Stabilizing Consistency Training: A Flow Map Analysis and Self-Distillation

Anonymous authors
Paper under double-blind review

Abstract

Consistency models have been proposed for fast generative modeling, achieving results competitive with diffusion and flow models. However, these methods exhibit inherent instability and limited reproducibility when training from scratch, motivating subsequent work to explain and stabilize these issues. While these efforts have provided valuable insights, the explanations remain fragmented, and the theoretical relationships remain unclear. In this work, we provide a theoretical examination of consistency models by analyzing them from a flow map-based perspective. This joint analysis clarifies how training stability and convergence behavior can give rise to degenerate solutions. Building on these insights, we revisit self-distillation as a practical remedy for certain forms of suboptimal convergence and reformulate it to avoid excessive gradient norms for stable optimization. We demonstrate that our strategy extends beyond image generation to diffusion-based policy learning, without reliance on pretrained diffusion models for initialization, illustrating its broader applicability.

1 Introduction

Diffusion (Song & Ermon, 2019; 2020; Ho et al., 2020; Song et al., 2021) and flow matching models (Liu et al., 2023; Lipman et al., 2023) have achieved remarkable performance across a wide range of applications. This progress stems from flow-based modeling and multi-step inference, but is limited by multiple network evaluations during generation.

Consistency Models (Song et al., 2023; Frans et al., 2025; Geng et al., 2025a; Peng et al., 2025) have been proposed as few-step generation methods for both training-from-scratch and distillation settings. Earlier consistency training methods have often exhibited training instability and limited reproducibility in scratch settings, motivating subsequent work to stabilize training (Song & Dhariwal, 2024; Geng et al., 2025c; Lu & Song, 2025). While prior efforts have provided meaningful insights, the explanations remain fragmented, and the theoretical relationships among existing approaches remain unclear.

Prior work has explored distillation perspectives on consistency models from complementary viewpoints. Boffi et al. (2025b) provides a flow-map framework that organizes few-step distillation methods into Eulerian, Lagrangian, and semigroup-type formulations. This framework offers a unified perspective for distillation objectives, while also identifying objective-level discrepancies between training-from-scratch and distillation.

Since these discrepancies may steer training toward undesired solutions, self-distillation (Boffi et al., 2025a) recasts training-from-scratch methods through a distillation view. This reduces the objective-level gap by aligning the guiding velocity with the learned flow. However, Eulerian self-distillation is reported to produce large gradient norms in high-dimensional settings, and its direct use can still pose optimization challenges.

In this work, we systematically examine training instability and reproducibility in consistency models. We jointly analyze existing approaches through two training objectives, Eulerian distillation and consistency training. Along these objectives, we also explore key design factors: weight initialization and time condition.

This perspective clarifies how distinct objectives lead to different convergence behaviors. First, while prior work has identified mismatches between training objectives, it has not been formally established whether such differences necessarily imply different optima. We address this question by characterizing the optimality

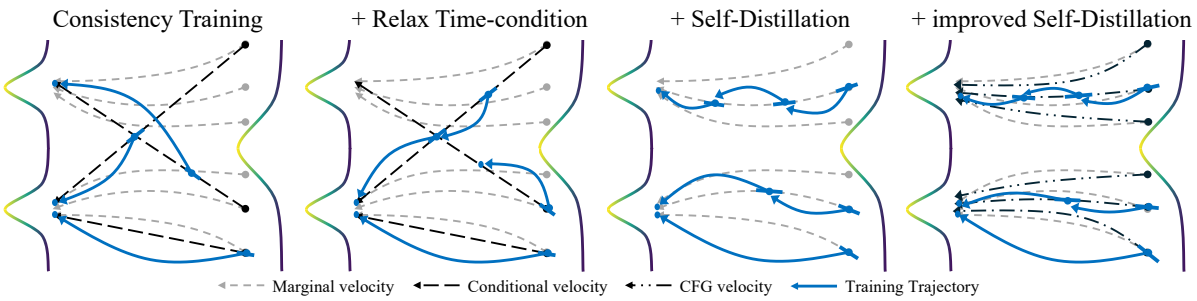


Figure 1: From consistency training to improved Self-Distillation. Consistency training learns a mapping over conditional velocity, often suffering from training instability and reproducibility issues. Relaxing the time condition mitigates this instability, and self-distillation aligns a target with the marginal velocity field. However, directly applying self-distillation leads to unstable training. We therefore reformulate the objective and incorporate classifier-free guidance, further stabilizing training and improving reproducibility.

conditions of Eulerian distillation. We show that training with conditional velocity can induce degenerate solutions, whereas training with marginal velocity can yield the desired optimum.

Second, while consistency training admits a fixed-point solution corresponding to the desired optimum, we show that this holds only at the expectation level. Under finite-batch optimization, the residual gap can perturb the fixed-point training dynamics, providing a possible explanation for the effect of small batch sizes.

Third, we examine complementary factors that affect which solution is reached in practice. The fixed-point structure contributes to sensitivity to initialization, undermining reproducibility, while the time condition affects the loss variance and spikes of the flow-map objective.

Motivated by the analysis, we revisit self-distillation as a practical mechanism for addressing certain forms of instability and reproducibility issues. Our analysis indicates that marginal velocity guidance can yield desired optima under these settings, while the relaxed time condition improves training stability. Self-distillation naturally provides this form of marginal guidance, but existing formulations are not directly compatible with consistency training due to the gradient norm problem in high-dimensional settings (Boffi et al., 2025a).

Building on the consistency-based reformulation introduced by Sabour et al. (2025), we reformulate the self-distillation objective to constrain gradient magnitudes while preserving its intended guidance. This enables stable optimization in practice and enables effective yet reproducible self-distillation in training-from-scratch settings. Finally, we show that this reformulated objective can be seamlessly combined with classifier-free guidance (Ho & Salimans, 2021). We provide an overview of our flow map perspective in Fig. 1.

We evaluate our approach on ImageNet-1K 256×256 in few-step generation. It exhibits stable behavior and achieves performance comparable to that of recent methods without relying on pretrained diffusion models for initialization. This supports the practical relevance of our theoretical insights. We extend our evaluation to diffusion-based policy learning, demonstrating applicability beyond the image generation domain.

2 Related Work

Diffusion and Flow Matching Models. Diffusion models (Ho et al., 2020; Song & Ermon, 2019) and flow matching models (Albergo & Vanden-Eijnden, 2023; Boffi et al., 2025b; Liu et al., 2023) are generative models that gradually transform a tractable noise distribution into the data distribution. These models have achieved remarkable progress in high-fidelity generation (Rombach et al., 2022; Podell et al., 2024; Esser et al., 2024). However, their reliance on a multi-step sampling requires substantial computational resources.

Few-step Generation. Several works have explored improving sampling efficiency of diffusion models (Salimans & Ho, 2022; Xiao et al., 2022; Rombach et al., 2022). These approaches aim to distill pretrained diffusion models into fewer-step generators, adopt GANs, or leverage VAEs to reduce input size. In parallel,

score distillation (Yin et al., 2024b;a; Zhou et al., 2024) has been proposed to construct one-step generators, while both rely on additional distillation stages or auxiliary networks, increasing training cost.

Consistency Models. Consistency Models (Song et al., 2023) are designed to predict a sample directly from any point along a flow trajectory. Both distillation and training methods have been proposed, whereas training from scratch can exhibit large loss variance, motivating earlier work on stabilization (Song & Dhariwal, 2024; Geng et al., 2025c; Lu & Song, 2025). Other studies (Boffi et al., 2025b; Issenhuth et al., 2025; Silvestri et al., 2025) have suggested that a discrepancy between distillation and training objectives can lead to high loss variance when training from scratch. Network-induced couplings (Issenhuth et al., 2025; Silvestri et al., 2025) are introduced to reduce loss variance, and self-distillation (Boffi et al., 2025a) distills shortcut mappings from its jointly learned flow to reduce this discrepancy. However, Eulerian self-distillation has been reported to encounter large gradient norms in high-dimensional settings.

Unified Framework. Recently, several studies have aimed to relate flow matching and consistency models. UCGM (Sun et al., 2025) integrates both paradigms, but does not account for the relaxed mapping constraint of arbitrary time points. Boffi et al. (2025b); Kim et al. (2024) present mathematical frameworks for consistency models, defining a model that learns *flow maps* as mappings between any two points on the same trajectory. While these works offer valuable theoretical perspectives, instability mechanisms arising from optimization noise, initialization, or time conditioning are not the primary focus of these analyses.

3 Preliminary

Flow Matching. Given a dataset X with underlying distribution p_X, flow matching models are trained to match the velocity fields of continuous flows, starting from a tractable distribution p_Z. Prior work constructs such flows via an interpolation, $x_t = \alpha_t x + \sigma_t z$, where $x \sim p_X$ and $z \sim p_Z$.

Let α_t and σ_t be continuous and monotone, with $\alpha_0 = \sigma_T = 1$ and $\alpha_T = \sigma_0 = 0$ for $t \in [0, T]$. A marginal distribution ρ_t induced by the flow satisfies $\rho_0 = p_X$ and $\rho_T = p_Z$. For a well-defined flow map, $\nu_t = \alpha_t \sigma_t' - \sigma_t \alpha_t' \neq 0$, and we further assume $\nu_t = \nu \neq 0$ for all $t \in [0, T]$, where ν is a constant. Notably, linear and trigonometric interpolations satisfy this with $\nu = 1$, which are our primary focus (see Sec. A.4).

With the constructed flow, the flow matching models optimize the squared error between the conditional velocity $v_t(x_t|x) = \alpha_t' x + \frac{\sigma_t'}{\sigma_t}(x_t - \alpha_t x)$ and a network $F_\theta(x_t; t)$:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{x,z,t}\left[\|F_\theta(x_t;t) - v_t(x_t|x)\|_2^2\right], \qquad F_\theta(x_t;t) \approx v_t^*(x_t) = \mathbb{E}_{x|x_t}[v_t(x_t|x)]. \tag{1}$$

Conditional flow matching $\mathcal{L}_{\text{CFM}}$ converges to the flow induced by the marginal velocity $v_t^*(x_t)$. If $v_t^*(x)$ is Lipschitz continuous in both t and x, the ODE $dx_t = v_t^*(x_t)dt$ has a unique solution (Lipman et al., 2023). However, flow models are known to suffer from mean collapse, in which one-step samples collapse to the mean of the data distribution (see Sec. A.1 and Karras et al. (2022)).

Flow Map. From flows defined by interpolation, our goal is to draw samples from the target distribution in a few sampling steps. To this end, we adopt a *flow map* $f(x_t; t, s)$, which is a mapping between two points x_t and x_s $(s < t)$ on the same trajectory (Kim et al., 2024; Boffi et al., 2025b):

$$f(x_t; t, s) = x_t + \int_t^s v_\tau^*(x_\tau) d\tau = x_s. \tag{2}$$

If the marginal velocity is Lipschitz continuous, the flow map is well-defined and injective (see Sec. A.2).

Training Flow Map. Since the flow map is defined as an integral, direct supervision from scratch is challenging. Recent studies adopt consistency training (Song et al., 2023) to avoid data generation, which can be derived from the Eulerian equation (see Sec. A.3 and Boffi et al. (2025b)):

$$\partial_t f(x_t; t, s) + v_t^*(x_t) \cdot \nabla_x f(x_t; t, s) = 0. \tag{3}$$

Note that if f is continuous in t and s, Lipschitz continuous in x, and satisfies the boundary condition $f(x_t; t, t) = x_t$, the flow map f is the unique solution to the Eulerian equation.

To train a flow map network $f_\theta(x_t; t, s)$ using this equation, Eulerian distillation (Boffi et al., 2025b) formulates the objective as a squared minimization problem:

$$\mathcal{L}_{\mathrm{ED}} = \mathbb{E}_{x,z,t,s}\left[\|\partial_t f_\theta(x_t; t, s) + v_t^*(x_t) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right]. \tag{4}$$

This is the basic objective for training flow maps. It reduces to consistency distillation when a fixed-point-style update is adopted, and s is fixed to zero (Song et al. (2023), see Sec. A.5):

$$\mathcal{L}_{\mathrm{CD}} = \mathbb{E}\left[\|f_\theta(x_t; t, s) - \mathrm{sg}\left[f_\theta(x_t; t, s) - [\partial_t f_\theta(x_t; t, s) + v_t^*(x_t) \cdot \nabla_x f_\theta(x_t; t, s)]\right]\|_2^2\right], \tag{5}$$

where $\mathrm{sg}[\cdot]$ is a stop-gradient operator. For consistency training from scratch, using $v_t(x_t|x)$ instead of $v_t^*(x_t)$ is a more common choice:

$$\mathcal{L}_{\mathrm{CT}} = \mathbb{E}\left[\|f_\theta(x_t; t, s) - \mathrm{sg}\left[f_\theta(x_t; t, s) - [\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)]\right]\|_2^2\right]. \tag{6}$$

Since $\mathcal{L}_{\mathrm{ED}}$ in Eq. (4) involves a Jacobian-vector product (JVP), its optimization requires second-order differentiation, introducing computational overhead. In contrast, consistency distillation $\mathcal{L}_{\mathrm{CD}}$ in Eq. (5) and consistency training $\mathcal{L}_{\mathrm{CT}}$ in Eq. (6) use a stop-gradient operation, avoiding this issue. However, they alter gradient dynamics, potentially affecting training stability (Prop. 4.3).

