Beyond Trajectory Matching: Reflow with Marginal Distribution Alignment

Chen Wang*
Tsinghua University
wangchen_23@mails.tsinghua.edu.cn

Peiran Yun*
Tsinghua University
yunpr22@mails.tsinghua.edu.cn

Pan Xie
ByteDance
xiepan.01@bytedance.com

Ke Deng†
Tsinghua University
kdeng@tsinghua.edu.cn

Abstract

Diffusion and continuous-flow generative models achieve high-quality generation, and their deterministic sampling can be formulated as solving learned ODE dynamics. However, accurate ODE discretization often requires many steps, making efficient few-step generation a key challenge. Among acceleration strategies, reflow-based distillation simplifies teacher ODE trajectories so that a student model can approximate the teacher transport with fewer steps. We identify a theoretical limitation of this paradigm, namely that trajectory matching can under-determine the distribution induced by the student model. In particular, two student models can attain the same trajectory-matching loss while inducing different endpoint marginal distributions, which may lead to different generation quality. To address this limitation, we introduce a marginal-alignment regularizer that penalizes the discrepancy between the student-induced marginal and the corresponding teacher marginal at the endpoint of each distillation interval. The regularizer is computed by tracking log-density changes along the ODE induced by the student model and evaluating scores from the frozen teacher model, without requiring auxiliary trainable networks or adversarial optimization. The resulting framework applies uniformly to the reflow family, including vanilla reflow and piecewise reflow. We further prove a telescoping total-variation bound showing that local marginal alignment controls the final-time discrepancy between the student-induced and teacher-induced distributions. Experiments on benchmark backbones demonstrate the effectiveness of the proposed method for few-step generation.

1 Introduction

In recent years, diffusion models [12, 46, 47] and *continuous normalizing flow* (CNF) models [6, 25, 26] have achieved remarkable progress in multimodal generation, enabling broad applications such as text-to-image generation [36, 40], text-to-video generation [2, 13], and image editing [3, 35]. A typical CNF-based generative model gradually transports samples drawn from a simple initial distribution to samples from the target distribution through an *ordinary differential equation* (ODE) with a learned or designed velocity field [6]. Although diffusion models are usually formulated through stochastic noising and denoising processes, their deterministic sampling can be performed by solving probability flow ODEs (PF-ODEs) [47]. This ODE view provides a common language for discussing both model families.

*Equal contribution.
†Corresponding author.

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

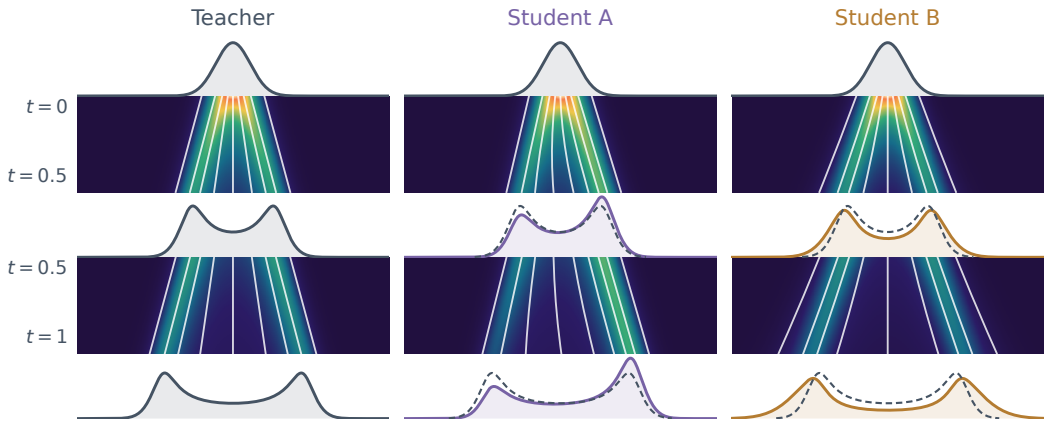


Figure 1: An example of the under-determination described in Theorem 1. Relative to the teacher, Students A and B attain the same trajectory matching loss in each window, yet induce different marginals at $t{=}0.5$ and $t{=}1$. The heatmaps show how the marginal of each model evolves from $t{=}0$ to $t{=}1$, with white lines tracing representative trajectories. The curves show the marginals at $t{=}0$, 0.5, and 1, and the dashed curves give the teacher marginals for reference.

In practice, however, ODE dynamics often vary substantially across time and space, so coarse discretization can introduce large numerical error and noticeably degrade sample quality. To reduce the number of sampling steps while preserving sample quality, many efforts have been made from this ODE-based sampling perspective. Among common solutions, training-free sampler design [29, 46] attempts to improve the numerical solver without additional training, whereas distillation [26, 42, 48] adapts the model and often provides better quality under very small step budgets.

Within the distillation route, reflow-style methods provide a principled paradigm for improving few-step sampling from generative ODEs by simplifying teacher ODE trajectories. Reflow [26] introduced this paradigm by learning ODE dynamics that connect the endpoints of teacher trajectories with simpler paths, so that accurate sampling can be achieved with fewer Euler steps. Piecewise variants such as PeRFlow [55] extend this principle by partitioning the full time interval into several sub-intervals and simplifying trajectories within each window, thereby approaching piecewise linear probability flows. These methods have made few-step generation more effective, yet their trajectory-level objectives leave open a basic question: *does trajectory-matching quality fully determine the quality of the induced marginals?* This question is central because generation quality ultimately depends on the induced marginals rather than merely on trajectory-level accuracy.

Our analysis gives a negative answer to this question. In practical distillation settings, velocity matching is typically imperfect due to finite model capacity and limited optimization. We prove that, under such imperfect matching, trajectory-level error does not fully determine the quality of the induced marginal distributions. We formalize this limitation in Theorem 1: two student velocity fields can attain *identical* trajectory MSE while inducing different marginal distributions at the interval endpoint, which can translate into different sample quality. Figure 1 gives an example of such a pair of students.

Motivated by this observation, we introduce a *marginal-alignment regularizer* for reflow-style distillation. The regularizer directly aligns the student-induced marginal with the corresponding teacher marginal at each time-interval endpoint, and applies uniformly to the reflow family, with vanilla reflow and PeRFlow as special cases. The regularizer complements the trajectory-matching loss by adding a distribution-level signal that favors better endpoint marginal alignment. We further show in Theorem 2 that the local marginal discrepancies penalized by the regularizer control the final-time discrepancy between the student-induced and teacher-induced distributions through a telescoping total-variation (TV) bound. The computation of the regularizer relies only on log-density tracking along the ODE induced by the student model and score evaluations from the frozen teacher model, requiring no auxiliary trainable networks or adversarial optimization. Empirically, augmenting reflow-based distillation with marginal alignment consistently improves few-step generation on benchmark backbones such as SD 1.5 and SDXL. On SD 1.5, for example, marginal alignment reduces the 4-step COCO-2014 FID of PeRFlow from 12.01 to 10.66.

In summary, our contributions are as follows:

1. We identify a distribution-level under-determination in reflow-style distillation under imperfect velocity matching, and demonstrate it through a constructive example.
2. We propose a marginal-alignment regularizer for the reflow family, which encourages interval-endpoint marginal alignment through an augmented ODE solve for the student dynamics and frozen-teacher score evaluations, without introducing additional trainable parameters.
3. We provide a theoretical guarantee for marginal alignment, showing that interval-wise marginal discrepancies control the final-time discrepancy between the student-induced and teacher-induced distributions via a telescoping total-variation bound.

2 Related Works

2.1 Continuous Normalizing Flows

Let $\pi_0 = \mathcal{N}(0, I_d)$ be the initial noise distribution, and π_1 be the data distribution. A CNF evolves π_0 into a model distribution close to π_1 under deterministic ODE dynamics. Given an initial random variable $X_0 \sim \pi_0$ and a time-varying velocity field $\vec{v} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, the trajectory $\{X_\tau\}_{\tau\in[0,1]}$ evolves according to

$$\frac{\mathrm{d}X_\tau}{\mathrm{d}\tau} = \vec{v}(X_\tau, \tau), \qquad \tau \in [0,1]. \tag{1}$$

For any $0 \le s \le t \le 1$, solving the ODE in Equation (1) from the initial state-time pair (x, s) to time t yields a flow map $\Phi_{\vec{v},\, s\to t}$:

$$\Phi_{\vec{v},\, s\to t}(x) := X_t, \qquad \frac{\mathrm{d}X_\tau}{\mathrm{d}\tau} = \vec{v}(X_\tau, \tau), \quad X_s = x, \quad \tau \in [s,t]. \tag{2}$$

For a neural velocity field $\vec{g}_\psi(x,t)$ with parameters ψ, we denote the induced marginal at time t along the CNF path by $q_{\psi,t} := \big(\Phi_{\vec{g}_\psi,\, 0\to t}\big)_{\#}\pi_0$, where $(T)_{\#}\pi$ denotes the distribution of $T(X)$ for $X \sim \pi$. In particular, $q_{\psi,0} = \pi_0$, and $q_{\psi,1}$ is the approximated model distribution by velocity field $\vec{g}_\psi$.

When $q_{\psi,t}$ admits a density, its evolution is governed by the Liouville equation,

$$\partial_\tau q_{\psi,\tau}(x) + \nabla_x \cdot \big(q_{\psi,\tau}(x)\vec{g}_\psi(x,\tau)\big) = 0, \tag{3}$$

which implies the instantaneous change-of-variables formula along a trajectory [6, 10]:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\log q_{\psi,\tau}(X_\tau) = -\operatorname{div}_x \vec{g}_\psi(X_\tau,\tau), \tag{4}$$

where $\operatorname{div}_x \vec{g}_\psi$ is the trace of the Jacobian $\nabla_x \vec{g}_\psi$.