4 Analysis of Instability

In this section, we analyze instability and reproducibility in consistency models from a flow map perspective, focusing on how different design choices affect convergence properties and training dynamics. Based on this analysis, we explain why training from scratch may deviate from the desired flow map in practical settings.

4.1 Flow Map Representation for Analysis

To facilitate a joint analysis of recent consistency models, we adopt a flow map representation to express different training objectives in a common form. This representation serves as an analytical tool rather than a new assumption, and will be used throughout the paper to compare convergence behaviors and training dynamics across methods.

Flow Map Representation. To enable a systematic analysis of training instability in consistency models, we propose a flow map representation for commonly used interpolations, expressed as a one-step Euler solution with a network F_θ (see Sec. A.4):

$$f_\theta(x_t; t, s) = \nu^{-1}(A'_{t,s} x_t - A_{t,s} F_\theta(x_t; t, s)), \qquad A_{t,s} = \sigma_t \alpha_s - \sigma_s \alpha_t. \tag{7}$$

Under this representation, UCGM (Sun et al., 2025) can be viewed as a special case with fixed $s = 0$. TiM (Wang et al., 2026) also considers arbitrary (t, s), and it shares the functional structure under ν-assumption, while differing in parameterization.

With this representation, we propose a joint interpretation of recent consistency models (proof in Sec. A.5).

Proposition 4.1. *(Interpretation of Recent Methods) Recent consistency models can be interpreted within the flow map representation, satisfying the following transport equation:*

$$\partial_t f_\theta(x_t; t, s) + \tau_t(x_t, x) \cdot \nabla_x f_\theta(x_t; t, s) = 0, \tag{8}$$

where x_t is given by the Interpolant, τ_t by the Trajectory, and t, s by the Timestep, as summarized in Tab. 1.

Based on this interpretation, the objectives $\mathcal{L}_{\mathrm{ED}}$, $\mathcal{L}_{\mathrm{CD}}$, and $\mathcal{L}_{\mathrm{CT}}$ provide a common basis for analyzing the methods in Tab. 1. In the next subsection, we analyze these objectives and show how they can lead to certain training failures.

Table 1: Summary of recent consistency models interpreted under the flow map representation. $\Phi_t(x_t)$ is the teacher network and $F_\theta(x_t; t, s)$ is the training network. Our method is aligned with the learned marginal velocity $F_\theta(x_t; t, t)$ and the relaxed time condition $s < t$, enabled by a reformulation of Eulerian self-distillation (ESD) to improve training stability.

Model	Interpolant x_t	Trajectory τ_t	Timestep	JVP	Objective
Distillation-based Methods					
FMM-EMD (Boffi et al., 2025b)	Linear	$\Phi_t(x_t)$	$s < t$	Exact	$\mathcal{L}_{\text{ED}}$
AYF-EMD (Sabour et al., 2025)	Linear	$\Phi_t(x_t)$	$s < t$	Exact	$\mathcal{L}_{\text{CD}}$
sCD (Lu & Song, 2025)	Trigonometric	$\Phi_t(x_t)$	$s = 0$	Exact	$\mathcal{L}_{\text{CD}}$
Consistency Training Methods					
MeanFlow (Geng et al., 2025a)	Linear	$v_t(x_t\|x)$	$s < t$	Exact	$\mathcal{L}_{\text{CT}}$
ConsistencyFM (Yang et al., 2024)	Linear	$v_t(x_t\|x)$	$s = 0$	Approx.	$\mathcal{L}_{\text{CT}}$
sCT (Lu & Song, 2025)	Trigonometric	$v_t(x_t\|x)$	$s = 0$	Exact	$\mathcal{L}_{\text{CT}}$
UCGM (Sun et al., 2025)	Any	$v_t(x_t\|x)$	$s = 0$	Approx.	$\mathcal{L}_{\text{CT}}$
Self-Distillation Methods					
Shortcut Model (Frans et al., 2025)	Linear	$F_\theta(x_t; t, t)$	$s < t$	Approx.	$\mathcal{L}_{\text{SD}}$
ESD (Boffi et al., 2025a)	Linear	$F_\theta(x_t; t, t)$	$s < t$	Exact	$\mathcal{L}_{\text{SD}}$
iSD (Ours)	Any (ν-constrained)	$F_\theta(x_t; t, t)$	$s < t$	Approx.	$\mathcal{L}_{\text{SD-R}}$

4.2 Suboptimality and Instability

Guidance Velocity. Most consistency training approaches are guided by a *conditional velocity*. However, this setup can contribute to training instability. To examine its effect, we consider Eulerian distillation in Eq. (4) with a conditional velocity field. Following Boffi et al. (2025b), we refer to this as *direct training*.

$$\mathcal{L}_{\text{DT}} = \mathbb{E}\left[\|\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right]. \tag{9}$$

Since the global optimum of Eulerian distillation corresponds to the flow map, we show that direct training does not guarantee convergence to this solution by comparing their stationary conditions (proof in Sec. A.6).

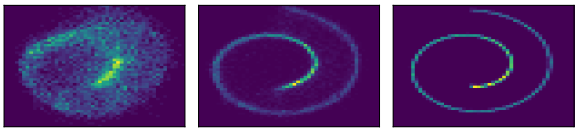
Proposition 4.2. *(Stationary condition of direct training) A stationary point of direct training satisfies*

$$\mathcal{S}_{\text{DT}} = \mathcal{S}_{\text{ED}} + C = 0, \quad C = \nabla \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_\theta), \tag{10}$$

where $\mathcal{S}_{\text{ED}}$ is the stationary condition of Eulerian distillation, $\Delta v = v_t(x_t|x) - v_t^*(x_t)$, and $\Sigma_{\Delta v|x_t}$ is the conditional covariance $\text{Cov}_{x|x_t}[\Delta v]$. This follows from applying the Euler-Lagrange equation to $\mathcal{L}_{\text{DT}}$. This suggests that training can result in a degenerate flow map ($\mathcal{S}_{\text{ED}} = -C \neq 0$), which is a suboptimal solution.

Thus, the discrepancy is not only an objective-level mismatch: through the term C, *it can actually shift the stationary point and lead to degenerate flow maps.* This term arises from the gap Δv between the marginal velocity $v_t^*(x_t)$ and the per-sample conditional velocity $v_t(x_t|x)$. We later show how this gap also affects stop-gradient objectives such as consistency training $\mathcal{L}_{\text{CT}}$ (Prop. 4.3).

To demonstrate this suboptimality of direct training, we conduct experiments on a toy dataset, as shown in Fig. 2. Compared to Eulerian distillation, which successfully learns the ground-truth distribution, direct training converges to a degenerate distribution, consistent with the proposition (see also Sec. C.3).



(a) $\mathcal{L}_{\text{DT}}$ (b) $\mathcal{L}_{\text{ED}}$ (c) Ground-truth

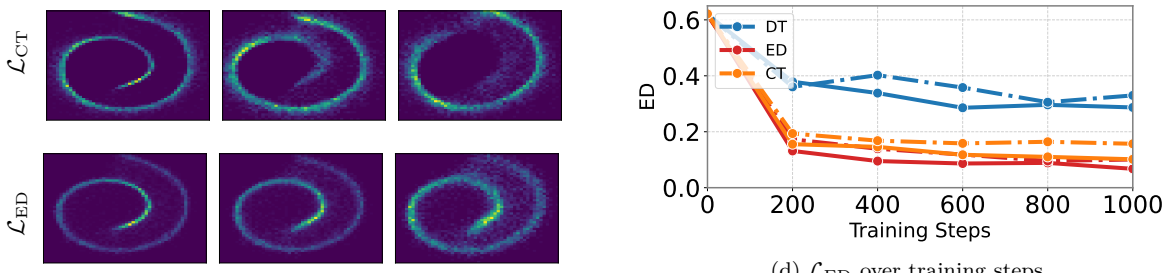
Figure 2: Toy experiments with a 5-layer MLP (batch size of 2048). $\mathcal{L}_{\text{DT}}$ drives flow map training toward a suboptimal solution, while $\mathcal{L}_{\text{ED}}$ leads to a solution close to the ground truth.

When a stop-gradient operation is applied to $\mathcal{L}_{\text{DT}}$, it coincides with the consistency training objective $\mathcal{L}_{\text{CT}}$ in Eq. (6). This objective can recover solutions satisfying the Eulerian equation, even when guided by conditional velocity (Song et al., 2023). However, this holds only at the expectation level, and a small batch size can induce a residual gap (see Sec. A.7).

Proposition 4.3. *(Expectation-level realization of consistency training) Let $L_* f_\theta = \partial_t f_\theta + v_t^* \cdot \nabla_x f_\theta$ denote the Eulerian operator in Eq. (3) applied to f_θ. The stationary condition of consistency training is*

$$\mathcal{S}_{\mathrm{CT}} = L_* f_\theta + \mathbb{E}_{x|x_t}[\Delta v \cdot \nabla_x f_\theta] = 0, \tag{11}$$

when $\rho_t > 0$. This admits the desired flow map as a fixed-point solution, $L_* f_\theta = 0$, since the second term vanishes by $\mathbb{E}_{x|x_t}[\Delta v] = 0$ in expectation. However, under finite-batch estimation, the empirical residual induced by Δv does not vanish in general, which can perturb the fixed-point training dynamics.



(a) $B = 2048$ (b) $B = 512$ (c) $B = 128$ (d) $\mathcal{L}_{\mathrm{ED}}$ over training steps.

Figure 3: (Left) Training results across batch sizes. The experiment follows Fig. 2. Consistency training $\mathcal{L}_{\mathrm{CT}}$ exhibits drift toward degenerate distributions when the batch size B decreases. (Right) Estimating the optimality of flow maps using $\mathcal{L}_{\mathrm{ED}}$. Solid lines indicate a batch size of 2048, and dash-dot lines indicate 128.

Fig. 3 shows *how this partial realization can lead $\mathcal{L}_{\mathrm{CT}}$ to a degenerate solution when batch size decreases.* We also measure $\mathcal{L}_{\mathrm{ED}}$ for the toy dataset in Fig. 3d, which serves as a proxy for estimating the optimality of flow maps. While it does not fully reflect generation performance, consistency training with small batch sizes and direct training yield higher $\mathcal{L}_{\mathrm{ED}}$ values, which can be interpreted as a signal of a degenerate solution. This helps explain why recent consistency models use large batch sizes.

From this perspective, both the bias term in Prop. 4.2 and the finite-batch residual in Prop. 4.3 originate from the same conditional-marginal velocity mismatch. We therefore consider incorporating the marginal velocity as the underlying flow field to reduce this gap in the next section.

Weight Initializations. We further observe that consistency models with $s = 0$ are sensitive to weight initialization, which can affect reproducibility. Recent approaches rely on initialization with pretrained flow models, where the network is already approximately aligned with target flows. This reduces certain early-stage instability, and is often referred to as a *preconditioner* (Geng et al., 2025a). We find that performance varies with the preconditioner, and reproducing results without known preconditioners can be challenging.

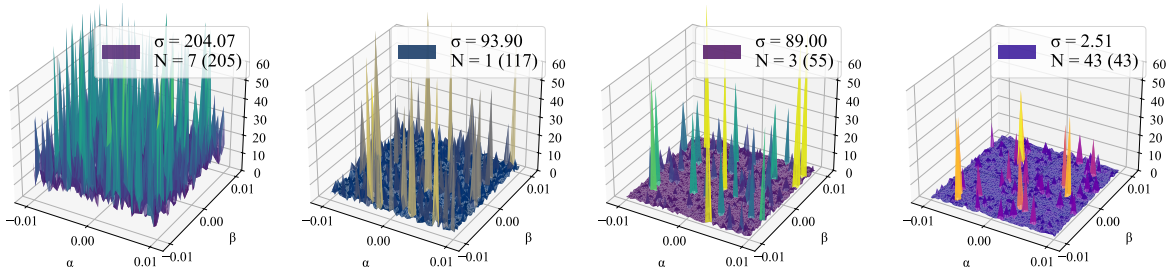
As shown in Tab. 2, we evaluated an open-source consistency model (Sun et al., 2025) on ImageNet-1K 256×256. With the baseline preconditioner, the model achieves reasonable results but remains above the reported FID. However, with other pretrained flow models, the FIDs become worse.

The resulting few-step FIDs are inconsistent: stronger preconditioners do not necessarily yield better consistency models. Additionally, models diverge when initialized randomly, suggesting that the optimum is inaccessible in this case.

Table 2: Consistency training results under different preconditioners. *Preconditioner FID* denotes the FIDs of pretrained networks for given ODE solver and sampling-step pairs. *2-FID* denotes two-step FIDs of consistency models initialized from the corresponding preconditioner (details in Sec. C.1).

Preconditioner	Preconditioner FID↓	2-FID↓
Multi-step Baseline	1.21 (UCGM-S, 30-step)	2.52
LightningDiT	2.17 (Euler, 250-step)	9.59
In-house Flow Model	2.41 (UCGM-S, 30-step)	5.78
w/o Preconditioner	-	419.60
Reported Baseline	1.21 (UCGM-S, 30-step)	1.42

As a complementary hypothesis, this can be understood from the stop-gradient (fixed-point) updates of consistency models: *initialization matters as training is driven toward nearby fixed points rather than the global optimum.* The result of Tab. 2 suggests the possible existence of attracting basins under different initialization weights. Without the known preconditioner, the training becomes difficult to reproduce.



(a) Consistency Model (b) Time Relaxation (c) Self-Distillation (ESD) (d) iSD (Ours)

Figure 4: Loss landscapes of four methods. The standard deviation σ and the number of spikes N serve as diagnostic proxies for training stability. iSD shows lower variance and fewer spikes under the common reference range. α and β denote the top-2 eigenvectors of the Hessian on ImageNet-1K with DiT-B/4. N denotes the number of samples outside each method's own empirical 95% range. Values in parentheses report the number of samples exceeding the range defined by iSD, as a common reference (details in Sec. C.6).

Time Condition. Some studies (Geng et al., 2025a; Frans et al., 2025) enable training from scratch without a preconditioner. The key difference is that they allow $s < t$, while others fix $s = 0$. Intuitively, training long-range mappings is more challenging than short-range ones. Under the flow map representation in Eq. (2), Eulerian distillation can be written, especially for linear interpolation, as

$$\mathcal{L}_{\text{ED}} = \mathbb{E}\left[\left\|A''_{t,s}x_t + A'_{t,s}(v^*_t - F_\theta) - A_{t,s}\frac{dF_\theta}{dt}\right\|_2^2\right] = \mathbb{E}\left[\left\|v^*_t - F_\theta - (t-s)\frac{dF_\theta}{dt}\right\|_2^2\right]. \tag{12}$$

We observe that $s \to t$ amplifies the flow matching term $v^*_t - F_\theta$ since $A''_{t,t} = A_{t,t} = 0$, while $s \to 0$ amplifies the linearization term involving the JVP dF_θ/dt. As the JVP is structurally more complex, we hypothesize that it may give rise to more undesired fixed points. This suggests that fixing $s = 0$ makes optimization less stable, while relaxing to $s < t$ balances the terms and mitigates instability (see Sec. A.8).

To support our hypothesis, we examine the loss landscapes of each objective, using their variance and number of spikes as proxies for training stability. As shown in Fig. 4, time relaxation reduces loss variance and spikes, which can be interpreted as reduced risk of training divergence, supporting our claim. We also note that our method in the next section, iSD, produces even smoother landscapes, resulting in more stable training.

Summary. Consistency training admits the desired flow map only in expectation due to conditional velocity guidance, and small batches can introduce a non-vanishing empirical residual. We observe that performance is sensitive to weight initialization, which can be attributed to fixed-point updates. Time-condition relaxation is associated with training stability and can enable random weight initialization, contributing to reproducibility.

These mechanisms provide an explanation for limited stability and reproducibility across initializations and training settings. Motivated by this observation, we seek to improve them by eliminating reliance on preconditioners through time-condition relaxation and reducing this residual via marginal velocity guidance.

5 Method

From Tab. 1, we identify that Eulerian Self-Distillation (ESD; Boffi et al., 2025a) uses marginal velocity under a relaxed time condition. This can make random initialization feasible and reduce reliance on preconditioners. However, ESD exhibits large gradient norms on high-dimensional data, destabilizing training.

Since ESD is not intended to explicitly address the training stability of existing consistency models, we build on the consistency-based formulation of Sabour et al. (2025) and reformulate self-distillation to be compatible with these models. This reformulation is designed to improve training stability and reproducibility. We further show that this can be seamlessly combined with classifier-free guidance, yielding additional performance gains. We call this approach *improved Self-Distillation (iSD)*.

5.1 improved Self-Distillation (iSD)

Eulerian Self-Distillation (ESD) jointly trains $F_\theta(x_t; t, t)$ via flow matching and $f_\theta(x_t; t, s)$ via Eulerian distillation, guided by its own velocity:

$$\mathcal{L}_{\text{SD}} = \mathbb{E}\left[\|\partial_t f_\theta(x_t; t, s) + F_{\theta^-}(x_t; t, t) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right]. \tag{13}$$

Compared to direct training, this utilizes the model-induced marginal velocity $v_t^*(x_t) \approx F_\theta(x_t; t, t)$ instead of the per-sample conditional velocity $v_t(x_t|x)$, without requiring teacher networks. This encourages convergence to the marginal flow map and addresses suboptimality in Prop. 4.2 (see Sec. B.1).

Reformulation. To enable ESD while incorporating design choices, we propose to reformulate self-distillation $\mathcal{L}_{\text{SD}}$ using the flow map representation (Eq. 7). In this case, the guidance velocity $v_\theta(x_t; t) = F_\theta(x_t; t, t)$ corresponds to the marginal velocity under joint training of flow matching:

$$\mathcal{L}_{\text{SD}} = \mathbb{E}\left[\left\|A''_{t,s}x_t + A'_{t,s}(v_\theta(x_t; t) - F_\theta(x_t; t, s)) - A_{t,s}\frac{dF_\theta(x_t; t, s)}{dt}\right\|_2^2\right]. \tag{14}$$

To reduce gradient norms, we consider two stop-gradient placements: only the spatial derivative and the full derivative. Both yield theoretically well-defined objectives admitting the desired fixed points, and we empirically select the latter for lower variance and computational efficiency (see Sec. A.7):

$$\mathcal{L}_{\text{SD-R}} = \mathbb{E}\left[\|F_\theta(x_t; t, s) - \text{sg}\left[F_{\text{tgt}}\right]\|_2^2\right], \quad F_{\text{tgt}} = F_\theta + A''_{t,s}x_t + A'_{t,s}(v_\theta - F_\theta) - A_{t,s}\frac{dF_\theta}{dt}. \tag{15}$$

Thus, our final objective of improved Self-distillation follows:

$$\mathcal{L}_{\text{iSD}} = \lambda_1\mathcal{L}_{\text{CFM}} + \lambda_2\mathcal{L}_{\text{SD-R}}, \qquad \mathcal{L}_{\text{CFM}} = \mathbb{E}[\|F_\theta(x_t; t, t) - v_t(x_t|x)\|_2^2]. \tag{16}$$

Compared to consistency models using the per-sample velocity for the flow-map target, iSD utilizes self-contained marginal velocity, which can reduce mismatch introduced in Prop. 4.3 and stabilize training by reducing loss variance (see Sec. B.2).

Computing JVP. To align with the marginal velocity, dF_θ/dt needs to follow $dF_\theta/dt = \partial_t F_\theta + v_\theta \cdot \nabla_x F_\theta$. This can be implemented using standard JVP APIs with inputs (x_t, t, s) and tangents $(v_\theta(x_t; t), 1, 0)$. This differs from prior work (Lu & Song, 2025; Geng et al., 2025a), which assumes tangents $(v_t(x_t|x), 1, 0)$.

Alternatively, we can approximate it using a finite difference. Given a small step ϵ, we define displaced inputs as $\hat{x}_{t\pm\epsilon} = x_t \pm \epsilon \cdot v_\theta$, and approximate $dF_\theta/dt \approx [F_\theta(\hat{x}_{t+\epsilon}; t + \epsilon, s) - F_\theta(\hat{x}_{t-\epsilon}; t - \epsilon, s)]/(2\epsilon)$.

Classifier-free Guidance. Classifier-free Guidance (CFG) is a widely used technique for improving diffusion models. However, since flow maps parameterize mappings rather than velocities, directly applying CFG does not guarantee mappings along the CFG velocity (see Sec. B.4).

We thus separate two extensions of iSD: Post-CFG and Pre-CFG. Post-CFG operates as a classical CFG:

$$\tilde{f}_\theta(x_t; t, s, c) = \nu^{-1}(A'_{t,s}x_t - A_{t,s}\tilde{F}_c), \quad \tilde{F}_c = F_\theta(x_t; t, s, \varnothing) + \omega(F_\theta(x_t; t, s, c) - F_\theta(x_t; t, s, \varnothing)) \tag{17}$$

where $\varnothing$ is the null class label, c is the class label, and ω is the guidance scale. Although this formulation is not guaranteed to follow the CFG velocity field, it can be readily applied after training.

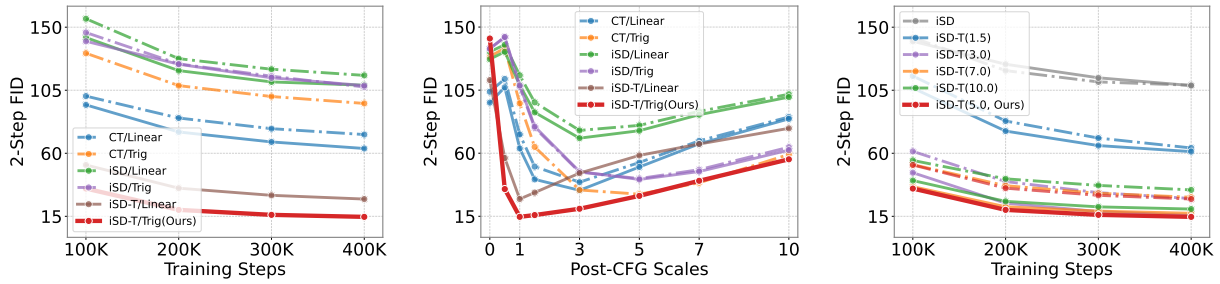
To ensure that the flow map follows the CFG field, we adopt a training-time CFG formulation (Geng et al., 2025a), Pre-CFG. This incorporates CFG directly by replacing flow matching with:

$$\mathcal{L}_{\text{CFM}}^{\omega} = \mathbb{E}\left[\|F_\theta(x_t; t, t, c) - \tilde{v}_t(x_t|x)\|_2^2\right], \qquad \tilde{v}_t(x_t|x) = F_\varnothing + \omega(v_t(x_t|x) - F_\varnothing). \tag{18}$$

We then train the flow map using the following objective:

$$\mathcal{L}_{\text{iSD-T}} = \lambda_1\mathcal{L}_{\text{CFM}}^{\omega} + \lambda_2\mathcal{L}_{\text{SD-R}}. \tag{19}$$

This objective encourages the flow map to align with the ground-truth CFG velocity (see Sec. A.9). We refer to this case as iSD-T (**T**raining-time CFG). We adopt iSD-T for class-conditional settings and vanilla iSD for class-unconditional settings.



(a) Design choices (b) Post-CFG scales (c) Pre-CFG scales

Figure 5: Design choices. (a) FIDs over training steps. Solid lines indicate the JVP approximation, and dash-dot lines indicate the exact JVP. (b) FIDs of Post-CFG over guidance scales. (c) FIDs of Pre-CFG over training steps. Solid lines indicate trigonometric interpolation, and dash-dot lines indicate linear.

6 Experiments

Experimental Settings. To evaluate our method on image generation, we conduct experiments on the ImageNet-1K (Deng et al., 2009) dataset. We use downsampled $32 \times 32 \times 4$ latent variables from 256×256 images encoded by a VAE (Rombach et al., 2022), and adopt a DiT (Peebles & Xie, 2023) in BF16 precision.

For unconditional settings, we validate our method on the CelebA-HQ (Karras et al., 2018) dataset, using the same latent variable settings as in the ImageNet experiments. Sample quality is measured with FID (Heusel et al., 2017) and Inception Score (IS; Salimans et al., 2016). More details are provided in Sec. C.2.

We further evaluate our method on diffusion-based policy learning to assess the applicability beyond image generation. We adopt diffusion policy benchmarks as in Chi et al. (2023), and report task success rates. This is intended to examine the applicability rather than to establish state-of-the-art results (see Sec. C.5).

6.1 Ablation Study

We conduct our ablation study on ImageNet-1K using DiT-B/4. The model is trained for 400K steps. We set $\omega = 5.0$ for $\mathcal{L}_{\text{iSD-T}}$ and use conditional velocity guidance for $\mathcal{L}_{\text{CT}}$. All weights are randomly initialized.

Table 3: Ablation study on ImageNet with DiT-B/4. 2-step evaluation results with standard deviations σ.

Case	FID↓ ($\pm\sigma$)	IS↑ ($\pm\sigma$)
Consistency Training	120.5 ($\pm$ 3.71)	9.69 ($\pm$ 0.47)
+ Joint training w/$\mathcal{L}_{\text{CFM}}$	96.28 ($\pm$ 15.9)	14.11 ($\pm$ 4.15)
+ Relaxing to $s < t$	69.18 ($\pm$ 2.84)	21.45 ($\pm$ 1.21)
+ iSD-T ($\omega = 5.0$)	26.91 ($\pm$ 0.50)	74.70 ($\pm$ 2.43)
+ Trigonometric Interp.	**14.62 ($\pm$ 0.06)**	**184.53 ($\pm$ 0.77)**

Key factors. To validate our analysis, we perform ablation experiments aimed at improving the stability and reproducibility of consistency training, as summarized in Tab. 3. We measure the variance of quantitative results across five runs with different random initializations, and report mean values.

We begin with consistency training $\mathcal{L}_{\text{CT}}$ using linear interpolation. Joint training with flow matching and relaxing time improve results and reduce variance. We observe that self-distillation alone converges more slowly (Tab. 4, Fig. 5a), while incorporating Pre-CFG $\mathcal{L}_{\text{iSD-T}}$ accelerates convergence and improves performance.

Interpolation, Post-CFG. As shown in Fig. 5 and Tab. 4, linear interpolation tends to perform better under conditional velocity guidance, whereas trigonometric one achieves a lower FID under self-distillation, particularly with Post-CFG. As performance varies across settings, we leave further analysis to future work.

Table 4: FIDs across design choices (2-Step). The numbers in the header denote Post-CFG scales.