In practice, the velocity field $\vec{g}_\psi(x,t)$ modeled by a deep neural network with parameter ψ is trained on a collection of representative samples from π_1 using objectives such as flow matching [25]. After training, the resulting model distribution is expected to approximate the data distribution, namely $q_{\psi,1} \approx \pi_1$. Given the learned velocity field $\vec{g}_\psi$, a sample Y from the model distribution $q_{\psi,1}$ can be generated from a sample $Z \sim \pi_0$ by following the ODE dynamics in Equation (1). With an M-step discretization of the time interval $[0,1]$, this trajectory takes the form

$$Z = X_0 \to X_{t_1} \to X_{t_2} \to \cdots \to X_{t_{M-1}} \to X_1 = Y, \tag{5}$$

where the symbol '$X_s \to X_t$' stands for a single Euler step:

$$X_t = X_s + \vec{g}_\psi(X_s, s)\cdot(t-s).$$

2.2 Distillation by Piecewise Reflow

Given a pretrained CNF model with velocity field $\vec{g}_\psi$, a more efficient CNF model with velocity field $\vec{f}_\theta$ parameterized by θ can be obtained by distillation. Hereinafter, we refer to $\vec{g}_\psi$ and $\vec{f}_\theta$ as the teacher and student velocity fields, respectively. In distillation, the student is trained to define alternative ODE dynamics that approximate the teacher transport with simpler trajectories and fewer

sampling steps. A popular distillation strategy is Piecewise Reflow (PeRFlow) [55], which partitions the time interval $[0, 1]$ into K windows ($K \ll M$) with cutting points $0 = t_0 < t_1 < \cdots < t_K = 1$, and trains $\vec{f}_\theta$ by the following total *trajectory matching loss*:

$$\mathcal{L}_{\mathrm{TM}}(\theta) = \sum_{k=1}^{K} \mathcal{L}_{\mathrm{TM}}^{(k)}(\theta), \tag{6}$$

where the k-th window-wise component is defined as

$$\mathcal{L}_{\mathrm{TM}}^{(k)}(\theta) = \mathbb{E}_{X\sim\pi_{t_{k-1}}} \int_{t_{k-1}}^{t_k} \left\| \vec{f}_\theta \left(X + \frac{\tau - t_{k-1}}{t_k - t_{k-1}}(Y - X),\, \tau \right) - \frac{Y - X}{t_k - t_{k-1}} \right\|^2 \mathrm{d}\tau, \tag{7}$$

where π_t is the marginal of $X_t = a(t)X_0 + b(t)X_1$ with independent $X_0 \sim \pi_0$, $X_1 \sim \pi_1$ and a noise schedule $\{(a(t), b(t))\}_{t\in[0,1]}$ satisfying $a(0) = 1, b(0) = 0, a(1) = 0, b(1) = 1$, and $Y := \Phi_{\vec{g}_\psi,\, t_{k-1}\to t_k}(X)$ is the endpoint of the $\vec{g}_\psi$-determined trajectory starting from X in the time window $W_k = [t_{k-1}, t_k]$. The trajectory matching loss $\mathcal{L}_{\mathrm{TM}}^{(k)}(\theta)$ encourages $\vec{f}_\theta$ to match the constant velocity along the straight line connecting X and Y within each time window $W_k = [t_{k-1}, t_k]$. This piecewise-linear approximation allows PeRFlow to reduce the number of sampling steps from M to K while largely preserving sample quality. When $K = 1$, PeRFlow recovers the vanilla reflow procedure of Rectified Flow [26].

2.3 Other Distillation Methods

Other reflow-family methods. Reflow-based distillation originates from Rectified Flow [26], which first proposed the trajectory-matching objective for learning straighter ODE dynamics. InstaFlow [27] scaled this idea to text-to-image generation. Subsequent work refined rectified-flow training [22], and Rectified Diffusion [50] further showed that reflow is compatible with arbitrary noise schedules and prediction parameterizations. Meanwhile, PeRFlow [55] extended reflow to the piecewise setting, and ProReflow [16] further improved its training by progressively reducing the number of windows and decomposing the velocity-matching loss to emphasize directional alignment. A major limitation of these methods lies in the fact that they rely on trajectory supervision alone and may under-determine the induced marginals, as formalized in Theorem 1.

Consistency distillation. Consistency Models [48] learn to map any point along an ODE trajectory directly to the endpoint at t=1, enabling one-step or few-step sampling. Following works have extended this idea to latent diffusion models [31], piecewise consistency constraints [49], mappings to arbitrary intermediate points along the trajectory [58], and improved training stability at scale [28, 39]. These methods differ from ours in that the student maps directly between time points rather than parameterizing a continuous-time velocity field.

Shortcut and flow-map distillation. Shortcut and flow-map models were introduced by CTM [18] and Shortcut Models [8] to learn the transport between any two points along an ODE trajectory. Align Your Flow [41] extended this idea to distilling large text-to-image models into flow maps, and SCFM [4] further distilled the teacher directly into an instantaneous velocity field with a shortcut-style objective. Similar to reflow-based methods, these methods rely on trajectory-level supervision alone and do not explicitly align the induced marginals.

Adversarial distillation. Adversarial distillation trains a discriminator to provide distribution-level supervision for few-step generators. ADD [44] introduced this paradigm in pixel-space, and LADD [43] moved the adversarial signal to latent space. Subsequent work combined adversarial losses with progressive distillation [23] and integrated them into unified acceleration pipelines [5]. While these methods introduce an explicit distributional signal, the additional discriminator network increases training complexity and can introduce instability.

Distribution matching distillation. Another line of work aligns generated distributions by comparing their noised score functions against those of a pretrained diffusion model. Since the noised score of the generated distribution is generally unavailable, SDS [37] sidesteps this by using only the teacher score, while VSD [51], Diff-Instruct [33], DMD [57], and DMD2 [56] train auxiliary

networks to approximate the student score. TDM [34] further extends this objective from the terminal distribution to the intermediate marginals of a few-step student. However, such learned proxies can lag behind the evolving student distribution, creating a moving-target problem.

In this paper, we augment reflow-based distillation with an explicit marginal-alignment objective, combining trajectory and marginal supervision while retaining a continuous-velocity student without requiring extra networks.

3 Method

3.1 Limitation of Trajectory Matching

Guided by the trajectory matching loss defined in Equation (7), PeRFlow imposes pointwise supervision on the student velocity field along interpolated teacher trajectories, thereby constraining the geometry of sample paths. However, as highlighted by Equation (4), the evolution of marginal densities along the flow depends on $\mathrm{div}_x \vec{f}_\theta$, a first-order spatial derivative of the velocity field, which is not constrained by the L^2 velocity error. Consequently, two velocity fields with the same trajectory matching loss may lead to very different endpoint marginals as summarized by Theorem 1.

Theorem 1 (Trajectory matching cannot guarantee marginal alignment)**.** *There exist a teacher velocity field $\vec{g}_\psi$ and two student velocity fields $\vec{f}_1, \vec{f}_2$ such that*

$$\mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_1) = \mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_2) \quad \textit{for all } k = 1, \dots, K, \quad \textit{but} \quad (\Phi_{\vec{f}_1,\, 0\to 1})_{\#}\pi_0 \neq (\Phi_{\vec{f}_2,\, 0\to 1})_{\#}\pi_0. \tag{8}$$

Section A.2 provides the detailed proof via an explicit construction. For simplicity, we present a Gaussian example here. For a constant teacher velocity field $\vec{g}_\psi(x,t) \equiv \mu$, choose a unit vector e with $\langle e, \mu\rangle = 0$ and define two student velocity fields

$$\vec{f}_1(x,t) = \mu + \varepsilon e \qquad \text{and} \qquad \vec{f}_2(x,t) = \mu + \varepsilon\langle x, e\rangle e, \tag{9}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product and $\varepsilon > 0$ is an arbitrary perturbation magnitude. When $\mathbb{E}_{X\sim\pi_{t_{k-1}}}\langle X, e\rangle^2 = 1$ for every k, it can be shown that the two students have the same trajectory matching loss on every window. However, the two student velocity fields have different divergences, $\mathrm{div}_x \vec{f}_1 = 0$ and $\mathrm{div}_x \vec{f}_2 = \varepsilon$, and by Equation (4) the marginals they induce also differ. To be specific, compared with the teacher marginal at every $t \in (0,1]$, the marginal induced by $\vec{f}_1$ has a shifted mean and that induced by $\vec{f}_2$ has a larger variance along e, and the two marginals also have different KL divergences to the teacher marginal. Section A.2 further constructs an example with a non-Gaussian teacher marginal, which is visualized in Figure 1.

3.2 Reflow with Additional Marginal-Alignment Regularizer

A natural idea to improve the defective trajectory matching loss is to enhance it by an extra regularizer that aligns the endpoint distribution induced by the student flow with the corresponding teacher marginal. To be concrete, let $q_{\psi,t_k} := (\Phi_{\vec{g}_\psi,\, 0\to t_k})_{\#}\pi_0$ be the inference-time marginal at time point t_k induced by the teacher velocity field $\vec{g}_\psi$. For each time window $W_k = [t_{k-1}, t_k]$, define the *training-protocol marginal* of the student velocity field $\vec{f}_\theta$ at time point t_k as

$$\tilde{p}_{\theta^{(k)}} := (\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k})_{\#}\pi_{t_{k-1}}, \tag{10}$$

that is, the distribution obtained by pushing the reference noising marginal $\pi_{t_{k-1}}$ through the student flow over the k-th window. We then define the *marginal-alignment loss* as

$$\mathcal{L}_{\mathrm{MA}}(\theta) := \sum_{k=1}^{K} \mathcal{L}_{\mathrm{MA}}^{(k)}(\theta) \qquad \text{where} \qquad \mathcal{L}_{\mathrm{MA}}^{(k)}(\theta) := \mathrm{KL}\big(\tilde{p}_{\theta^{(k)}} \,\big\|\, q_{\psi,t_k}\big) \tag{11}$$

measures the discrepancy between the student training-protocol marginal $\tilde{p}_{\theta^{(k)}}$ and the teacher endpoint marginal q_{ψ,t_k}. Enhancing the trajectory matching loss $\mathcal{L}_{\mathrm{TM}}$ with the marginal alignment loss $\mathcal{L}_{\mathrm{MA}}$, we get the following aggregated total loss with $\lambda \geq 0$ as hyper-parameter:

$$\mathcal{L}_{\mathrm{total}}(\theta) := \mathcal{L}_{\mathrm{TM}}(\theta) + \lambda\mathcal{L}_{\mathrm{MA}}(\theta). \tag{12}$$

Through the diffusion–flow correspondence reviewed in Section B, this marginal-alignment objective can also be applied to diffusion-model distillation under different noise schedules and prediction parameterizations.

Let $p_{\theta,t_k} := (\Phi_{\vec{f}_\theta,\,0\to t_k})_{\#}\pi_0$ be the inference-time marginal at time point t_k induced by the student velocity field $\vec{f}_\theta$. The following theorem shows that the window-wise marginal-alignment losses control the discrepancy between the terminal distributions of the student and the teacher at inference time.