Loss	Interp.	JVP	1.0	3.0	5.0	7.0
$\mathcal{L}_{\text{CT}}$	Linear	Exact	69.18	37.39	52.41	68.46
$\mathcal{L}_{\text{iSD}}$	Linear	Exact	114.56	74.17	78.20	88.48
$\mathcal{L}_{\text{CT}}$	Linear	Approx	63.02	33.40	50.18	67.27
$\mathcal{L}_{\text{iSD}}$	Linear	Approx	108.70	69.69	75.53	86.72
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	26.91	44.92	57.60	65.92
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	**14.62**	20.49	30.21	41.15

JVP operation. We observe that JVP approximation often yields better empirical results than exact computation, as illustrated in Fig. 5a. Under BF16 precision, it tends to be more robust, as exact one is sensitive to numerical precision. The VRAM requirement of the training decreases from 25GB to 18GB on iSD-T, and the training speed improves from 1.48 steps/sec to 4.04 steps/sec on an A100 GPU.

Pre-CFG. As shown in Fig. 5c and Tab. 4, incorporating Pre-CFG improves FIDs compared to the vanilla $\mathcal{L}_{\text{iSD}}$. It also outperforms Post-CFG, which deviates from the intended CFG field, whereas Pre-CFG aligns the flow map with the CFG velocity during training. Combining both often degrades performance, as it results in applying the guidance twice, overly skewing the trajectory.

Table 5: Comparison on ImageNet with DiT-B/4. 2-step results with standard deviations σ under a controlled setup: the backbone, data, optimization, and training budgets are fixed, while the only difference lies in the training objective. Each method is evaluated using the CFG scale reported as optimal in its work.

Case	**FID**$\downarrow$ ($\pm\sigma$)	**IS**$\uparrow$ ($\pm\sigma$)
MeanFlow ($\omega = 3.0$)	**11.48** ($\pm$ 1.68)	167.60 ($\pm$ 20.4)
Shortcut Model ($\omega = 1.5$)	87.52 ($\pm$ 24.3)	14.94 ($\pm$ 3.68)
FACM ($\omega = 1.75$)	25.52 ($\pm$ 1.73)	65.42 ($\pm$ 4.84)
iSD-T (Ours, $\omega = 5.0$)	14.62 ($\pm$ **0.06**)	**184.53** ($\pm$ **0.77**)

Reproducibility. Following Tab. 3 and Tab. 5, we examine the variance of quantitative results to assess reproducibility. Compared to prior work, our setting exhibits lower variance across runs while achieving comparable performance. This indicates that a main advantage of iSD-T is more reproducible training outcomes with comparable sample quality. See Sec. C.3 for further ablation studies including hyperparameters.

6.2 Comparison with Prior Work

Unconditional Generation. We compare our method with prior approaches on CelebA-HQ 256×256 under a class-unconditional setting, using vanilla iSD without CFG. As shown in Tab. 6, iSD achieves comparable performance in both few-step and multi-step generation.

Conditional Generation. We further compare consistency-based approaches on ImageNet-1K under class-conditional settings, as shown in Tab. 5. Using DiT-B/4, iSD achieves performance comparable to prior methods while exhibiting lower variance, indicating improved reproducibility.

To examine scalability, we evaluate iSD with DiT-XL. In Tab. 7, iSD maintains stable training behavior and comparable sample quality at this scale, without exhibiting the instability observed in prior methods.

Figure 6: 2-step results trained on ImageNet, iSD-T.

We note that achieving strong FID scores at this scale requires incorporating performance-oriented conditioning techniques, such as conditioning on a guidance scale or variations in latent representation. These choices are orthogonal to our analysis and are not required

Table 6: Comparison on Unconditional CelebA-HQ.

Model	4-FID (↓)	128-FID (↓)
Distillation-based Methods		
CD (Song et al., 2023)	39.6	59.5
Reflow (Liu et al., 2023)	18.4	16.1
Training from scratch		
Flow Matching (Lipman et al., 2023)	63.3	7.3
CT (Song et al., 2023)	19.0	53.7
Shortcut Model (Frans et al., 2025)	13.8	**6.9**
iSD (Ours)	**11.3**	7.1

Table 7: Comparison on ImageNet-1K. iSD-T uses linear interpolation with a guidance scale of 5.0.

METHOD	**NFE** (↓)	**FID** (↓)
iCT (Song & Dhariwal, 2024)	2	20.3
Shortcut Model (Frans et al., 2025)	1	10.6
UCGM (Sun et al., 2025)	1	2.10
IMM (Zhou et al., 2025)	2	7.77
MeanFlow (Geng et al., 2025a)	2	2.20
FACM (Peng et al., 2025)	2	1.32
iSD-T (Ours, SD-VAE)	2	2.76
iSD-T (Ours, VA-VAE)	2	2.26

for stability. Importantly, applying such conditioning does not reintroduce instability, indicating that iSD provides stable training even when combined with performance-oriented augmentations.

6.3 Diffusion-based Policy Learning

Table 8: Success rates (↑) of robotic control tasks.

Model	NFE (↓)	Push-T	Transport
Diffusion Policy (Chi et al., 2023)	100	0.95	1.00
	1	0.12	0.00
Shortcut Model (Frans et al., 2025)	1	0.87	0.80
MeanFlow (Geng et al., 2025a)	1	0.85	0.92
iSD (Ours)	2	0.92	1.00
	1	0.86	0.94

To assess the applicability of iSD beyond image generation, we adopt it as a diffusion policy objective in robotic control tasks. Experimental settings follow prior work on diffusion-based policies (Chi et al., 2023) for transformer and state-based architectures.

As shown in Tab. 8, iSD with linear interpolation yields performance comparable to other few-step methods, under the same evaluation protocol. With 2-step sampling, iSD achieves success rates comparable to those of diffusion policies that require more sampling steps. These results indicate that iSD can serve as a stable alternative to diffusion-based objectives in policy learning.

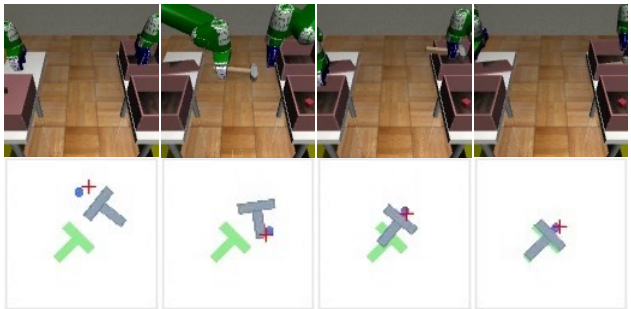


Figure 7: Simulation results of the iSD policy with 2-NFE on Transport (top) and Push-T (bottom) tasks.

7 Conclusion

In this work, we conducted a theoretical examination of consistency-based generative models from a flow map perspective, clarifying the sources of suboptimal convergence and training instability that limit reproducibility. This analysis explains how objective design choices influence convergence behavior in practice. Building on these insights, we revisited self-distillation and reformulated it to be compatible with consistency training, enabling more stable optimization without reliance on a pretrained diffusion model, namely a preconditioner. Finally, we demonstrated that the strategy is applicable to diffusion-based policy learning beyond image generation.

Broader Impact Statement

This paper presents work aimed at advancing the field of machine learning by improving the theoretical understanding and training stability of consistency-based generative models. The contributions focus on analysis and objective reformulation, without introducing new model architectures, expanding model capacity, or relying on additional data.

As such, the potential ethical and societal impacts of this work are aligned with those already well established in generative modeling, including concerns related to data bias, misuse of generated content, and downstream applications. Improving training stability and reproducibility may contribute positively to research efficiency and reduce computational waste, but the work does not directly address broader societal risks, which remain important considerations for future research.

References

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025b. ISSN 2835-8856.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

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

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models, 2025b.

Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers, 2020.

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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Thibaut Issenhuth, Sangchul Lee, Ludovic Dos Santos, Jean-Yves Franceschi, Chansoo Kim, and Alain Rakotomamonjy. Improving consistency models with generator-augmented flows, 2025.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

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

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable, 2025.

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

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. Flow-anchored consistency models, 2025.

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 *The Twelfth International Conference on Learning Representations*, 2024.

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 (CVPR)*, pp. 10684–10695, June 2022.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation, 2025.

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

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Gianluigi Silvestri, Luca Ambrogioni, Chieh-Hsin Lai, Yuhta Takida, and Yuki Mitsufuji. VCT: Training consistency models with variational noise coupling. In *Forty-second International Conference on Machine Learning*, 2025.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12438–12448. Curran Associates, Inc., 2020.

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.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

Peng Sun, Yi Jiang, and Tao Lin. Unified continuous generative models. *arXiv preprint arXiv:2505.07447*, 2025.

Zidong Wang, Yiyuan Zhang, Xiaoyu Yue, Xiangyu Yue, Yangguang Li, Wanli Ouyang, and Lei Bai. Transition models: Rethinking the generative learning objective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29178–29189, June 2026.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency, 2024.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *ICML workshop on Beyond First-Order Optimization Methods in Machine Learning*, 2020.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

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*, 2024b.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Forty-second International Conference on Machine Learning*, 2025.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

A Theoretical Analysis of Flow Map Models

A.1 Mean collapse of Diffusion and Flow Matching Models

Posterior Distribution. First, consider the data distribution $p_X(x) = \mathcal{N}(\mu_X, \sigma_X^2)$ and the interpolation $x_t = \alpha_t x + \sigma_t z$. The conditional distribution is given by $p(x_t = y|x) = \mathcal{N}(y; \alpha_t x, \sigma_t^2)$. By Bayes' rule,

$$\begin{aligned} p(x|x_t = y) &\propto p(x_t = y|x)p(x) \\ &= \mathcal{N}(y; \alpha_t x, \sigma_t^2)\mathcal{N}(x; \mu_X, \sigma_X^2) \\ &\propto \exp\left(-\frac{(y-\alpha_t x)^2}{2\sigma_t^2} - \frac{(x-\mu_X)^2}{2\sigma_X^2}\right) \\ &= \exp\left(-\left(\frac{1}{2\sigma_X^2} + \frac{\alpha_t^2}{2\sigma_t^2}\right)x^2 + \left(\frac{\mu_X}{\sigma_X^2} + \frac{\alpha_t y}{\sigma_t^2}\right)x - \left(\frac{\mu^2}{2\sigma_X^2} + \frac{y^2}{2\sigma_t^2}\right)\right). \end{aligned}$$

This can be organized as a Gaussian with a closed form $p(x|x_t = y) = \mathcal{N}(\mu_{x|y,t}, \sigma_{x|y,t}^2)$ where

$$\mu_{x|y,t} = \frac{\alpha_t \sigma_X^2 y + \mu_X \sigma_t^2}{\sigma_t^2 + \sigma_X^2 \alpha_t^2}, \quad \sigma_{x|y,t}^2 = \frac{\sigma_X^2 \sigma_t^2}{\sigma_t^2 + \sigma_X^2 \alpha_t^2}.$$

Extending the data distribution to a mixture of Gaussians $p_X(x) = \sum_i \pi_i \mathcal{N}(x; \mu_i, \sigma_i^2)$, we introduce the latent variable π for handling π_i:

$$p(\pi = i) = \pi_i, \quad p(x|\pi = i) = \mathcal{N}(x; \mu_i, \sigma_i^2).$$

Then, the marginal distribution $p(x_t = y|\pi = i)$ can be expressed as

$$\begin{aligned} p(x_t = y|\pi = i) &= \int p(x_t = y|x)p(x|\pi = i)dx \\ &= \int \mathcal{N}(y; \alpha_t x, \sigma_t^2)\mathcal{N}(x; \mu_i, \sigma_i^2)dx \\ &= \mathcal{N}(y; \alpha_t \mu_i, \alpha_t^2 \sigma_i^2 + \sigma_t^2). \end{aligned}$$

And we define responsibilities $r_i(y)$ as posterior distribution

$$r_{i,t}(y) = p(\pi = i|x_t = y) = \frac{p(x_t = y|\pi = i)p(\pi = i)}{\sum_j p(x_t = y|\pi = j)p(\pi = j)} = \frac{\pi_i \mathcal{N}(y; \alpha_t \mu_i, \alpha_t^2 \sigma_i^2 + \sigma_t^2)}{\sum_j \pi_j \mathcal{N}(y; \alpha_t \mu_j, \alpha_t^2 \sigma_j^2 + \sigma_t^2)}.$$

Therefore, the posterior distribution $p(x|x_t = y)$ is

$$\begin{aligned} p(x|x_t = y) &= \sum_i p(\pi = i|x_t = y)p(x|x_t = y, \pi = i) \\ &= \sum_i r_{i,t}(y)\mathcal{N}(x; \mu_{x|i,y,t}, \sigma_{x|i,y,t}^2) \\ &\text{where } \mu_{x|i,y,t} = \frac{\alpha_t \sigma_i^2 y + \mu_i \sigma_t^2}{\sigma_t^2 + \sigma_i^2 \alpha_t^2}, \ \sigma_{x|i,y,t}^2 = \frac{\sigma_i^2 \sigma_t^2}{\sigma_t^2 + \sigma_i^2 \alpha_t^2}. \end{aligned}$$

In particular, we observe that $\mu_{x|i,y,1} = \mu_i$, $\sigma_{x|i,y,1}^2 = \sigma_i^2$ and $r_{i,1}(y) = \pi_i$ when $\alpha_1 = 0$.