Theorem 2 ($\mathcal{L}_{\mathrm{MA}}$ bounds the distributional gap)**.** *Let $\delta_k = \mathrm{TV}(q_{\psi,t_k}, \pi_{t_k})$ measure the gap between the teacher-induced marginal q_{ψ,t_k} and the reference noising marginal π_{t_k} in terms of the total variation distance, where $TV(\mu,\nu) := \sup_{A\in\mathcal{F}} |\mu(A) - \nu(A)|$ with $\mathcal{F}$ being the common sigma field of the two probability measures μ and ν. Then under Assumptions 1 and 3,*

$$\mathrm{TV}(p_{\theta,t_K}, q_{\psi,t_K}) \le \sum_{k=1}^{K} \sqrt{\tfrac{1}{2}\,\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta)} + \sum_{k=1}^{K-1} \delta_k. \tag{13}$$

The proof, given in Section A.3, bounds $e_k := \mathrm{TV}(p_{\theta,t_k}, q_{\psi,t_k})$ recursively. Since inference starts the k-th window from $p_{\theta,t_{k-1}}$ rather than $\pi_{t_{k-1}}$, the triangle inequality and the data processing inequality give $e_k \le e_{k-1} + \delta_{k-1} + \mathrm{TV}(\tilde{p}_{\theta^{(k)}}, q_{\psi,t_k})$, and Pinsker's inequality bounds the last term by $\sqrt{\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta)/2}$. Telescoping from $e_0 = 0$ yields Equation (13). The residuals δ_k measure the discrepancy between the teacher-induced and reference noising marginals and vanish when the two coincide. Since δ_k does not depend on θ, reducing the marginal-alignment loss tightens this upper bound.

Why the training-protocol marginal. One could instead try to minimize $\mathrm{KL}(p_{\theta,t_k} \| q_{\psi,t_k})$ directly. Here $p_{\theta,t_k} := (\Phi_{\vec{f}_\theta,\,0\to t_k})_{\#}\pi_0$ denotes the inference-time marginal induced by composing the student flows over the first k windows. However, computing this objective and its gradient would require unrolling the sampler from t_0 to t_k, so each window update would depend on the full prefix of the sampling chain. The training-protocol marginal $\tilde{p}_{\theta^{(k)}}$ reduces this cost by localizing the objective to the k-th window. It uses the reference noising marginal $\pi_{t_{k-1}}$, which is independent of θ, as the input distribution instead of the recursively generated marginal $p_{\theta,t_{k-1}}$. This choice remains tied to inference-time behavior through the telescoping bound in Theorem 2.

Role of trajectory matching. Marginal alignment is introduced as a complement to trajectory matching, not as a replacement for it. The PeRFlow objective provides trajectory-level supervision by matching the student velocity to the teacher-induced transport within each window, thereby distilling teacher ODE trajectories into local transports that are easier to execute with few-step sampling. The marginal-alignment loss then favors velocity fields that induce better density evolution among candidates with comparable trajectory error. This complementarity is nontrivial precisely when $\mathcal{L}_{\mathrm{TM}} > 0$, which is the realistic regime under finite model capacity. In principle, the marginal-alignment loss can also be combined with other trajectory-based distillation methods, such as the shortcut and flow-map methods discussed in Section 2.3.

3.3 Computation of the Marginal-Alignment Loss

The following two identities, proved in Sections A.4 and A.5, give a computable representation of $\mathcal{L}_{\mathrm{MA}}^{(k)}$ and its gradient.

Lemma 3 (Liouville representation)**.** *Under the regularity conditions specified in Section A.1,*

$$\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta) = \mathbb{E}_{X\sim\pi_{t_{k-1}}}\left[\log \pi_{t_{k-1}}(X) + \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta) - \log q_{\psi,t_k}\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)\right)\right], \tag{14}$$

where $X^{\mathrm{Stu}}_{t_{k-1}\to t}(X;\theta) := \Phi_{\vec{f}_\theta,\,t_{k-1}\to t}(X)$ denotes the student trajectory at time t initialized from X at time t_{k-1}, and $\Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta) := -\int_{t_{k-1}}^{t_k} \mathrm{div}_x\, \vec{f}_\theta\Big(X^{\mathrm{Stu}}_{t_{k-1}\to\tau}(X;\theta),\,\tau\Big)\,\mathrm{d}\tau$ is the corresponding log-density change.

Theorem 4 (Gradient formula)**.** *For brevity, write $X^{\mathrm{Stu}}_k := X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ and $\Delta^{\mathrm{Stu}}_k := \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$. Under the regularity conditions specified in Section A.1, the gradient of $\mathcal{L}_{\mathrm{MA}}^{(k)}$*

Algorithm 1 Reflow with Additional Marginal-Alignment Regularizer (MA-Reflow). Instantiated with PeRFlow.

Input: Training dataset $\mathcal{D}$, noise schedule $(a(t), b(t))$, teacher velocity field $\vec{g}_\psi$ with associated score s_ψ, window partition $\{t_k\}_{k=0}^K$, loss weight λ, alternating ratio m

Initialize the student model $\vec{f}_\theta$
repeat
  Sample data $X_1 \sim \mathcal{D}$ and noise $X_0 \sim \mathcal{N}(0, I_d)$
  Sample window $k \sim \mathrm{Uniform}\{1, \dots, K\}$ and time $t \sim \mathrm{Uniform}(t_{k-1}, t_k)$
  Get $X = a(t_{k-1})X_0 + b(t_{k-1})X_1$
  $Y \leftarrow \Phi_{\vec{g}_\psi,\, t_{k-1}\to t_k}(X)$ // teacher endpoint
  $z_t \leftarrow X + \frac{t-t_{k-1}}{t_k - t_{k-1}}(Y - X)$
  $\ell \leftarrow \left\| \vec{f}_\theta(z_t, t) - \frac{Y-X}{t_k - t_{k-1}} \right\|^2$ // PeRFlow loss
  if every m iterations **then**
    $(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}, \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}) \leftarrow \Phi_{\vec{f}^{\mathrm{aug}}_\theta,\, t_{k-1}\to t_k}(X, 0)$ // augmented student flow
    $\ell_{\mathrm{MA}} \leftarrow \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k} - \mathrm{stopgrad}(s_\psi(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}, t_k))^\top X^{\mathrm{Stu}}_{t_{k-1}\to t_k}$
    $\ell \leftarrow \ell + \lambda \ell_{\mathrm{MA}}$ // total loss
  end if
  Update θ using $\nabla_\theta \ell$
until convergence

can be written as

$$\nabla_\theta \mathcal{L}^{(k)}_{\mathrm{MA}}(\theta) = \mathbb{E}_{X\sim\pi_{t_{k-1}}}\left[\nabla_\theta \Delta^{\mathrm{Stu}}_k - s_\psi\big(X^{\mathrm{Stu}}_k, t_k\big)^\top \nabla_\theta X^{\mathrm{Stu}}_k\right], \tag{15}$$

where $s_\psi(x, t_k) := \nabla_x \log q_{\psi,t_k}(x)$ *is the teacher score function. Equivalently, the gradient in Equation* (15) *is obtained by optimizing the following stop-gradient form:*

$$\mathbb{E}_{X\sim\pi_{t_{k-1}}}\left[\Delta^{\mathrm{Stu}}_k - \mathrm{stopgrad}\big(s_\psi\big(X^{\mathrm{Stu}}_k, t_k\big)\big)^\top X^{\mathrm{Stu}}_k\right], \tag{16}$$

where $\mathrm{stopgrad}(\cdot)$ *is the standard deep-learning operator that leaves its argument unchanged in the forward pass and blocks gradients in the backward pass.*

In practice, we obtain $X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ and $\Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ by integrating the augmented student velocity field

$$\vec{f}^{\mathrm{aug}}_\theta((x, c), t) := \begin{pmatrix} \vec{f}_\theta(x, t) \\ -\mathrm{div}_x\, \vec{f}_\theta(x, t) \end{pmatrix} \tag{17}$$

from the initial augmented state $(X, 0)$ over $[t_{k-1}, t_k]$. The divergence is estimated by the Hutchinson identity $\mathrm{div}_x\, \vec{f}_\theta(x, t) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I_d)}[\epsilon^\top \nabla_x \vec{f}_\theta(x, t)\epsilon]$ [14]. Following the finite-difference implementation in JKO-iFlow [53], we approximate the directional derivative by

$$\nabla_x \vec{f}_\theta(x, t)\epsilon \approx \frac{\vec{f}_\theta(x + \sigma\epsilon, t) - \vec{f}_\theta(x, t)}{\sigma}, \tag{18}$$

and estimate the resulting trace with Monte Carlo samples during training. Additional estimator and solver details are given in Section C. The frozen teacher score s_ψ is derived from the teacher velocity field $\vec{g}_\psi$ via the velocity–score correspondence reviewed in Section B.

Algorithm 1 summarizes the training procedure of the proposed MA-Reflow method. In our experiments, we instantiate the trajectory-matching component with PeRFlow. Since the marginal-alignment loss requires integrating the augmented student velocity field, we evaluate it every m iterations and add it to the PeRFlow loss to reduce the computational cost of KL evaluation.

3.4 Comparison with Existing Methods

Table 1 compares MA-Reflow with the distillation families reviewed in Section 2.3 along four axes: whether the method uses trajectory-level supervision, whether it introduces an explicit marginal-level

Table 1: Positioning of MA-Reflow relative to representative prior work.

Methods	Trajectory loss	Marginal loss	No extra network	Continuous velocity student
Reflow-based distillation	✓		✓	✓
Consistency distillation	✓		✓	
Shortcut and flow-map distillation	✓		✓	✓
Adversarial distillation		✓		
Distribution matching distillation		✓		
MA-Reflow (ours)	✓	✓	✓	✓

objective, whether it avoids any auxiliary trainable network (e.g., critic, discriminator, or student-score estimator), and whether the student remains a continuous-time velocity field compatible with variable-step ODE solvers. We check the last axis for shortcut and flow-map distillation since SCFM provides an instantaneous velocity field. MA-Reflow is the only method that combines trajectory and marginal supervision while retaining a continuous-velocity student and requiring no extra networks.

4 Experiments

4.1 Experimental Setup

We evaluate the proposed marginal-alignment regularizer on two text-to-image backbones, Stable Diffusion v1.5 (SD 1.5) [40] and Stable Diffusion XL (SDXL) [36]. For both backbones, the student is initialized from a converged PeRFlow checkpoint [3] and further optimized with the joint objective in Equation (12). This setup reduces training cost and matches the role of our regularizer, which improves marginal alignment among students with comparable trajectory-matching quality. Additional experiments with a latent flow matching (LFM) teacher [7] on CelebA-HQ, covering both teacher and PeRFlow initialization, are provided in Section E.1.

Training configuration. For both SD 1.5 and SDXL experiments, training images are sampled from LAION-Aesthetics v2 5+ [45], following the PeRFlow training setup. Images are center-cropped and resized to 512×512 for SD 1.5 and 1024×1024 for SDXL. To support classifier-free guidance (CFG), we apply 10% text-conditioning dropout during training. The distillation uses $K=4$ windows. Within each window, the ϵ-prediction teacher produces the target endpoint using an 8-step DDIM sampler [46]. This sampler is known to correspond to Euler integration in a rescaled coordinate system, and Section B.2 restates this correspondence in our notation. The marginal-alignment regularizer is computed by solving the augmented ODE with a 4-step RK4 solver. The regularization weight λ is set to 10^{-6}. For efficiency, the marginal-alignment loss is evaluated once every nine PeRFlow updates. For other hyperparameters, including learning rate and weight decay, we follow the Hugging Face training scripts for Stable Diffusion.[4] All training runs are conducted on $2\times$NVIDIA H200 GPUs with BF16 mixed precision. Under this setup, fine-tuning for 5K steps takes approximately 50 GPU hours for SD 1.5 and 160 GPU hours for SDXL. The SDXL training time is approximately $1.94\times$ that of PeRFlow, and the per-update time and peak memory are reported in Section D.3.