One-step Generation. Under the linear trajectory $x_t = (1-t)x + tz$, the conditional velocity is $v_t(x_t|x) = z - x$. Thus, one-step generation is defined by

$$\begin{aligned} f_F(x_t; t) &= x_t - tF^*(x_t; t) = x_t - t\,\mathbb{E}_{x|x_t}[v_t(x_t|x)] \\ &= \mathbb{E}_{x,z|x_t}\left[x_t - t(z - x)\right] \\ &= \mathbb{E}_{x|x_t}[x]. \end{aligned}$$

In the unimodal Gaussian case, $\mathbb{E}_{x|z}[x] = \mu_{x|z,1} = \mu_X$ and the one-step samples generated from $t = 1$ collapse to the mean of the data distribution. Similarly, in the mixture of Gaussians case, one-step generated samples collapse to the mixture mean.

$$f_F(z) = \mathbb{E}_{x|z}[x] = \sum_i r_i(z)\, \mu_{i|z,1} = \sum_i \pi_i\, \mu_i = \mu_X.$$

Thus, one-step generation collapses to the data mean μ_X regardless of the input. □

A.2 Injectivity of Flow Map

Since the marginal velocity is assumed to be Lipschitz continuous, the Picard-Lindelöf theorem guarantees a unique solution to the ODE $dx_t = v_t^*(x_t)dt$ for any initial value. The non-crossing property of trajectories follows directly, since any crossing at the same time would contradict uniqueness. Thus, since the flow map is formulated as the solution of the ODE initialized at x_t, it is well-defined.

For injectivity, suppose that $f(x; t, s) = f(y; t, s)$ for some x and y. By the uniqueness guaranteed by the Picard-Lindelöf theorem, the two trajectories must coincide along the interval. In particular, $f(x; t, t) = f(y; t, t)$. By the boundary condition of the flow map, $x = y$ and thus the flow map is injective. □

A.3 Eulerian Equation and Uniqueness of the Flow Map

Suppose the ground-truth flow map is defined as

$$f_{t,s}^*(x_t) = x_t + \int_t^s v_\tau^*(x_\tau)d\tau = x_s.$$

By construction, the identity mapping $f_{t,s}^*(f_{s,t}^*(x_s)) = x_s$ satisfies. Differentiating both sides w.r.t. t yields

$$\frac{d}{dt} f_{t,s}^*(f_{s,t}^*(x_s)) = \partial_t f_{t,s}^*(f_{s,t}^*(x_s)) + \partial_t f_{s,t}^*(x_s) \cdot \nabla_x f_{t,s}^*(f_{s,t}^*(x_s)) = \frac{d}{dt} x_s = 0.$$

Using $f_{s,t}^*(x_s) = x_t$ and $\partial_t f_{s,t}^*(x_s) = \partial_t x_t = v_t^*(x_t)$, we obtain the Eulerian equation:

$$\frac{d}{dt} f_{t,s}^*(x_t) = \partial_t f_{t,s}^*(x_t) + v_t^*(x_t) \cdot \nabla_x f_{t,s}^*(x_t) = 0.$$

Suppose a trainable network $f_\theta(x; t, s) = f_{t,s}^\theta(x)$ is sufficiently differentiable, continuous in x, t, s, Lipschitz continuous in x, and satisfies the boundary condition $f_{s,s}^\theta(x) = x$ for all s. If f_θ satisfies the Eulerian equation, $f_{t,s}^\theta$ remains constant along the characteristic curve induced by $v_t^*(x_t)$.

Let χ_τ denote the characteristic curve defined on $[s, t]$ by $\chi_t = x$ and $\chi'_\tau = v_\tau^*(\chi_\tau)$. Along this curve, $f_{\tau,s}^\theta(\chi_\tau)$ is constant and evaluating at $\tau = t$ and $\tau = s$ yields

$$f_{t,s}^\theta(x) = f_{t,s}^\theta(\chi_t) = f_{s,s}^\theta(\chi_s) = \chi_s = f_{t,s}^*(x),$$

since $f_{t,s}^*$ generates the characteristic curve χ_τ by its definition. Thus, the learned mapping coincides with the exact flow map. □

A.4 Flow Map Formulation and Interpolation Assumption

We begin by explicitly deriving the one-step Euler solution in Eq. (7). Supposing $x_t = \alpha_t x + \sigma_t z$, the velocity $v_t^*(x_t) = \mathbb{E}_{x,z|x_t}[\alpha'_t x + \sigma'_t z]$ gives:

$$\begin{aligned}
v_t^*(x_t) &= \alpha'_t \mathbb{E}_{x|x_t}[x] + \sigma'_t \mathbb{E}_{z|x_t}[z] \\
&= \alpha'_t \mathbb{E}_{x|x_t}[x] + \frac{\sigma'_t}{\sigma_t}(x_t - \alpha_t \mathbb{E}_{x|x_t}[x]), \\
\mathbb{E}_{x|x_t}[x] &= \left(\alpha'_t - \frac{\sigma'_t}{\sigma_t}\alpha_t\right)^{-1} \left(v_t^*(x_t) - \frac{\sigma'_t}{\sigma_t} x_t\right) \\
&= \frac{\sigma'_t x_t - \sigma_t v_t^*(x_t)}{\alpha_t \sigma'_t - \sigma_t \alpha'_t} = \nu_t^{-1}(\sigma'_t x_t - \sigma_t v_t^*(x_t)).
\end{aligned}$$

Similarly, the posterior mean of z is given by:

$$\begin{aligned} v_t^*(x_t) &= \frac{\alpha_t'}{\alpha_t}(x_t - \sigma_t \mathbb{E}_{z|x_t}[z]) + \sigma_t' \mathbb{E}_{z|x_t}[z], \\ \mathbb{E}_{z|x_t}[z] &= \left(\sigma_t' - \frac{\alpha_t'}{\alpha_t}\sigma_t\right)^{-1}\left(v_t^*(x_t) - \frac{\alpha_t'}{\alpha_t}x_t\right) \\ &= \frac{\alpha_t v_t^*(x_t) - \alpha_t' x_t}{\alpha_t \sigma_t' - \sigma_t \alpha_t'} = \nu_t^{-1}(\alpha_t v_t^*(x_t) - \alpha_t' x_t). \end{aligned}$$

Based on these, we can construct the conditional mean at time s:

$$\begin{aligned} \mathbb{E}_{x,z|x_t}[x_s] &= \nu_t^{-1}((\alpha_s\sigma_t' - \sigma_s\alpha_t')x_t - (\alpha_s\sigma_t - \sigma_s\alpha_t)v_t^*(x_t)) \\ &= \nu_t^{-1}(A_{t,s}' x_t - A_{t,s} v_t^*(x_t)), \end{aligned}$$

where $A_{t,s} = \alpha_s\sigma_t - \sigma_s\alpha_t$, $A_{t,s}' = \partial_t A_{t,s}$, and $A_{t,s}'' = \partial_t^2 A_{t,s}$. Prior work formulates a flow map f_θ with a pseudo-velocity network F_θ, given by

$$f_\theta(x_t; t, s) = \nu_t^{-1}(A_{t,s}' x_t - A_{t,s} F_\theta(x_t; t, s)).$$

Since it contains ν_t in its denominator, ν_t should be non-zero for an interval $t \in [0, T]$. Since $A_{t,t} = 0$ and $A_{t,t}' = \partial_t A_{t,s}|_{s=t} = \nu_t$, this formulation naturally induces the boundary condition $f_\theta(x_t; t, t) = x_t$.

When we assume $\alpha_t = \cos t$ and $\sigma_t = \sin t$ for trigonometric interpolation, $\nu_t = 1$, $A_{t,s} = \sin(t-s)$, and $A_{t,s}' = \cos(t-s)$. For linear interpolation $\alpha_t = 1 - t$ and $\sigma_t = t$, each term reduces to $\nu_t = 1$, $A_{t,s} = t - s$, and $A_{t,s}' = 1$. Based on these settings, prior work formulates flow maps as $f_\theta(x_t; t, s) = \cos(t-s)x_t - \sin(t-s)F_\theta(x_t; t, s)$ for trigonometric interpolation (Lu & Song, 2025), and $f_\theta(x_t; t, s) = x_t - (t-s)F_\theta(x_t; t, s)$ for linear interpolation (Geng et al., 2025a; Yang et al., 2024; Sabour et al., 2025).

We observe that assuming $\nu_t = \nu \neq 0$ for all $t \in [0, T]$ gives $A_{t,t}' = \partial_t A_{t,s}|_{s=t} = \nu$ and $A_{t,t}'' = \partial_t^2 A_{t,s}|_{s=t} = 0$, when ν is a constant. This induces the important property of vanishing linearization, which helps proofs in Sec. A.8 and Sec. B.1. Additionally, this assumption can simplify differentiation of the flow map formulation, which is required in the consistency training objective. For these reasons, we assume the *ν-condition*. As we mentioned, both linear interpolation on $[0, 1]$ and trigonometric interpolation on $[0, \pi/2]$ satisfy $\nu = 1$.

Possible ν-constrained interpolations. Suppose that γ_t is monotonically increasing over $t \in [0, 1]$ and satisfies the boundary conditions $\gamma_1 = 1$ and $\gamma_0 = 0$. Consider the interpolation defined by $\alpha_t = (1-\gamma_t)^c$ and $\sigma_t = \gamma_t^c$ for some constant $c \in [0.5, 1]$. Then, ν_t can be written as $\nu_t = c(1-\gamma_t)^{c-1}\gamma_t^{c-1}\gamma_t'$. Imposing $\nu_t = \nu$ gives $\gamma_t' = \nu[c(1-\gamma_t)^{c-1}\gamma_t^{c-1}]^{-1}$ and $c(1-\gamma)^{c-1}\gamma^{c-1}d\gamma = \nu dt$. Integrating both sides yields

$$c\int (1-\gamma)^{c-1}\gamma^{c-1}d\gamma = \nu\int dt = \nu(t + C),$$

where the constant C vanishes due to $\gamma_0 = 0$. For the incomplete beta function B, this becomes

$$cB(\gamma_t; c, c) = c\int_0^{\gamma_t}(1-\eta)^{c-1}\eta^{c-1}d\eta = \nu\int_0^t d\tau = \nu t.$$

By the boundary condition, $cB(c, c) = cB(1; c, c) = \nu$, and thus γ_t is characterized by

$$cB(\gamma_t; c, c) = \nu\frac{cB(\gamma_t; c, c)}{cB(c, c)} = \nu I_{\gamma_t}(c, c) = \nu t \implies \gamma_t = I_t^{-1}(c, c),$$

where I denotes the regularized incomplete beta function.

In this case, γ_t is characterized regardless of ν, and *this is a possible generalization of interpolations satisfying ν-assumption.* In particular, when $c = 0.5$,

$$B\left(\gamma; \frac{1}{2}, \frac{1}{2}\right) = \int_0^\gamma \frac{d\eta}{\sqrt{\eta(1-\eta)}} = 2\arcsin\sqrt{\gamma} \implies \gamma_t = \sin^2\left(\frac{\pi}{2}t\right),$$

which yields trigonometric interpolation on the interval $[0, 1]$. In this case, the corresponding constant is $\nu = \pi/2$. On the other hand, when $c = 1$, we have $\gamma_t = t$, $\alpha_t = 1 - t$, $\sigma_t = t$, which reduces to linear interpolation. Interpolating c between 0.5 and 1.0 is a promising direction for exploring interpolations.

A.5 Recent Consistency-based Generative Models are Flow Map Models

sCT. Under trigonometric interpolation $x_t = \cos(t)x + \sin(t)z$,

$$f_\theta(x_t; t, s) = \cos(t-s)x_t - \sin(t-s)F_\theta(x_t; t, s).$$

When $s = 0$

$$f_\theta(x_t; t) = \cos(t)x_t - \sin(t)F_\theta(x_t; t),$$

which exactly recovers the sCT formulation. If consistency training is formulated without the stop-gradient operation, then the objective reduces to the direct training objective $\mathcal{L}_{\mathrm{DT}}$ in Eq. (9) as $\Delta t \to 0$.

$$\begin{aligned}
&\frac{1}{\Delta t^2} \cdot \mathbb{E}\left[\|f_\theta(x_t; t, s) - f_\theta(x_{t-\Delta t}; t-\Delta t, s)\|_2^2\right] \\
&= \frac{1}{\Delta t^2} \cdot \mathbb{E}\left[\left\|f_\theta(x_t; t, s) - \left[f_\theta(x_t; t, s) - \partial_t f_\theta(x_t; t, s) \cdot \Delta t - v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s) \cdot \Delta t + O(\Delta t^2)\right]\right\|_2^2\right] \\
&= \frac{1}{\Delta t^2}\mathbb{E}\left[\left\|(\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)) \cdot \Delta t + O(\Delta t^2)\right\|_2^2\right] \\
&= \mathbb{E}\left[\|\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right] + O(\Delta t) \\
&= \mathcal{L}_{\mathrm{DT}} + O(\Delta t).
\end{aligned}$$

When we use the stop-gradient, the continuous-time consistency training objective is given by

$$\begin{aligned}
&\frac{1}{2\Delta t} \cdot \nabla_\theta \mathbb{E}\left[\|f_\theta(x_t; t, s) - f_{\theta^-}(x_{t-\Delta t}; t-\Delta t, s)\|_2^2\right] \\
&= \frac{1}{\Delta t} \cdot \mathbb{E}\left[\nabla_\theta f_\theta(x_t; t, s) \cdot (f_{\theta^-}(x_t; t, s) - f_{\theta^-}(x_{t-\Delta t}; t-\Delta t, s))\right] \\
&= \nabla_\theta \mathbb{E}\left[f_\theta(x_t; t, s) \cdot \frac{f_{\theta^-}(x_t; t, s) - f_{\theta^-}(x_{t-\Delta t}; t-\Delta t, s)}{\Delta t}\right] \\
&\to \nabla_\theta \mathbb{E}\left[f_\theta(x_t; t, s)\frac{df_{\theta^-}(x_t; t, s)}{dt}\right] = \nabla_\theta \mathcal{L}_{\mathrm{CT}}
\end{aligned}$$

as $\Delta t \to 0$, where $\theta^- = \mathrm{sg}[\theta]$ for the stop-gradient operation $\mathrm{sg}[\cdot]$. Using $\nabla_\theta f_\theta^T y = \frac{1}{2}\nabla_\theta \|f_\theta - \mathrm{sg}[f_\theta - y]\|_2^2$, the practical implementation of this objective is given by

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\mathrm{CT}} &= \nabla_\theta \mathbb{E}\left[f_\theta(x_t; t, s)\frac{df_{\theta^-}(x_t; t, s)}{dt}\right] \\
&= \nabla_\theta \mathbb{E}\left[\frac{1}{2}\left\|f_\theta(x_t; t, s) - f_{\theta^-}(x_t; t, s) + \frac{df_{\theta^-}(x_t; t, s)}{dt}\right\|_2^2\right] \\
&= \nabla_\theta \mathbb{E}\left[\frac{1}{2}\left\|f_\theta(x_t; t, s) - \mathrm{sg}\left[f_\theta(x_t; t, s) - [\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)]\right]\right\|_2^2\right],
\end{aligned}$$

which coincides with Eq. (6), since $df_\theta/dt = \partial_t f_\theta + v_t \cdot \nabla_x f_\theta$. We note that the interior of the formulation can be interpreted as a fixed-point-style update, $f_\theta \leftarrow f_\theta - df_\theta/dt$, of the direct training objective. Similarly, consistency distillation can be interpreted as a fixed-point-style update of Eulerian distillation.