Evaluation protocol. Following standard evaluation practice for text-to-image generation, we report Fréchet Inception Distance (FID; 11) and CLIP score [38]. Our evaluation uses the 2014 and 2017 validation sets of MS-COCO [24], following the BK-SDM [17] setup for the 2014 validation set and the Rectified Diffusion [50] setup for the 2017 validation set. For SDXL, we further evaluate human-preference alignment with ImageReward [54], HPSv2 [52], and PickScore [21] on 5K prompts from the 2017 validation set, and compositional generation with GenEval [9].

Baselines. Besides training-free ODE solvers, we compare with representative consistency, adversarial, distribution matching, and reflow-based distillation methods from Section 2.3.

[3] We use the PeRFlow checkpoints released on Hugging Face: `hansyan/perflow-sd15-delta-weights` for SD 1.5 and `hansyan/perflow-sdxl-base` for SDXL.

[4] `diffusers/examples/text_to_image`

Table 2: SD 1.5 results on COCO-2014 and COCO-2017 validation sets.

(a) COCO-2014 validation set.

Method	Steps	FID (↓)
Training-free ODE solvers		
DPMSolver [29]	25	9.78
DPMSolver [29]	8	22.44
DPMSolver++ [30]	4	22.36
DDIM [46] (teacher)	32	10.05
Distilled few-step models		
LCM-LoRA [32]	4	23.62
TCD [58]	4	16.02
2-ReFlow [26]	4	15.32
Flash Diffusion [5]	4	12.41
PeRFlow [55]	4	12.01
PCM [49]	4	11.70
ProReflow-I [16]	4	11.16
ProReflow-II [16]	4	10.70
MA-Reflow (ours, K=4)	4	**10.66**

(b) COCO-2017 validation set.

Method	Steps	FID (↓)	CLIP (↑)
Training-free ODE solvers			
DPMSolver [29]	25	20.10	31.80
DDIM [46] (teacher)	32	21.98	31.22
Distilled few-step models			
LCM-LoRA [32]	4	36.46	29.10
2-ReFlow [26]	4	32.97	28.93
TDM [34]	4	25.16	30.68
LCM [31]	4	24.39	30.50
PCM [49]	4	24.15	30.11
PeRFlow [55]	4	23.81	30.24
MA-Reflow (ours, K=4)	4	**22.76**	**31.03**

Table 3: SDXL results under 4-step sampling on COCO-2017 and GenEval.

Method	CLIP (↑)	ImageReward (↑)	HPSv2 (↑)	PickScore (↑)	GenEval (↑)
LCM-LoRA [32]	32.04	0.502	0.280	22.12	0.508
TCD [58]	32.38	0.682	0.297	22.44	0.511
PCM [49]	32.62	0.673	0.309	22.44	0.504
SDXL-Lightning [23]	32.90	0.716	0.306	22.64	0.514
HyperSD [39]	33.10	**0.906**	0.312	22.78	0.544
DMD2 [56]	33.13	0.899	0.316	22.76	0.575
PeRFlow [55]	33.44	0.804	0.298	22.38	0.550
MA-Reflow (ours, K=4)	**34.14**	0.897	**0.321**	**22.88**	**0.579**

4.2 Main Results

Tables 2 and 3 summarize the main quantitative results. Across both SD 1.5 and SDXL, the proposed method consistently improves few-step generation under the same 4-step inference budget. For SD 1.5, our method reduces COCO-2014 FID from 12.01 to 10.66 relative to PeRFlow and outperforms representative 4-step distillation baselines, including Flash Diffusion, PCM, and ProReflow. On COCO-2017, it also improves both evaluation metrics, reducing FID from 23.81 to 22.76 and increasing CLIP score from 30.24 to 31.03. Qualitative comparisons are provided in Section E.2. For SDXL, our method improves over PeRFlow on all five metrics in Table 3. Among all compared methods, it also achieves the best CLIP score, HPSv2, PickScore, and GenEval, and remains close to HyperSD and DMD2 on ImageReward. FID results are reported in Section E.3, where our method also achieves the best performance on COCO-2014-10K among the compared 4-step methods and remains close to Rectified Diffusion on COCO-2017.

4.3 Ablation and Sensitivity Analysis

We conduct ablation and sensitivity analyses on SD 1.5, covering the source of the improvements, the number of sampling steps, the CFG scale, and the weight of the marginal-alignment regularizer, with the latter two summarized in Figure 2. The effects of the hyperparameters involved in computing the marginal-alignment loss itself are reported in Section D.2.

Effect of marginal alignment. We verify in two ways that the improvements of the proposed method come from marginal alignment. First, we continue PeRFlow training from the same checkpoint for the same number of updates with both the LFM and SD 1.5 teachers, as reported in Sections E.1 and E.2, respectively. In both cases, unlike our method, continued PeRFlow training does not improve FID. Second, we estimate the reverse KL divergence between the student and

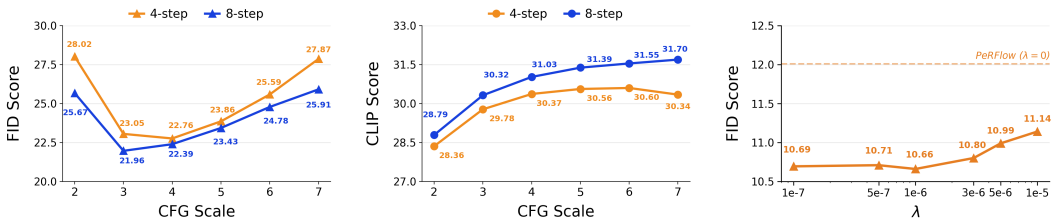


Figure 2: Sensitivity analysis on SD 1.5 (COCO-2017 validation). **Left and middle columns**: FID and CLIP score under 4-step and 8-step sampling across CFG scales $w \in [2, 7]$. **Right column**: FID score under 4-step sampling across λ values. The dashed line marks the PeRFlow baseline (λ=0).

teacher marginals at the four window endpoints on SD 1.5, as reported in Section D.1. Compared with PeRFlow, our method reduces this divergence at every endpoint.

Number of sampling steps. Section E.2 compares representative distilled SD 1.5 students under 2, 4, and 8 sampling steps on COCO-2017. Our method achieves the best FID and CLIP score under both 4-step and 8-step sampling, the step counts supported by its K=4 windows.

CFG sensitivity. CFG controls the balance between sample fidelity and text-image alignment during inference and can substantially affect the generation results of Stable Diffusion models. We sweep the guidance scale $w \in [2, 7]$ on COCO-2017 under both 4-step and 8-step sampling. The resulting curves exhibit the standard trade-off between FID and CLIP score, with stable behavior across the tested range. This pattern indicates that the marginal-alignment regularizer is compatible with standard CFG-based inference rather than relying on a narrowly tuned guidance scale.

λ sensitivity. The coefficient λ controls the relative strength of marginal alignment in the joint objective. The default value λ=10^{-6} is chosen according to the observed relative gradient magnitudes of the PeRFlow loss and the KL term. The right panel of Figure 2 further examines nearby choices of λ and shows improvements over the PeRFlow baseline for $\lambda \in [10^{-7}, 10^{-5}]$ under 4-step sampling, with the best result obtained at λ=10^{-6}. This sensitivity study supports the default scale used in the main experiments and suggests that the method does not require fine-grained tuning of λ.

5 Conclusion

In this work, we revisit reflow-based distillation from a distributional perspective. We show that trajectory matching, although effective for simplifying teacher ODE trajectories and improving the performance of few-step sampling, is insufficient to determine the distributions produced by the student when velocity matching is imperfect. This limitation motivates a complementary distribution-level signal. We therefore introduce a marginal-alignment regularizer that penalizes window-wise discrepancies from the corresponding teacher distributions without introducing auxiliary trainable networks. We further provide theoretical justification for this local-alignment strategy by connecting the window-wise penalties to the final-time distributional discrepancy. Empirically, the results across diffusion and flow-based settings indicate that marginal alignment is a useful complement to trajectory-based distillation for accelerating generative models.

Limitations. Although the marginal-alignment objective is compatible with other ODE-based reflow settings, including different noise schedules and the special case of K=1, empirical evaluation of these extensions, as well as of pretrained models with higher-dimensional latent spaces, is left to future work due to computational constraints. Extending the framework to students that do not define a velocity field, such as consistency models, would require a different mechanism for assessing distribution-level discrepancies.

Acknowledgments and Disclosure of Funding

This work is partially supported by Shenzhen Science and Technology Program (Grant No. AI2026019 & AI2026027). The authors declare no competing interests.

References

[1] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025.

[2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[4] Xu Cai, Yang Wu, Qianli Chen, Haoran Wu, Lichuan Xiang, and Hongkai Wen. Shortcutting pre-trained flow matching diffusion models is almost free lunch. In *Advances in Neural Information Processing Systems*, pages 153066–153094, 2025.

[5] Clément 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*, pages 15686–15695, 2025.

[6] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

[7] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

[8] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025.

[9] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, pages 52132–52152, 2023.

[10] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

[11] 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. In *Advances in Neural Information Processing Systems*, 2017.

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

[13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, pages 8633–8646, 2022.

[14] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

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

[16] Lei Ke, Haohang Xu, Xuefei Ning, Yu Li, Jiajun Li, Haoling Li, Yuxuan Lin, Dongsheng Jiang, Yujiu Yang, and Linfeng Zhang. ProReflow: Progressive reflow with decomposed velocity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28029–28038, 2025.

[17] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. BK-SDM: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pages 381–399, 2024.

[18] 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 *International Conference on Learning Representations*, 2024.

[19] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems*, pages 65484–65516, 2023.

[20] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, pages 21696–21707, 2021.

[21] 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. In *Advances in Neural Information Processing Systems*, pages 36652–36663, 2023.

[22] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. In *Advances in Neural Information Processing Systems*, pages 63082–63109, 2024.

[23] Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[24] 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, 2014.

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

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

[27] 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 *International Conference on Learning Representations*, 2024.

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

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

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, 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] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. LCM-LoRA: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023.

[33] 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. In *Advances in Neural Information Processing Systems*, pages 76525–76546, 2023.