From another perspective, since the time derivative of $f_\theta(x_t; t, s)$ is given by

$$\begin{aligned}
\frac{df_\theta(x_t; t, s)}{dt} &= \partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s) \\
&= L_* f_\theta(x_t; t, s) + \Delta v \cdot \nabla_x f_\theta(x_t; t, s),
\end{aligned}$$

where $\Delta v = v_t(x_t|x) - v_t^*(x_t)$ and $L_* f_\theta = \partial_t f_\theta + v_t^* \cdot \nabla_x f_\theta$, the objective can be written as

$$\mathcal{L}_{\mathrm{CT}} = \mathbb{E}\left[f_\theta \cdot (L_* f_{\theta^-}) + f_\theta \cdot (\Delta v \cdot \nabla_x f_{\theta^-})\right].$$

Here and below, we omit x_t, t, and s for brevity. The first term on the right-hand side corresponds to consistency distillation. By the tower property, the second term vanishes under conditional expectation:

$$\begin{aligned}\mathbb{E}\left[f_\theta \cdot (\Delta v \cdot \nabla_x f_{\theta^-})\right] &= \mathbb{E}\left[\mathbb{E}_{x|x_t}\left[f_\theta \cdot (\Delta v \cdot \nabla_x f_{\theta^-})\right]\right] \\ &= \mathbb{E}\left[f_\theta \cdot \left(\mathbb{E}_{x|x_t}\left[\Delta v\right] \cdot \nabla_x f_{\theta^-}\right)\right] \\ &= \mathbb{E}\left[f_\theta \cdot (0 \cdot \nabla_x f_{\theta^-})\right] \\ &= 0.\end{aligned}$$

Thus, $\mathcal{L}_{\mathrm{CT}}$ reduces to consistency distillation at the expectation level, even along the conditional trajectory. The instability of these gradient dynamics is discussed in Sec. A.7. □

MeanFlow. Suppose a flow map model under linear interpolation $x_t = (1-t)x + tz$:

$$f_\theta(x_t; t, s) = x_t + (s-t)F_\theta(x_t; t, s).$$

The corresponding direct training objective is

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right] = \mathbb{E}\left[\left\|\frac{df_\theta(x_t; t, s)}{dt}\right\|_2^2\right],$$

where

$$\frac{d}{dt} f_\theta(x_t; t, s) = v_t(x_t|x) - F_\theta(x_t; t, s) + (s-t)\frac{d}{dt}F_\theta(x_t; t, s).$$

Recall the MeanFlow objective from Geng et al. (2025a)

$$\mathcal{L}(\theta) = \mathbb{E}[\|u_\theta(z_t; r, t) - \mathrm{sg}[v_t - (t-r)(v_t \cdot \partial_z u_\theta + \partial_t u_\theta)]\|_2^2]. \tag{20}$$

Rewriting the MeanFlow objective using flow map notation gives

$$\begin{aligned}&\nabla_\theta \mathbb{E}\left[\|F_\theta(x_t; t, s) - \mathrm{sg}[v_t(x_t|x) - (t-s)(v_t(x_t|x) \cdot \nabla_x F_\theta + \partial_t F_\theta)]\|_2^2\right] \\ &= \nabla_\theta \mathbb{E}\left[\left\|F_\theta(x_t; t, s) - v_t(x_t|x) + (t-s) \cdot \frac{d}{dt}F_{\theta^-}(x_t; t, s)\right\|_2^2\right] \\ &= \nabla_\theta \mathbb{E}\left[\left\|F_\theta(x_t; t, s) - F_{\theta^-}(x_t; t, s) - \left[v_t(x_t|x) - F_{\theta^-}(x_t; t, s) + (s-t)\frac{d}{dt}F_{\theta^-}(x_t; t, s)\right]\right\|_2^2\right] \\ &= \nabla_\theta \mathbb{E}\left[\frac{1}{2(t-s)} f_\theta(x_t; t, s)\frac{df_{\theta^-}(x_t; t, s)}{dt}\right].\end{aligned}$$

Thus, the MeanFlow objective is a weighted instance of the continuous-time consistency training under linear interpolation. We note that their ideal MeanFlow identity coincides with Eulerian distillation in Eq. (4). However, their practical implementation in Eq. (20) uses stop-gradient and conditional velocity, rather than marginal velocity, therefore it reduces to the continuous-time consistency training objective. □

Shortcut Model. From Frans et al. (2025), the Shortcut Model objective consists of the flow matching objective and the consistency objective.

$$\mathcal{L}_{\mathrm{SC}} = \mathbb{E}[\|s_\theta(x_t; t, 0) - v_t\|_2^2 + \|s_\theta(x_t; t, 2d) - [s_\theta(x_t; t, d) + s_\theta(x'_{t+d}; t+d, d)]/2\|_2^2],$$

with $x'_{t+d} = x_t + d \cdot s_\theta(x_t, t, d)$. By setting $d = s - t$ and $F_\theta(x_t; t, s) = s_\theta(x_t; t, s-t)$, sampling $t \sim \mathcal{U}[0, 1]$, and choosing $s = t - 2^{-d'}$ for $d' \sim Cat[1, 7]$, we obtain the flow map under linear interpolation

$$f_\theta(x_t; t, s) = x_t + (s-t)F_\theta(x_t; t, s) = x'_{t+d}.$$

We can rewrite the flow matching objective of the Shortcut model as

$$\|s_\theta(x_t; t, 0) - v_t\|_2^2 = \|F_\theta(x_t; t, t) - v_t(x_t|x)\|_2^2.$$

For $r = s + d$, the consistency objective of the Shortcut Model can be written in the form of the semigroup condition, $f_{t,r}(x_t) = f_{s,r}(f_{t,s}(x_t))$:

$$\begin{aligned}
&\|s_\theta(x_t; t, 2d) - [s_\theta(x_t; t, d) + s_\theta(x'_{t+d}; t + d, d)]/2\|_2^2 \\
&= \|F_\theta(x_t; t, r) - [F_\theta(x_t; t, s) + F_\theta(f_\theta(x_t; t, s); s, r)]/2\|_2^2 \\
&= \frac{1}{4d^2}\|x_t + 2d \cdot F_\theta(x_t; t, r) - x_t - 2d[F_\theta(x_t; t, s) + F_\theta(f_\theta(x_t; t, s); s, r)]/2\|_2^2 \\
&= \frac{1}{4d^2}\|x_t + 2d \cdot F_\theta(x_t; t, r) - [x_t + d \cdot F_\theta(x_t; t, s) + d \cdot F_\theta(f_\theta(x_t; t, s); s, r)]\|_2^2 \\
&= \frac{1}{4d^2}\|f_\theta(x_t; t, r) - [f_\theta(x_t; t, s) + d \cdot F_\theta(f_\theta(x_t; t, s); s, r)]\|_2^2 \\
&= \frac{1}{4d^2}\|f_\theta(x_t; t, r) - f_\theta(f_\theta(x_t; t, s); s, r)\|_2^2.
\end{aligned}$$

Hence, the objective of the Shortcut Model is

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|F_\theta(x_t; t, t) - v_t\|_2^2 + \frac{1}{4d^2}\|f_\theta(x_t; t, r) - f_\theta(f_\theta(x_t; t, s); s, r)\|_2^2\right].$$

With the Taylor approximation of $F_{t,s} = F_\theta(x_t; t, s)$

$$F_{t,r} = F_{t,s} + d \cdot \partial_s F_{t,s} + O(d^2), \quad F_{s,r} = F_{t,s} + d \cdot \partial_t F_{t,s} + d \cdot \partial_s F_{t,s} + d \cdot F_{t,s} \cdot \nabla_x F_{t,s} + O(d^2),$$

we obtain

$$\begin{aligned}
&d[2F_{t,r} - F_{t,s} - F_{s,r}] \\
&= d[2[F_{t,s} + d \cdot \partial_s F_{t,s}] - F_{t,s} - [F_{t,s} + d \cdot \partial_t F_{t,s} + d \cdot \partial_s F_{t,s} + d \cdot F_{t,s} \cdot \nabla_x F_{t,s}]] + O(d^3) \\
&= d^2[\partial_s F_{t,s} - \partial_t F_{t,s} - F_{t,s} \cdot \nabla_x F_{t,s}] + O(d^3).
\end{aligned}$$

Thus,

$$\frac{1}{4d^2}\|f_{t,r}(x_t) - f_{s,r}(f_{t,s}(x_t))\|_2^2 = \frac{d^2}{4}\|\partial_s F_{t,s} - \partial_t F_{t,s} - F_{t,s} \cdot \nabla_x F_{t,s}\|_2^2 + O(d^3).$$

The differentiation of the linear flow map with respect to timestep t is given by

$$\frac{df_{t,s}}{dt} = v_t^* - F_{t,s} + (s - t) \cdot (\partial_t F_{t,s} + v_t^* \cdot \nabla_x F_{t,s}).$$

With the Taylor approximation and the relation $F_{t,t} \approx v_t^*$ obtained from $\mathcal{L}_{\text{CFM}}$, we have

$$F_{t,s} = F_{t,t} + d \cdot \partial_s F_{t,t} + O(d^2) \approx v_t^* + d \cdot \partial_s F_{t,t} + O(d^2).$$

The identity $\partial_s F_{t,s} = \partial_s F_{t,t} + O(d)$ implies

$$v_t^* - F_{t,s}(x_t) = -d \cdot \partial_s F_{t,s} + O(d^2).$$

Hence,

$$\frac{df_{t,s}}{dt} = d[\partial_t F_{t,s} + v_t^* \cdot \nabla_x F_{t,s} - \partial_s F_{t,s}] + O(d^2).$$

Since $v_t^* \approx F_{t,t} = F_{t,s} + O(d)$, we further obtain

$$\frac{df_{t,s}}{dt} = d[\partial_t F_{t,s} + F_{t,s} \cdot \nabla_x F_{t,s} - \partial_s F_{t,s}] + O(d^2)$$

and

$$\mathcal{L} = \mathbb{E}_{x,z,t,s}\left[\|F_\theta(x_t; t, t) - v_t\|_2^2 + \frac{1}{4}\left\|\frac{d}{dt} f_\theta(x_t; t, s)\right\|_2^2 + O(d^3)\right].$$

The model learns $F_\theta(x_t; t, t) \approx v_t^*(x_t)$ due to the flow matching loss term. This can be interpreted as the model learning a flow map corresponding to the trajectory induced by an approximated marginal velocity. Therefore, the Shortcut Model can be seen as Eulerian self-distillation under an $O(d^3)$ bound. □

Consistency Trajectory Model. Consistency Trajectory Model (CTM; Kim et al., 2024) can be interpreted as training a network to satisfy the semigroup condition of the flow map, $f_{t,r}(x_t) = f_{s,r}(f_{t,s}(x_t))$, using the following objective:

$$\mathcal{L}_{\text{CTM}} = \mathbb{E}[\|f_{\theta^-}(f_\theta(x_t; t, r); r, 0) - f_{\theta^-}(f_{\theta^-}(\Phi(x_t; t, s); s, r); r, 0)\|_2^2],$$

where Φ denotes a flow map along the learned flow of the teacher network, and timesteps satisfy $0 < r < s < t$. They show that, under suitable assumptions, it converges to the teacher flow map in the version without the stop-gradient operation:

$$\mathbb{E}[\|f_\theta(f_\theta(x_t; t, r); r, 0) - f_\theta(f_\theta(\Phi(x_t; t, s); s, r); r, 0)\|_2^2].$$

In the flow map perspective, while they assume a variance-exploding process and an x-prediction network in this setting, their velocity model follows the same formulation as linear flow (i.e., $v_t(x_t|x) = t^{-1}(x_t - x)$), and the flow map formulation is also consistent with it. Therefore, for analytical convenience, we adopt a linear interpolation-based formulation.

As shown in relation to the shortcut model, the semigroup-based objective, $\|f_\theta(x_t; t, r) - f_\theta(f_\theta(x_t; t, s); s, r)\|_2^2$, can be interpreted as a weighted Eulerian self-distillation framework. In addition, Boffi et al. (2025a) proves that the semigroup-based objective, denoted as progressive self-distillation, encourages the network to learn the desired flow map.

Since CTM aims to distill the knowledge of the teacher velocity network, it replaces the inner flow map $f_\theta(x_t; t, s)$ with the teacher flow $\Phi(x_t; t, s)$. If we assume that CTM uses the metric function $d_r(x, y) = \|f_{\theta^-}(x; r, 0) - f_{\theta^-}(y; r, 0)\|_2^2$, this objective reduces to $d_r(f_\theta(x_t; t, r), f_{\theta^-}(\Phi(x_t; t, s); s, r))$. We can view it as a teacher-guided, metricized semigroup-based flow map model, which admits an Eulerian distillation interpretation under an asymptotic gap. □

Consistency Flow Matching. Suppose a flow map under linear interpolation with fixed $s = 0$:

$$f_\theta(x_t; t) = x_t - tF_\theta(x_t; t).$$

Then, the Consistency Flow Matching objective from Yang et al. (2024) becomes

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|f_\theta(x_t; t) - f_{\theta^-}(x_{t-\Delta t}; t - \Delta t)\|_2^2 + \alpha\|F_\theta(x_t; t) - F_{\theta^-}(x_{t-\Delta t}; t - \Delta t)\|_2^2\right].$$

The first term on the right side corresponds to the consistency training objective, and the second term is the regularizer. Hence, we can interpret Consistency Flow Matching as training a flow map model via consistency training with regularization. □

UCGM. For arbitrary interpolation of α_t, σ_t, setting $s = 0$ yields

$$f_\theta(x_t; t) = \nu_t^{-1}(\sigma_t' x_t - \sigma_t F_\theta).$$

We can reformulate the objective while preserving the gradient direction up to a positive scalar weight:

$$\begin{aligned}
&\nabla_\theta \|f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)\|_2^2 \\
&= 2 \cdot \nabla_\theta f_\theta(x_t; t) \cdot (t - \lambda t) \frac{f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)}{t - \lambda t} \\
&\propto 2 \cdot \nabla_\theta f_\theta(x_t; t) \cdot \frac{f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)}{t - \lambda t} \\
&= -2\frac{\sigma_t}{\nu_t} \nabla_\theta F_\theta(x_t; t) \cdot \frac{f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)}{t - \lambda t} \\
&= \nabla_\theta \left\| F_\theta(x_t; t) - F_{\theta^-}(x_t; t) - \frac{\sigma_t[f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)]}{\nu_t(t - \lambda t)} \right\|_2^2,
\end{aligned}$$

which gives the objective of UCGM. When $\lambda = 0$, this reduces to the flow matching objective since $\lambda t = 0$ collapses $f_{\theta^-}(x_0; 0) = x_0$. In this case, the objective becomes origin prediction, which in turn yields F_θ with a velocity-matching objective. Otherwise, setting $\lambda \to 1$ reduces the objective to consistency training by $\Delta = F_\theta(x_t; t) - F_{\theta^-}(x_t; t) - \frac{\sigma_t}{\nu_t}\frac{df_{\theta^-}(x_t;t)}{dt}$. For $\lambda \in (0, 1)$, the objective $\|f_\theta(x_t; t) - f_{\theta^-}(x_{\lambda t}; \lambda t)\|_2^2$ yields consistency along the geometric sequence $\mathcal{T}_\lambda(t) = \{\lambda^k t\}_{k=0}^N$.

Define

$$g_t(x_t) = (x_t - \alpha_t f_t(x_t))\sigma_t^{-1}.$$

Then, when $f_t(x_t) = x$, it follows that $g_t(x_t) = z$ for $x_t = \alpha_t x + \sigma_t z$. Using this, the flow map can be formulated in a DDIM-like manner as

$$f_{t,s}(x_t) = \alpha_s f_t(x_t) + \sigma_s g_t(x_t) = \frac{\sigma_s}{\sigma_t} x_t + (\alpha_s - \frac{\sigma_s}{\sigma_t}\alpha_t) f_t(x_t).$$

Assuming the composition chain

$$f_{s,r}(f_{t,s}(x_t)) = f_{s,r}(\tilde{x}_s) = \frac{\sigma_r}{\sigma_s}\tilde{x}_s + (\alpha_r - \frac{\sigma_r}{\sigma_s}\alpha_s) f_s(\tilde{x}_s),$$

for $\tilde{x}_s = f_{t,s}(x_t)$, we obtain

$$f_{t,r}(x_t) - f_{s,r}(f_{t,s}(x_t)) = (\alpha_r - \frac{\sigma_r}{\sigma_s}\alpha_s)(f_t(x_t) - f_s(\tilde{x}_s)).$$

For $s = \lambda^k t$ for some $k \in \mathbb{N}$, if it follows that $\tilde{x}_s = f_{t,s}(x_t) \approx x_s$, then $f_s(x_s) = f_t(x_t)$ and $f_{t,r} = f_{s,r} \circ f_{t,s}$. In this case, the flow map can be constructed along the geometric sequence $\mathcal{T}_\lambda(t)$.

In general, the velocity of the DDIM map is given by $\frac{d}{ds} f_{t,s} = \alpha_s' f_t + \sigma_s' g_t$. Since the marginal velocity is

$$v_t^*(x_t) = \alpha_t' \mathbb{E}_{x|x_t}[x] + \sigma_t' \mathbb{E}_{x|x_t}[(x_t - \alpha_t x)\sigma_t^{-1}],$$

the DDIM map coincides with the flow map only when $v_s^*(f_{t,s}(x_t)) = \frac{d}{ds} f_{t,s}(x_t)$, which implies $\mathbb{E}_{x|\tilde{x}_s}[x] = f_t(x_t)$. Setting $s \to t$ reduces this condition to $\mathbb{E}_{x|x_t}[x] = f_t(x_t)$ by the identity assumption. However, this condition fails to preserve the injectivity of the flow map at $t = 1$ due to the mean collapse problem. Therefore, the DDIM-style map does not generally coincide with the flow map. □

Reflow. Rectified flows introduce Reflow to straighten trajectories after training. In Reflow, sampling from the trained model is performed via

$$x_0 = z + \int_1^0 v_\theta(x_t; t)dt \approx \text{ODESolver}(v_\theta, z, 1, 0),$$

followed by finetuning w.r.t. the coupling $\Pi_{Z,\theta} = p_Z(z) p_{v_\theta}(x|z)$. The velocity $\hat{v}_t$ of the trajectory induced by the coupling $\Pi_{Z,\theta}$ is given by

$$\hat{v}_t = z - \left(z + \int_1^0 v_\theta(x_t; t)dt\right) = \int_0^1 v_\theta(x_t; t)dt,$$

which corresponds to the displacement of the flow map. Therefore, Reflow can be interpreted as direct supervision of the flow map under linear interpolation.

A.6 Suboptimality of direct training

Unlike Eulerian distillation, direct training does not guarantee convergence to the optimal flow map. Consider the direct training objective using the conditional velocity:

$$\mathcal{L}_{\text{DT}} = \mathbb{E}_{x,z,t,s}\left[\|\partial_t f_\theta(x_t; t, s) + v_t(x_t|x) \cdot \nabla_x f_\theta(x_t; t, s)\|_2^2\right].$$

By defining the velocity error as $\Delta v = v_t(x_t|x) - v_t^*(x_t)$, we can rewrite the objective in the form of Eulerian distillation:

$$\begin{aligned}\mathcal{L}_{\mathrm{DT}} &= \mathbb{E}\left[\|\partial_t f_\theta(x_t;t,s) + (\Delta v + v_t^*(x_t)) \cdot \nabla_x f_\theta(x_t;t,s)\|_2^2\right] \\ &= \mathbb{E}\left[\|\partial_t f_\theta(x_t;t,s) + \Delta v \cdot \nabla_x f_\theta(x_t;t,s) + v_t^*(x_t) \cdot \nabla_x f_\theta(x_t;t,s)\|_2^2\right] \\ &= \mathbb{E}\left[\|\partial_t f_\theta(x_t;t,s) + v_t^*(x_t) \cdot \nabla_x f_\theta(x_t;t,s)\|_2^2\right] + \mathbb{E}\left[\|\Delta v \cdot \nabla_x f_\theta(x_t;t,s)\|_2^2\right],\end{aligned}$$

where the cross term vanishes, since $\mathbb{E}_{x|x_t}[\Delta v] = 0$ and the remaining terms are fixed given x_t. In this case, the second term can be represented as

$$\mathbb{E}\left[\mathrm{Var}_{x|x_t}\left[\Delta v \cdot \nabla_x f_\theta(x_t;t,s)\right]\right].$$

Under an independent coupling, the velocity error $\Delta v = v_t(x_t|x) - v_t^*(x_t)$ is typically nonzero. Consequently, unless $\|\nabla_x f_\theta(x_t;t,s)\|$ collapses to zero, the objective function inherently contains a larger variance term compared to that of Eulerian distillation. To minimize the overall loss, the optimizer faces a trade-off involving this variance.

Note that for flow matching, even when the loss term $\mathbb{E}_{x,z,t}[\|v_t(x_t|x) - F_\theta(x_t;t)\|_2^2]$ is decomposed as follows

$$\mathbb{E}\left[\|\Delta v + v^*(x_t) - F_\theta(x_t;t)\|_2^2\right] = \mathbb{E}\left[\|v^*(x_t) - F_\theta(x_t;t)\|_2^2\right] + \mathrm{Var}[\Delta v].$$

The variance term, $\mathrm{Var}[\Delta v]$, is independent of the network. Therefore, it does not affect convergence to the global optimum.

Stationary Condition. Since the gradients of Eulerian distillation and direct training differ, the optima obtained by gradient-based optimization can also diverge. To find and compare the stationary points of the direct training objective, we apply the Euler-Lagrange equation. The objective can be represented in the vector form as

$$\mathcal{L}_{\mathrm{DT}} = \mathbb{E}_{x,z,t,s}\left[\|\partial_t f_{t,s}(x_t) + v_t(x_t|x)^T \nabla_x f_{t,s}(x_t)\|_2^2\right] = \iiint_\Omega \rho_t L(f, \partial_t f, \nabla_x f)\, dx_t\, dt\, ds.$$

We set the conditional expectation as the Lagrangian,

$$\rho_t L(f, \partial_t f, \nabla_x f) = \rho_t(x_t) \cdot \mathbb{E}_{x|x_t}[\|\partial_t f_{t,s}(x_t) + v_t(x_t|x)^T \nabla_x f_{t,s}(x_t)\|_2^2].$$

The corresponding Euler-Lagrange equation is

$$\frac{\partial(\rho_t L)}{\partial f} - \partial_t\left(\frac{\partial(\rho_t L)}{\partial(\partial_t f)}\right) - \nabla_x \cdot \left(\frac{\partial(\rho_t L)}{\partial(\nabla_x f)}\right) = 0 \iff \mathcal{EL} = \partial_t(\rho_t \mathbb{E}_{x|x_t}[R]) + \nabla_x \cdot (\rho_t \mathbb{E}_{x|x_t}[v_t R]) = 0,$$

where the residue is defined as

$$R = \partial_t f_{t,s}(x_t) + v_t(x_t|x)^T \nabla_x f_{t,s}(x_t).$$

Letting $\Delta v = v_t(x_t|x) - v_t^*(x_t)$ and introducing the operator $L_* f = \partial_t f + (v_t^*)^T \nabla_x f$, the residue can be rewritten as $R = L_* f_{t,s} + \Delta v^T \nabla_x f_{t,s}$, so that

$$\begin{aligned}&\partial_t\left(\rho_t \mathbb{E}_{x|x_t}[R]\right) = \partial_t(\rho_t L_* f_{t,s}), \\ &\nabla_x \cdot (\rho_t \mathbb{E}_{x|x_t}[v_t R]) = \nabla_x \cdot (\rho_t \mathbb{E}_{x|x_t}[(v_t^* + \Delta v)(L_* f_{t,s} + \Delta v^T \nabla_x f_{t,s})]) \\ &\qquad = \nabla_x \cdot \left(\rho_t v_t^* L_* f_{t,s} + \rho_t \mathbb{E}_{x|x_t}[\Delta v(\Delta v^T \nabla_x f_{t,s})]\right) \\ &\qquad = \nabla_x \cdot \left(\rho_t v_t^* L_* f_{t,s} + \rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s}\right).\end{aligned}$$

Therefore, the optimality condition becomes

$$\mathcal{EL} = \partial_t(\rho_t L_* f_{t,s}) + \nabla_x \cdot (\rho_t v_t^* L_* f_{t,s}) + \nabla_x \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s}) = 0,$$

and using the continuity equation $\partial_t \rho_t + \nabla \cdot (\rho_t v_t^*) = 0$ further simplifies to

$$\begin{aligned}\mathcal{EL} &= (L_* f_{t,s})\partial_t \rho_t + \rho_t(\partial_t L_* f_{t,s}) + (L_* f_{t,s})\nabla \cdot (\rho_t v_t^*) + (\rho_t v_t^*)^T \nabla (L_* f_{t,s}) + \nabla \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s}) \\ &= \cancel{(L_* f_{t,s})(\partial_t \rho_t + \nabla \cdot (\rho_t v_t^*))} + \rho_t(\partial_t L_* f_{t,s} + (v_t^*)^T \nabla (L_* f_{t,s})) + \nabla \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s}) \\ &= \cancel{\rho_t L_*(L_* f_{t,s})} + \nabla \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s})\end{aligned}$$