[34] 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.

[35] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[36] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

[37] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *International Conference on Learning Representations*, 2023.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[39] 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. In *Advances in Neural Information Processing Systems*, pages 117340–117362, 2024.

[40] 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.

[41] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align Your Flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, pages 146459–146512, 2025.

[42] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[43] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

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

[45] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, pages 25278–25294, 2022.

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

[47] 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.

[48] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252, 2023.

[49] Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Xiaogang Wang, and Hongsheng Li. Phased consistency models. In *Advances in Neural Information Processing Systems*, pages 83951–84009, 2024.

[50] Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. In *International Conference on Learning Representations*, 2025.

[51] 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. In *Advances in Neural Information Processing Systems*, pages 8406–8441, 2023.

[52] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[53] Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by JKO scheme. In *Advances in Neural Information Processing Systems*, pages 47379–47405, 2023.

[54] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, pages 15903–15935, 2023.

[55] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. PeRFlow: Piecewise rectified flow as universal plug-and-play accelerator. In *Advances in Neural Information Processing Systems*, pages 78630–78652, 2024.

[56] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, pages 47455–47487, 2024.

[57] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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.

[58] 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.

A Proofs

A.1 Standing Assumptions

The following assumptions are used throughout the proofs. They ensure that the ODE systems are well posed and that the information-theoretic quantities in the analysis are finite.

Assumption 1 (Regularity)**.** *$\vec{f}_\theta, \vec{g}_\psi \in C^1(\mathbb{R}^d \times [0,1]; \mathbb{R}^d)$ are uniformly Lipschitz in x with at most linear growth. Consequently, the ODE solutions exist globally, are unique, and induce C^1-diffeomorphic flow maps Φ.*

Assumption 2 (Density regularity)**.** *All distributions appearing in the analysis admit positive C^1 densities with sufficient decay at infinity, so that integration by parts introduces no boundary terms.*

Assumption 3 (Finiteness)**.** *All KL divergences and expectations involved in the analysis are finite.*

A.2 Proof of Theorem 1 (Trajectory matching cannot guarantee marginal alignment)

We prove Theorem 1 by constructing a class of structured teachers and two students for each of them. As special cases, we then discuss teachers whose terminal distributions are Gaussian and non-Gaussian.

Teacher. Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be a C^2 map. For $t \in [0,1]$, define the interpolation map $\Psi_t(z) := (1-t)z + tF(z)$ and its Jacobian $J_t(z) := \nabla \Psi_t(z) = (1-t)I_d + t\nabla F(z)$. We assume that

(i) the singular values of $J_t(z)$ are bounded away from zero uniformly in (z,t);

(ii) ∇F and $z \mapsto (1+\|z\|)\nabla^2 F(z)$ are bounded.

By Hadamard's global inverse function theorem, condition (i) implies that Ψ_t is a C^1-diffeomorphism of $\mathbb{R}^d$ for every $t \in [0,1]$. Denote the inverse of Ψ_t by $u_t := \Psi_t^{-1}$. We then define the teacher velocity field

$$\vec{g}_\psi(x,t) := F(u_t(x)) - u_t(x). \tag{A.1}$$

Since $\partial_t \Psi_t(z) = F(z) - z = \vec{g}_\psi(\Psi_t(z), t)$, the flow map of $\vec{g}_\psi$ defined in Equation (2) is $\Phi_{\vec{g}_\psi,\, s\to t} = \Psi_t \circ u_s$ for $0 \le s \le t \le 1$. Setting $s = 0$ and noting that Ψ_0 is the identity, we obtain the teacher-induced marginal $q_{\psi,t} = (\Psi_t)_\# \pi_0$ for every $t \in [0,1]$.

Students. Fix a unit vector $e \in \mathbb{R}^d$, a magnitude $\varepsilon > 0$, and a C^1 function $c : [0,1] \to \mathbb{R}$ to be specified in Step 1. Define

$$\vec{f}_1(x,t) := \vec{g}_\psi(x,t) + \varepsilon\, c(t)\, J_t(u_t(x))\, e, \qquad \vec{f}_2(x,t) := \vec{g}_\psi(x,t) + \varepsilon\, \langle u_t(x), e\rangle\, J_t(u_t(x))\, e. \tag{A.2}$$

Under conditions (i) and (ii), the teacher and both students satisfy Assumption 1.

Step 1: Equal trajectory-matching losses. Fix a window $W_k = [t_{k-1}, t_k]$, let $X \sim \pi_{t_{k-1}}$, and set $U := u_{t_{k-1}}(X)$. The teacher trajectory starting from X at time t_{k-1} reaches $Y = \Phi_{\vec{g}_\psi,\, t_{k-1}\to t_k}(X) = \Psi_{t_k}(U)$ at time t_k. By the definition of Ψ_t, the interpolation and the target velocity in Equation (7) are $\Psi_\tau(U)$ and $F(U) - U = \vec{g}_\psi(\Psi_\tau(U), \tau)$, respectively. Hence, for any velocity field $\vec{f}$,

$$\mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}) = \mathbb{E} \int_{W_k} \left\| \vec{f}(\Psi_\tau(U), \tau) - \vec{g}_\psi(\Psi_\tau(U), \tau) \right\|^2 \mathrm{d}\tau. \tag{A.3}$$

Substituting $\vec{f}_1$ and $\vec{f}_2$ from Equation (A.2) and using $u_\tau(\Psi_\tau(U)) = U$, we obtain

$$\mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_1) = \varepsilon^2 \int_{W_k} c(\tau)^2\, w_k(\tau)\, \mathrm{d}\tau, \tag{A.4}$$

$$\mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_2) = \varepsilon^2 \rho_k, \tag{A.5}$$

where $w_k(\tau) := \mathbb{E}\|J_\tau(U)e\|^2$ and $\rho_k := \int_{W_k} \mathbb{E}\big[\langle U, e\rangle^2 \|J_\tau(U)e\|^2\big]\mathrm{d}\tau$. Both quantities are positive whenever $\langle U, e\rangle$ is not almost surely zero, since J_τ is invertible. Choose C^1 functions $\beta_k \ge 0$

supported in the interior of W_k and not identically zero, and set

$$c(t) := \sum_{k=1}^{K} \gamma_k \beta_k(t), \qquad \gamma_k := \left(\frac{\rho_k}{\int_{W_k} \beta_k(\tau)^2 w_k(\tau)\, \mathrm{d}\tau} \right)^{1/2}. \tag{A.6}$$

Then $\mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_1) = \mathcal{L}_{\mathrm{TM}}^{(k)}(\vec{f}_2)$ for every $k = 1, \ldots, K$.

Step 2: Different flow maps. For any velocity field $\vec{f} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ satisfying Assumption 1, define

$$\vec{h}(z,t) := J_t(z)^{-1} \big[\vec{f}(\Psi_t(z), t) - \vec{g}_\psi(\Psi_t(z), t) \big]. \tag{A.7}$$

Since $\partial_t \Psi_t(z) = \vec{g}_\psi(\Psi_t(z), t)$, the chain rule gives

$$\frac{\mathrm{d}}{\mathrm{d}t} \Psi_t\big(\Phi_{\vec{h},\, 0\to t}(z)\big) = \vec{f}\big(\Psi_t(\Phi_{\vec{h},\, 0\to t}(z)), t\big). \tag{A.8}$$

Because Ψ_0 is the identity, both sides start from z at $t = 0$, and the uniqueness of ODE solutions yields $\Phi_{\vec{f},\, 0\to t} = \Psi_t \circ \Phi_{\vec{h},\, 0\to t}$.

For $\vec{f}_1$, we have $\vec{h}_1(z,t) = \varepsilon c(t) e$, whose flow map is $\Phi_{\vec{h}_1,\, 0\to 1}(z) = z + \varepsilon m\, e$ with $m := \int_0^1 c(t)\, \mathrm{d}t > 0$. Hence

$$\Phi_{\vec{f}_1,\, 0\to 1}(z) = F\big(z + \varepsilon m\, e\big). \tag{A.9}$$

For $\vec{f}_2$, we have $\vec{h}_2(z,t) = \varepsilon \langle z, e \rangle e$, whose flow map is $\Phi_{\vec{h}_2,\, 0\to 1}(z) = z + (e^\varepsilon - 1)\langle z, e\rangle e$. Hence

$$\Phi_{\vec{f}_2,\, 0\to 1}(z) = F\big(z + (e^\varepsilon - 1)\langle z, e\rangle e\big). \tag{A.10}$$

Step 3: Different terminal marginals. Applying these maps to $\pi_0 = \mathcal{N}(0, I_d)$ gives

$$(\Phi_{\vec{f}_1,\, 0\to 1})_{\#}\pi_0 = F_{\#}\mathcal{N}(\varepsilon m\, e,\, I_d), \tag{A.11}$$

$$(\Phi_{\vec{f}_2,\, 0\to 1})_{\#}\pi_0 = F_{\#}\mathcal{N}\big(0,\, I_d + (e^{2\varepsilon} - 1) e e^\top\big). \tag{A.12}$$

The teacher-induced marginal is $q_{\psi,1} = (\Psi_1)_{\#}\pi_0 = F_{\#}\mathcal{N}(0, I_d)$. Since F is a bijection, the two student marginals differ from each other and from $q_{\psi,1}$.

Moreover, since KL divergence is invariant under bijections, the KL divergences of the two student marginals from $q_{\psi,1}$ are

$$\mathrm{KL}\left((\Phi_{\vec{f}_1,\, 0\to 1})_{\#}\pi_0 \,\|\, q_{\psi,1}\right) = \frac{\varepsilon^2 m^2}{2}, \tag{A.13}$$

$$\mathrm{KL}\left((\Phi_{\vec{f}_2,\, 0\to 1})_{\#}\pi_0 \,\|\, q_{\psi,1}\right) = \frac{1}{2}\big(e^{2\varepsilon} - 1 - 2\varepsilon\big). \tag{A.14}$$

The two values coincide only for particular choices of ε and m. □

Gaussian instance. Let $d \geq 2$ and $F(z) = z + \mu$ for some $\mu \neq 0$. Then $\vec{g}_\psi \equiv \mu$ and $J_t = I_d$, so conditions (i) and (ii) hold. With a unit vector e satisfying $\langle e, \mu \rangle = 0$ and $c \equiv 1$, the two students become $\vec{f}_1(x,t) = \mu + \varepsilon e$ and $\vec{f}_2(x,t) = \mu + \varepsilon \langle x, e\rangle e$. Under the data distribution $\pi_1 = \mathcal{N}(\mu, I_d)$ and the VP schedule in Section B.2, they have the same trajectory matching loss on every window, but their terminal marginals are $\mathcal{N}(\mu + \varepsilon e, I_d)$ and $\mathcal{N}(\mu, I_d + (e^{2\varepsilon} - 1) e e^\top)$, respectively.

Non-Gaussian instance. Let $d = 1$ and $F(z) = z + \alpha \tanh(\beta z)$ with $\alpha, \beta > 0$. Then $J_t(z) = 1 + t\alpha\beta\, \mathrm{sech}^2(\beta z) \geq 1$, and F'' decays exponentially, so conditions (i) and (ii) hold. In this case, the teacher-induced marginal $q_{\psi,1} = F_{\#}\mathcal{N}(0,1)$ is non-Gaussian. With $e = 1$ and $c(t) = \gamma_k$ on W_k, where γ_k is given by Step 1 with $\beta_k(t) = \mathbf{1}\{t \in W_k\}$, the two students are given by Equation (A.2). Under the data distribution $\pi_1 = q_{\psi,1}$ and the linear schedule $a(t) = 1 - t$, $b(t) = t$, they have the same trajectory matching loss on every window, but their terminal marginals are $F_{\#}\mathcal{N}(\varepsilon m, 1)$ and $F_{\#}\mathcal{N}(0, e^{2\varepsilon})$, respectively. Figure 1 visualizes this example for $\alpha = 3$, $\beta = 2$, $\varepsilon = 0.6$, and $K = 2$ windows split at $t = 0.5$.

A.3 Proof of Theorem 2 ($\mathcal{L}_{\mathrm{MA}}$ bounds the distributional gap)

We first establish a standard data-processing lemma for total variation.

Lemma 5 (Data processing inequality for TV)**.** *For any measurable map $T : \mathbb{R}^d \to \mathbb{R}^d$ and probability measures μ, ν,*

$$\mathrm{TV}(T_{\#}\mu, T_{\#}\nu) \le \mathrm{TV}(\mu, \nu). \tag{A.15}$$

Proof. For any Borel set A, $T_{\#}\mu(A) = \mu(T^{-1}(A))$. Since $\{T^{-1}(A) : A \in \mathcal{B}(\mathbb{R}^d)\} \subseteq \mathcal{B}(\mathbb{R}^d)$,

$$\begin{aligned}\mathrm{TV}(T_{\#}\mu, T_{\#}\nu) &= \sup_A |T_{\#}\mu(A) - T_{\#}\nu(A)| \\ &= \sup_A |\mu(T^{-1}(A)) - \nu(T^{-1}(A))| \le \mathrm{TV}(\mu, \nu).\end{aligned}$$

□

Proof of Theorem 2. Define $e_k := \mathrm{TV}(p_{\theta,t_k}, q_{\psi,t_k})$.

Base case. Since the teacher and student share the initial distribution π_0, we have $e_0 = \mathrm{TV}(\pi_0, q_{\psi,0}) = \mathrm{TV}(\pi_0, \pi_0) = 0$.

Inductive step. For $k \ge 1$,

$$e_k = \mathrm{TV}\Big((\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k})_{\#} p_{\theta,t_{k-1}},\ q_{\psi,t_k}\Big). \tag{A.16}$$

Introduce $\tilde{p}_{\theta^{(k)}} = (\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k})_{\#}\pi_{t_{k-1}}$. By the triangle inequality,

$$e_k \le \underbrace{\mathrm{TV}\Big((\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k})_{\#} p_{\theta,t_{k-1}},\ \tilde{p}_{\theta^{(k)}}\Big)}_{\text{(I)}} + \underbrace{\mathrm{TV}(\tilde{p}_{\theta^{(k)}},\ q_{\psi,t_k})}_{\text{(II)}}. \tag{A.17}$$

Bounding (I). By Lemma 5,

$$\text{(I)} \le \mathrm{TV}(p_{\theta,t_{k-1}}, \pi_{t_{k-1}}) \le \mathrm{TV}(p_{\theta,t_{k-1}}, q_{\psi,t_{k-1}}) + \mathrm{TV}(q_{\psi,t_{k-1}}, \pi_{t_{k-1}}) = e_{k-1} + \delta_{k-1}. \tag{A.18}$$

Bounding (II). By Pinsker's inequality,

$$\text{(II)} \le \sqrt{\tfrac{1}{2}\ \mathrm{KL}(\tilde{p}_{\theta^{(k)}} \| q_{\psi,t_k})} = \sqrt{\tfrac{1}{2}\,\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta)}. \tag{A.19}$$

Combining these two bounds gives

$$e_k \le e_{k-1} + \delta_{k-1} + \sqrt{\tfrac{1}{2}\,\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta)}. \tag{A.20}$$

Telescoping the recursion from $e_0 = 0$ yields

$$e_K \le \sum_{k=1}^{K} \sqrt{\tfrac{1}{2}\,\mathcal{L}_{\mathrm{MA}}^{(k)}(\theta)} + \sum_{k=0}^{K-1} \delta_k. \tag{A.21}$$

Since $\delta_0 = \mathrm{TV}(q_{\psi,0}, \pi_0) = 0$, the second sum equals $\sum_{k=1}^{K-1} \delta_k$. □

A.4 Proof of Lemma 3 (Liouville representation)

The Liouville representation rewrites the KL divergence $\mathcal{L}_{\mathrm{MA}}^{(k)}$ as an expectation over $\pi_{t_{k-1}}$, which can be estimated by Monte Carlo.

Proof. By Assumption 1, the flow map $\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k}$ is a C^1-diffeomorphism. The change-of-variables formula gives

$$\tilde{p}_{\theta^{(k)}}(\Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k}(x)) = \pi_{t_{k-1}}(x) \cdot \Big|\det \nabla_x \Phi_{\vec{f}_\theta,\, t_{k-1}\to t_k}(x)\Big|^{-1}. \tag{A.22}$$

Taking logarithms and applying Equation (4) gives

$$\log \tilde{p}_{\theta^{(k)}}\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)\right) = \log \pi_{t_{k-1}}(X) - \int_{t_{k-1}}^{t_k} \mathrm{div}_x \, \vec{f}_\theta\left(X^{\mathrm{Stu}}_{t_{k-1}\to \tau}(X;\theta),\tau\right) \mathrm{d}\tau$$
$$= \log \pi_{t_{k-1}}(X) + \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta). \tag{A.23}$$

Substituting this identity into the definition of KL yields

$$\mathcal{L}^{(k)}_{\mathrm{MA}}(\theta) = \mathbb{E}_{Y\sim\tilde{p}_{\theta^{(k)}}}\left[\log \tilde{p}_{\theta^{(k)}}(Y) - \log q_{\psi,t_k}(Y)\right]$$
$$= \mathbb{E}_{X\sim\pi_{t_{k-1}}}\left[\log \pi_{t_{k-1}}(X) + \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta) - \log q_{\psi,t_k}\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)\right)\right], \tag{A.24}$$

where the second line changes variables from $Y = X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ back to $X \sim \pi_{t_{k-1}}$. □

A.5 Proof of Theorem 4 (Gradient formula)

Proof. From Lemma 3, the term $\log \pi_{t_{k-1}}(X)$ is independent of θ. Hence,

$$\nabla_\theta \mathcal{L}^{(k)}_{\mathrm{MA}}(\theta) = \mathbb{E}_{X\sim\pi_{t_{k-1}}}\left[\nabla_\theta \Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta) - \nabla_\theta \log q_{\psi,t_k}\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)\right)\right]. \tag{A.25}$$

For the second term, the chain rule gives

$$\nabla_\theta \log q_{\psi,t_k}\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)\right) = s_\psi\left(X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta),\, t_k\right)^\top \nabla_\theta X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta). \tag{A.26}$$

The interchange of gradient and expectation follows from dominated convergence, since Assumption 1 ensures that $X^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ and $\Delta^{\mathrm{Stu}}_{t_{k-1}\to t_k}(X;\theta)$ have uniformly controlled θ-derivatives.

Since $s_\psi(\cdot, t_k)$ depends only on the fixed teacher parameters ψ and not on θ, the gradient does not propagate through s_ψ. This gives the stop-gradient form in Equation (16). □

B Diffusion–Flow Correspondence and Score Derivation

This section reviews the standard correspondence between diffusion models and continuous normalizing flows, covering arbitrary noise schedules and prediction parameterizations. These identities both provide the velocity-to-score conversion needed to evaluate the regularizer and establish that it applies to arbitrary noise schedules and prediction parameterizations. The correspondence reviewed below is well known and has been discussed by Kingma et al. [20], Karras et al. [15], Albergo et al. [1], and Kingma and Gao [19]. Throughout this section, θ denotes a generic network parameterization and does not necessarily refer to the student parameters used in the main text.

B.1 General Framework

Let $Z_0 \sim \mathcal{N}(0, I_d)$ and $Z_1 \sim q_1$ be independent, and let $(a(t), b(t))$ be a noise schedule satisfying $a(0) = 1, b(0) = 0, a(1) = 0, b(1) = 1$. We write $\dot{a}(t) := \frac{\mathrm{d}a(t)}{\mathrm{d}t}$ and $\dot{b}(t) := \frac{\mathrm{d}b(t)}{\mathrm{d}t}$. The interpolation $X_t = a(t)Z_0 + b(t)Z_1$ for $t \in [0, 1]$ induces a family of bridge distributions $\{p_t\}_{t\in[0,1]}$ between $\pi_0 = \mathcal{N}(0, I_d)$ and q_1, with

$$p_t(x) = \int \mathcal{N}(x;\, b(t)x_1,\, a(t)^2 I_d)\, q_1(x_1)\, \mathrm{d}x_1. \tag{B.1}$$

These marginals are induced by a CNF with velocity field

$$v^*(x,t) = \mathbb{E}[\dot{a}(t)Z_0 + \dot{b}(t)Z_1 \mid X_t = x]. \tag{B.2}$$

Velocity–score correspondence. Since $Z_0 = (X_t - b(t)Z_1)/a(t)$, substituting into Equation (B.2) and applying the Tweedie identity $\mathbb{E}[Z_1 \mid X_t = x] = \frac{1}{b(t)}(x + a(t)^2\nabla_x \log p_t(x))$ gives

$$v^*(x,t) = \frac{\dot{b}(t)}{b(t)}\, x + a(t)^2 \left(\frac{\dot{b}(t)}{b(t)} - \frac{\dot{a}(t)}{a(t)}\right) \nabla_x \log p_t(x). \tag{B.3}$$

Prediction parameterizations. Since $X_0 = Z_0$ and $X_1 = Z_1$, the velocity field can also be written as

$$v^*(x,t) = \dot{a}(t)\,\underbrace{\mathbb{E}[Z_0 \mid X_t = x]}_{\epsilon\text{-prediction}} + \dot{b}(t)\,\underbrace{\mathbb{E}[Z_1 \mid X_t = x]}_{x_0\text{-prediction}}. \tag{B.4}$$

Together with the identity $x = a(t)\mathbb{E}[Z_0 \mid X_t = x] + b(t)\mathbb{E}[Z_1 \mid X_t = x]$, any one of the velocity field, ϵ-prediction, x_0-prediction, and the score function determines the other three. The differences among flow matching, ϵ-prediction, x_0-prediction, and score matching therefore reduce to different weightings and reparameterizations of the same conditional expectation.

B.2 VP Schedule Specialization

SD 1.5 [40] and SDXL [36] both use ϵ-prediction with a VP noise schedule satisfying $a(t) = \sigma(t)$, $b(t) = \alpha(t)$, and $\alpha^2(t) + \sigma^2(t) = 1$, so that the forward process takes the form $z_t = \alpha(t)\, z_0 + \sigma(t)\, \epsilon$. Writing $\phi(t) = \arcsin(\sigma(t))$, so that $\alpha(t) = \cos(\phi(t))$ and $\sigma(t) = \sin(\phi(t))$, the general identities in Section B.1 specialize as follows.

Velocity field. From Equation (B.4), the velocity field corresponding to a network ε_θ that predicts $\mathbb{E}[Z_0 \mid X_t = x]$ is

$$V_\theta(z_t, t) = \frac{\dot{\phi}(t)}{\alpha(t)}\left[\varepsilon_\theta(z_t, t) - \sigma(t)\, z_t\right]. \tag{B.5}$$

Score function and x_0-prediction. From Equation (B.3), the score function is

$$s_\theta(z_t, t) = -\frac{\varepsilon_\theta(z_t, t)}{\sigma(t)}, \tag{B.6}$$

and the x_0-prediction is $X_\theta(z_t, t) = \frac{1}{\alpha(t)}[z_t - \sigma(t)\, \varepsilon_\theta(z_t, t)]$.

Backbone specifics. SD 1.5 and SDXL both use ϵ-prediction with the same VP schedule described above, differing only in backbone architecture and default resolution.

DDIM as a rescaled Euler step. The velocity field in Equation (B.5) can be further simplified by a change of variables. Define the rescaled state $\bar{z}_t = z_t/\alpha(t)$ and the rescaled time $\gamma(t) = \sigma(t)/\alpha(t) = \tan(\phi(t))$. Differentiating $\bar{z}_t$ gives

$$\mathrm{d}\bar{z}_t = \frac{1}{\alpha(t)}\left[V_\theta(z_t, t) + \dot{\phi}(t)\, \sigma(t)\, \bar{z}_t\right] \mathrm{d}t. \tag{B.7}$$

Substituting Equation (B.5) into Equation (B.7) and simplifying yields

$$\mathrm{d}\bar{z}_t = \varepsilon_\theta(z_t, t)\, \mathrm{d}\gamma, \tag{B.8}$$

in which the x_0-prediction term cancels and the dynamics are driven solely by the noise prediction ε_θ. A single Euler step of Equation (B.8) from $\gamma(r)$ to $\gamma(s)$, followed by the inverse rescaling $z_s = \alpha(s)\, \bar{z}_{\gamma(s)}$, recovers the standard DDIM update [46]:

$$z_s = \alpha(s)\, \frac{z_r - \sigma(r)\, \varepsilon_\theta(z_r, r)}{\alpha(r)} + \sigma(s)\, \varepsilon_\theta(z_r, r). \tag{B.9}$$

Therefore, using the DDIM sampler to compute the teacher endpoint $Y = \Phi_{\vec{g}_\psi,\, t_{k-1} \to t_k}(X)$ in the training procedure (Algorithm 1, line 6) amounts to solving the teacher ODE with an Euler step in the $(\bar{z}, \gamma)$ coordinate system.

C KL Estimator Details

This section provides implementation details for the marginal-alignment regularizer described in Section 3. The augmented ODE system and the teacher score derivation are given in Section 3 and Section B.2, respectively. Below we describe the divergence estimator and the ODE solver.

Hutchinson trace estimator with finite differences. We estimate the divergence $\mathrm{div}_x \vec{f}_\theta$ in the augmented ODE using the Hutchinson identity. For any random vector $\epsilon \in \mathbb{R}^d$ satisfying $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon\epsilon^\top] = I_d$,

$$\mathrm{div}_x \vec{f}_\theta(x,t) = \mathrm{tr}(\nabla_x \vec{f}_\theta(x,t)) = \mathbb{E}_\epsilon\Big[\epsilon^\top \nabla_x \vec{f}_\theta(x,t)\epsilon\Big]. \tag{C.1}$$

Following the finite-difference variant used in JKO-iFlow [53], we approximate the directional derivative by

$$\nabla_x \vec{f}_\theta(x,t)\epsilon \approx \frac{\vec{f}_\theta(x+\sigma\epsilon,t) - \vec{f}_\theta(x,t)}{\sigma}, \tag{C.2}$$

which yields the Monte Carlo estimator

$$\mathrm{div}_x \vec{f}_\theta(x,t) \approx \frac{1}{N_e}\sum_{i=1}^{N_e} \epsilon_i^\top \frac{\vec{f}_\theta(x+\sigma\epsilon_i,t) - \vec{f}_\theta(x,t)}{\sigma}. \tag{C.3}$$

Here $\epsilon_i \sim \mathcal{N}(0, I_d)$ are i.i.d. samples, and $\sigma = \sigma_0/\sqrt{d} > 0$ is the finite-difference scale. In our experiments, we use $N_e = 1$, with $\sigma_0 = 10$ for SD 1.5 and $\sigma_0 = 1$ for SDXL.

ODE solver. In each time window $W_k = [t_{k-1}, t_k]$, we solve the augmented ODE with a 4-step fixed-step-size RK4 solver and compute gradients via the adjoint method [6].

D Empirical Analysis of the Regularizer

This section analyzes the marginal-alignment regularizer, including its effect on the endpoint marginals, its sensitivity to the estimation hyperparameters, and its additional training cost.

D.1 Endpoint Marginal Discrepancy

For a prompt c drawn from the prompt distribution μ_c, let $p_{\theta,t_k}(\cdot \mid c)$ and $q_{\psi,t_k}(\cdot \mid c)$ denote the corresponding inference-time marginals of the student and the teacher at the window endpoint t_k. We measure their discrepancy by the prompt-averaged reverse KL divergence

$$D_k := \mathbb{E}_{c\sim\mu_c}\Big[\,\mathrm{KL}\left(p_{\theta,t_k}(\cdot \mid c) \,\big\|\, q_{\psi,t_k}(\cdot \mid c)\right)\Big], \tag{D.1}$$

and estimate it by

$$\widehat{D}_k := \frac{1}{N_c N_x}\sum_{i=1}^{N_c}\sum_{j=1}^{N_x}\Big[\log \widehat{p}_{\theta,t_k}\big(x_{ij}^{(k)} \mid c_i\big) - \log \widehat{q}_{\psi,t_k}\big(x_{ij}^{(k)} \mid c_i\big)\Big], \tag{D.2}$$

where $c_i \sim \mu_c$ and $x_{ij}^{(k)} \sim p_{\theta,t_k}(\cdot \mid c_i)$.

On SD 1.5, both log-densities are computed with Equation (4) by solving the augmented ODEs of the student and the teacher with eight RK4 steps per window, and the divergence terms are estimated as in Section C with $N_e = 5$ and $\sigma_0 = 10$. We use $N_c = 128$ prompts from the COCO-2014 validation set with $N_x = 5$ samples per prompt, and compute 95% confidence intervals for the difference between PeRFlow and MA-Reflow by paired bootstrap over prompts. As shown in Table D.1, MA-Reflow reduces $\widehat{D}_k$ at all four window endpoints, significantly at the 5% level.

Table D.1: Estimated reverse KL divergence $\widehat{D}_k$ between the student and teacher marginals at the window endpoints on SD 1.5 (lower is better).

Endpoint t_k	PeRFlow	MA-Reflow	Difference [95% CI]
0.25	17.97	**11.65**	6.32 [4.21, 8.31]
0.50	61.54	**42.41**	19.13 [10.51, 27.67]
0.75	118.99	**63.33**	55.66 [46.06, 65.23]
1.00	873.86	**362.47**	511.38 [473.87, 549.39]

D.2 Estimation Hyperparameters

On SD 1.5, we vary the number of Hutchinson samples N_e, the relative finite-difference scale σ_0, and the number of RK4 steps per window L one at a time around the default configuration $(N_e, \sigma_0, L) = (1, 10, 4)$. As shown in Table D.2, FID improves as N_e and L increase, consistent with lower Hutchinson variance and integration error, and remains stable around $\sigma_0 = 10$. Since larger N_e and L require more network evaluations per marginal-alignment update, we adopt the default configuration as a trade-off between training cost and generation quality.

Table D.2: 4-step COCO-2014 FID of SD 1.5 under different N_e, σ_0, and L.

N_e	1	3	5	10	
FID (↓)	10.66	10.49	10.28	10.15	
σ_0	0.1	0.5	1	10	25
FID (↓)	12.14	11.03	10.87	10.66	12.31
L	2	4	8		
FID (↓)	12.21	10.66	10.37		

D.3 Training Cost

We measure the training cost of SDXL distillation on a single NVIDIA H200 GPU with an effective batch size of 128 obtained through gradient accumulation. A PeRFlow update takes 60.31 s and a marginal-alignment update takes 629.97 s. With the marginal-alignment loss evaluated once every nine PeRFlow updates, Table D.3 reports the training time and peak memory over 2K iterations.

Table D.3: Training cost of SDXL distillation over 2K iterations on a single NVIDIA H200 GPU.

Method	Training time (h)	Peak memory (GiB)
PeRFlow	33.5	76.7
MA-Reflow	65.0	125.6

E Additional Distillation Results of MA-Reflow

E.1 LFM on CelebA-HQ

We conduct experiments on CelebA-HQ 256×256. The teacher is the pretrained LFM model of Dao et al. [7], a U-Net that directly predicts velocity, and the student adopts the same architecture and parameterization. We consider the following three training settings. As a reference, the pretrained teacher achieves an FID of 7.35 with its default 32-step Euler sampler.

Setting A: Training from teacher initialization. Starting from the pretrained LFM checkpoint, we train PeRFlow (K=4) and MA-Reflow (K=4) respectively for 100K iterations with identical hyperparameters, except that MA-Reflow additionally uses the marginal-alignment loss (λ=0.1, evaluated once every nine PeRFlow updates). Table E.1 reports the best FID achieved during training.

Table E.1: Best FID during training in Setting A.

Steps	LFM (teacher)	PeRFlow	MA-Reflow
4-step	54.41	18.17	**17.25**
8-step	21.29	**12.87**	13.04

Setting B: Fine-tuning from converged PeRFlow. Starting from the best PeRFlow checkpoint in Setting A, we continue training with MA-Reflow and, as a control, with PeRFlow for an additional 100K iterations. Table E.2 reports the best FID achieved during training. Continuing with PeRFlow degrades FID, whereas MA-Reflow improves it under both 4-step and 8-step sampling, suggesting that further gains at this checkpoint come from marginal alignment instead of trajectory matching.

Table E.2: Best FID during training in Setting B. "PeRFlow (init)" is the best PeRFlow checkpoint in Setting A, and "PeRFlow (cont.)" and MA-Reflow are both fine-tuned from it.

Steps	LFM (teacher)	PeRFlow (init)	PeRFlow (cont.)	MA-Reflow
4-step	54.41	18.17	23.87	**16.78**
8-step	21.29	12.87	13.85	**11.98**

Setting C: Training with two windows. To evaluate our method with fewer windows, we repeat Setting A with $K=2$. As shown in Table E.3, MA-Reflow outperforms PeRFlow at all step counts.

Table E.3: Best FID during training in Setting C.

Steps	LFM (teacher)	PeRFlow	MA-Reflow
2-step	107.21	24.72	**19.10**
4-step	54.41	20.98	**18.05**
8-step	21.29	14.79	**14.03**

Taken together, the three settings show that marginal-alignment regularization provides consistent benefits under different initialization conditions and window numbers, in line with the complementarity between trajectory matching and marginal alignment identified in Theorems 1 and 2.

E.2 Stable Diffusion 1.5

Continued PeRFlow training with equal updates. As in Setting B of Section E.1, we continue training PeRFlow for the same 5K iterations from the released checkpoint that initializes MA-Reflow. Table E.4 reports the best FID achieved during training. Continued PeRFlow training does not improve over the initialization, whereas MA-Reflow lowers FID under both 4-step and 8-step sampling.

Table E.4: Best FID on COCO-2014 during training. "PeRFlow (init)" is the shared initialization.

Steps	PeRFlow (init)	PeRFlow (cont.)	MA-Reflow
4-step	12.01	12.43	**10.66**
8-step	12.25	12.64	**10.78**

Number of sampling steps. We evaluated SD 1.5 students distilled by different representative methods with 2, 4, and 8 sampling steps, and the results on COCO-2017 are reported in Table E.5. PeRFlow and MA-Reflow use $K=4$ windows and thus require at least four steps. MA-Reflow achieves the best FID and CLIP score under both 4-step and 8-step sampling.

Table E.5: SD 1.5 results on COCO-2017 under different numbers of sampling steps.

Method	FID (↓)			CLIP (↑)		
	2-step	4-step	8-step	2-step	4-step	8-step
LCM-LoRA [32]	36.46	34.89	35.31	28.67	29.10	28.73
PCM [49]	31.88	24.15	23.01	29.65	30.11	30.45
TDM [34]	28.32	25.16	24.04	29.83	30.68	30.84
PeRFlow [55]	–	23.81	24.47	–	30.24	30.63
MA-Reflow (ours, $K=4$)	–	**22.76**	**22.35**	–	**31.03**	**31.55**

Qualitative comparison. Figure E.1 compares samples from the teacher and the distilled students.

E.3 Stable Diffusion XL

FID results. Table E.6 compares the FID of MA-Reflow and several representative methods on COCO-2014-10K and COCO-2017. MA-Reflow achieves the lowest FID on COCO-2014-10K and the second lowest on COCO-2017, close to Rectified Diffusion.

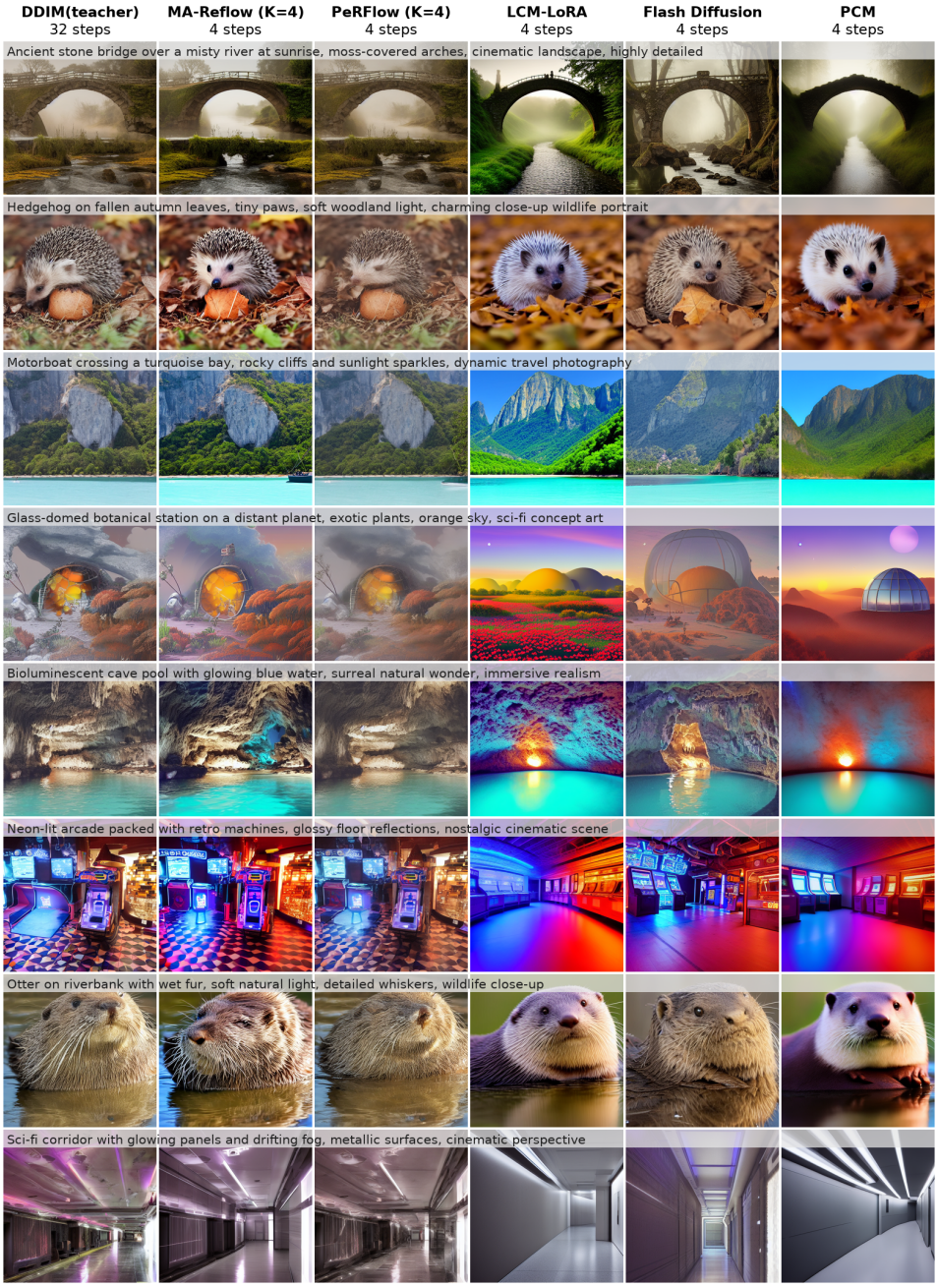


Figure E.1: Qualitative comparison on SD 1.5 under 4-step sampling. Each row shares the same prompt and initial noise. Our method achieves better visual quality and preserves richer fine-grained details than other distillation baselines.

Table E.6: SDXL results on COCO-2014-10K and COCO-2017 validation sets.

Method	Steps	COCO-2014-10K FID (↓)	COCO-2017 FID (↓)
SDXL-Lightning [23]	4	24.56	30.16
SDXL-Turbo [44]	4	23.19	30.92
HyperSD [39]	4	22.95	30.58
LCM [31]	4	22.16	27.09
PCM [49]	4	21.04	—
PeRFlow [55]	4	20.99	27.06
Rectified Diffusion [50]	4	19.71	**25.81**
DMD2 [56]	4	19.32	26.19
MA-Reflow (ours, $K=4$)	4	**19.17**	25.94

F Broader Impacts

This work studies efficient distillation methods for diffusion and flow-based generative models. Our experiments cover both unconditional image generation and conditional text-to-image synthesis, though the proposed approach may extend to other modalities such as audio and video. By reducing the number of sampling steps required for high-quality generation, our method can lower the computational cost and energy consumption of deploying these models. At the same time, more efficient generators could lower the barrier to producing disinformation or non-consensual imagery. This risk is shared by generative-model acceleration methods in general, and existing countermeasures such as output watermarking, provenance tracking, and synthetic-content detection remain fully applicable. All our experiments build on publicly available pretrained models and datasets.

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 abstract and introduction state the main theoretical and empirical claims, including the underdetermination of trajectory matching, the proposed marginal-alignment regularizer, the theoretical bound, and the experimental scope. These claims are supported by the results in Sections 3 and 4.

   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: The limitations are discussed in Section 5. In particular, empirical evaluation of other reflow settings (e.g., different noise schedules and $K=1$) and of pretrained models with higher-dimensional latent spaces is left to future work due to computational constraints, and extending the framework to student models that do not define a continuous-time velocity field (e.g., consistency models) would require a different mechanism for assessing distribution-level discrepancies.

   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: All theoretical results are stated in Section 3, with standing assumptions and complete proofs provided in Section A.

   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 the details necessary to reproduce our main experiments in Sections C and 4.1, covering model configurations, training hyperparameters, and hardware setup.

   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: [No]

   Justification: The code is being prepared and will be publicly released. The paper provides the details needed to reproduce the main results in Sections C and 4.1, and all datasets used in our experiments are publicly available.

   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: The experimental setup, including datasets, model configurations, training hyperparameters, and evaluation protocols, is described in Section 4.1 with additional details in Sections C and E.

   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: For the direct measurement of marginal alignment, we report 95% confidence intervals obtained by paired bootstrap over prompts (Section D.1). For FID and CLIP score, we do not report error bars because repeating large-scale text-to-image distillation runs is computationally expensive. Instead, we demonstrate robustness by evaluating across two model backbones (SD 1.5 and SDXL), two COCO validation sets, multiple CFG scales, and a range of regularization weights (Section 4.3).

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: The hardware, precision, and approximate training time are reported in Section 4.1, and the per-update time, end-to-end training time, and peak memory are reported in Section D.3.

   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 believe this work complies with it.

   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 discuss broader impacts in Section F.

    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: [N/A]

    Justification: This work improves the sampling efficiency of existing public models and does not introduce new misuse risks beyond those of the original models.

    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: All assets used in this work are cited and publicly available under open licenses.

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: This paper does not release new datasets or pretrained model weights.

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: This work does not involve 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: Not applicable.

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: LLMs were not used as a component of the core methodology.

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.