When $\Sigma_{\Delta v|x_t} = \mathrm{Cov}_{x|x_t}[v_t(x_t|x)] \to 0$, the condition reduces to $\rho_t L_*(L_* f_{t,s}) = 0$, which coincides with the stationary condition of Eulerian distillation. Thus, in Prop. 4.2, we denote it by $\mathcal{S}_{\mathrm{ED}} = \rho_t L_*(L_* f_{t,s})$, and the bias term by $C = \nabla \cdot (\rho_t \Sigma_{\Delta v|x_t} \nabla_x f_{t,s})$, such that $\mathcal{EL} = \mathcal{S}_{\mathrm{ED}} + C = 0$. Therefore, the stationary point of direct training can fail to satisfy the stationary condition of Eulerian distillation: $\mathcal{S}_{\mathrm{ED}} = -C \neq 0$. □

A.7 Instability of Consistency Training

The continuous-time consistency training objective employs a stop-gradient operation. In this case, the objective is defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{CT}} &= \mathbb{E}\left[f_{t,s}(x_t)^T \,\mathrm{sg}\left[\frac{df_{t,s}(x_t)}{dt}\right]\right] \\ &= \mathbb{E}\left[f_{t,s}(x_t)^T \,\mathrm{sg}\left[L_* f_{t,s} + \Delta v^T \nabla_x f_{t,s}\right]\right].\end{aligned}$$

Following Sec. A.6, the stationary condition induced by the stop-gradient objective is:

$$\begin{aligned}\mathcal{EL} &= \rho_t(x_t) \cdot \mathbb{E}_{x|x_t}\left[L_* f_{t,s} + \Delta v^T \nabla_x f_{t,s}\right] \\ &= \rho_t(x_t) \cdot \left(L_* f_{t,s} + \mathbb{E}_{x|x_t}\left[\Delta v^T \nabla_x f_{t,s}\right]\right) \\ &= \rho_t(x_t) \cdot L_* f_{t,s} = 0,\end{aligned}$$

and this stationary condition is equivalent to the Eulerian condition in Eq. (3) when $\rho_t(x_t) > 0$. The stationary condition of $\mathcal{L}_{\mathrm{CT}}$ reduces to that of consistency distillation at the expectation level, which is consistent with the observation in Sec. A.5. However, when the expectation is replaced by a finite-batch estimate, the empirical residual term corresponding to $\mathbb{E}[\Delta v^T \nabla_x f_{t,s}]$ may not vanish and may contribute to degenerate solutions in small batch settings (Fig. 3). □

Fixed-point property. We also note that since the objective is expressed as a linear term, the PSD (positive-semidefinite) curvature that encourages stable convergence at the optimum is absent. It only specifies the fixed point on $L_* f_{t,s} = 0$, and the gradient dynamics alone may fail to converge. Empirically, we show that different initializations can result in varying performance (Tab. 2). In some cases, training diverges to degenerate solutions and fails to generate meaningful images (details in Sec. C.1).

Stop-gradient Placement. Boffi et al. (2025b) analyzes the formulation that applies the stop-gradient operation only to the spatial gradient term:

$$\mathbb{E}\left[\left\|\partial_t f_{t,s} - \mathrm{sg}\left[v_t^T \nabla_x f_{t,s}\right]\right\|_2^2\right].$$

This formulation differs in structure from the consistency-style objectives discussed above.

While they show that the functional gradient of this formulation can align with that of Eulerian distillation and admits fixed-point solutions, this analysis does not directly extend to prior consistency models, whose numerical realizations and optimization dynamics differ.

Motivated by this distinction, we formally analyze consistency-style objectives and show that they also admit fixed-point solutions satisfying the Eulerian equation, without implying stable or global convergence of the resulting optimization dynamics.

To highlight differences in numerical realization, we compare the gradient norms of each method on toy data, as shown in Fig. 8. We observe that self-distillation with a stop-gradient applied to the spatial derivative can yield larger gradient norms than the original formulation, whereas our improved Self-Distillation yields consistently smaller gradient norms, which can help stabilize optimization.

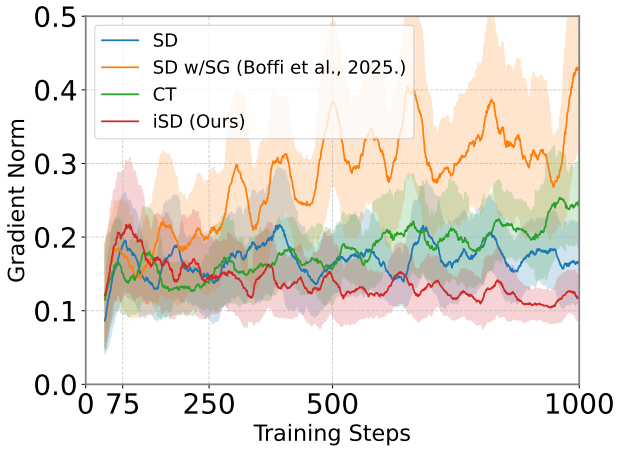


Figure 8: Gradient norms of each method. We estimate gradient norms under the same setting as Fig. 2. *SD* denotes Eulerian self-distillation (ESD), *SD w/SG* denotes the stop-gradient version of ESD introduced by Boffi et al. (2025a), *CT* denotes the continuous-time consistency training objective, and *iSD* denotes our improved Self-Distillation objective.

Network-Induced Coupling. Prior work (Issenhuth et al., 2025; Silvestri et al., 2025) has introduced network-induced couplings (NIC) to reduce loss variance. For x and z, NIC replaces one of them with a network prediction. Suppose that $\hat{x} = f_{t,0}(x_t)$ with stop-gradient: $\hat{x}_t = \alpha_t \,\mathrm{sg}[f_{t,0}(x_t)] + \sigma_t z$ for $x_t = \alpha_t x + \sigma_t z$. The conditional velocity of this coupling is $\hat{x}'_t = \alpha'_t \,\mathrm{sg}[f_{t,0}(x_t)] + \sigma'_t z$ and the marginal velocity becomes:

$$v_t^{\mathrm{NIC}}(\hat{x}_t) = \mathbb{E}_{x,z|\hat{x}_t}[\alpha'_t \,\mathrm{sg}[f_{t,0}(x_t)] + \sigma'_t z].$$

This differs from the marginal velocity v_t^* induced by the standard independent coupling, $v_t^{\mathrm{NIC}}(x_t) \neq v_t^*(x_t)$, in general. Therefore, the resulting flow map does not necessarily need to follow the standard flow field. We note that these couplings are known to reduce loss variance; however, we do not primarily focus on them, as the flow field is inconsistent with our targets.

Discrete-time models can be viewed as numerical approximations of corresponding continuous systems (Song et al., 2023). We therefore primarily focus on the behavior of continuous systems that discrete models approximate. The identified instability arises from the structure of the objective, rather than time discretization. Thus, our analysis provides plausible explanations for similar instabilities observed in practice.

A.8 Linearization Cost Hypothesis

Recall that $f_{t,s}(x_t) = \nu^{-1}(A'_{t,s}x_t - A_{t,s}F_{t,s}(x_t))$ for $A_{t,s} = \alpha_s\sigma_t - \sigma_s\alpha_t$. Under the given formulation of flow maps, a pseudo velocity network F_θ learns the weighted displacement between x_t and x_s. From the perspective of F_θ, this process can be seen as shifting the training target from a classical instantaneous velocity field to a weighted displacement field, which transforms the path between x_t and x_s into the linear form of x_t and F_θ. We refer to it as *linearization*.

Differentiating with respect to the starting point t gives

$$\frac{df_\theta(x_t;t,s)}{dt} = \nu^{-1}\left(A''_{t,s}x_t + A'_{t,s}(v_t^*(x_t) - F_\theta(x_t;t,s)) - A_{t,s}\frac{dF_\theta(x_t;t,s)}{dt}\right).$$

Following AYF-EMD (Sabour et al., 2025), the gradient of the consistency training can be written as

$$\begin{aligned}&\nabla_\theta \mathbb{E}\left[2f_\theta^T(x_t;t,s)\frac{df_{\theta^-}(x_t;t,s)}{dt}\right]\\ &\propto \nabla_\theta \mathbb{E}\left[-A_{t,s}\nu^{-2}F_\theta(x_t;t,s)\cdot\left(A''_{t,s}x_t + A'_{t,s}(v_t^*(x_t) - F_{\theta^-}(x_t;t,s)) - A_{t,s}\frac{dF_{\theta^-}(x_t;t,s)}{dt}\right)\right].\end{aligned}$$

In this case, $v_t^* - F_\theta$ can be interpreted as the flow matching term weighted by $A'_{t,s}$, and $A''_{t,s}x_t - A_{t,s}dF_\theta/dt$ as a linearization term involving the JVP, which penalizes the t-dependent outputs of f_θ. We note that $A_{t,t} = 0$ by definition, and $A''_{t,t} = \partial_t^2 A_{t,s}|_{s=t} = 0$ since the ν-condition gives $A'_{t,t} = \partial_t A_{t,s}|_{s=t} = \nu$, where ν is the time-independent constant. Thus, the linearization term vanishes when $s \to t$, and the objective becomes flow matching. Conversely, $s \to 0$ for fixed t increases the step size and relatively strengthens the linearization term.

We hypothesize that the linearization cost increases with step size, making optimization more challenging. This is because the linearization term involves a complex structure induced by the JVP, whereas the flow matching term requires only a simple forward pass. Such a complex structure may introduce additional fixed points or spikes, hindering convergence to the desired optimum. To demonstrate this hypothesis, we construct loss landscapes before and after time-condition relaxation in Sec. C.6, and observe that the relaxation reduces loss spikes and variances empirically.

A.9 Training-time Classifier-free Guidance

MeanFlow suggests Classifier-free Guidance (CFG; Ho & Salimans, 2021) for flow maps:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,s,c) - \mathrm{sg}\left[\tilde{v}_t(x_t|x) - (t-s)\left(\tilde{v}_t(x_t|x)\cdot\nabla_x F_\theta + \partial_t F_\theta\right)\right]\|\right] \\
&\text{where} \quad \tilde{v}_t(x_t|x) = F_{\theta^-}(x_t;t,t,\varnothing) + \omega(v_t(x_t|x) - F_{\theta^-}(x_t;t,t,\varnothing)),
\end{aligned}$$

for a conditional class label c, the null class label $\varnothing$ of unconditional generation, and a guidance scale ω. We find that this formulation encourages the flow map to follow the ground-truth CFG trajectory. The ground-truth CFG trajectory is given by $\tilde{v}_t^*(x_t;c) = v_t^*(x_t) + \omega(v_t^*(x_t;c) - v_t^*(x_t))$ where $v_t^*(x_t;c)$ is the class-conditional marginal velocity $v_t^*(x_t;c) = \mathbb{E}_{x|x_t,c}[v_t(x_t|x)]$ and $v_t^*(x_t)$ is the marginal velocity $v_t^*(x_t) = \mathbb{E}_{x|x_t}[v_t(x_t|x)] = \mathbb{E}_{c|x_t}[v_t^*(x_t;c)]$.

To prove this, we first consider only the flow matching setting:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,c) - \tilde{v}_t(x_t|x)\|_2^2\right] \\
&= \nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,c) - (F_{\theta^-}(x_t;t,\varnothing) + \omega\left(v_t(x_t|x) - F_{\theta^-}(x_t;t,\varnothing)\right))\|_2^2\right].
\end{aligned}$$

With label dropout $c = \varnothing$, the objective reduces to weighted unconditional flow matching that converges to the unconditional marginal velocity:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,\varnothing) - (F_{\theta^-}(x_t;t,\varnothing) + \omega(v_t(x_t|x) - F_{\theta^-}(x_t;t,\varnothing)))\|_2^2\right] \\
&= \mathbb{E}\left[2\omega\cdot\nabla_\theta F_\theta(x_t;t,\varnothing)\cdot(F_{\theta^-}(x_t;t,\varnothing) - v_t(x_t|x))\right] \\
&= \nabla_\theta \mathbb{E}\left[\omega\,\|F_\theta(x_t;t,\varnothing) - v_t(x_t|x)\|_2^2\right].
\end{aligned}$$

In this case, we can set $F_\theta(x_t;t,\varnothing) \approx v_t^*(x_t)$ if the network has sufficient capacity. For the conditional case $c \neq \varnothing$, applying the tower property yields:

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,c) - (F_\varnothing(x_t;t) + \omega(v_t(x_t|x) - F_\varnothing(x_t;t)))\|_2^2\right] \\
&= \nabla_\theta \mathbb{E}\left[\mathbb{E}_{x|x_t,c}\left[\|F_\theta(x_t;t,c) - (F_\varnothing(x_t;t) + \omega(v_t(x_t|x) - F_\varnothing(x_t;t)))\|_2^2\right]\right] \\
&= \nabla_\theta \mathbb{E}\left[\mathbb{E}_{x|x_t,c}\left[\|F_\theta(x_t;t,c) - (F_\varnothing(x_t;t) + \omega(v_t^*(x_t;c) - v_t^*(x_t;c) + v_t(x_t|x) - F_\varnothing(x_t;t)))\|_2^2\right]\right] \\
&= \nabla_\theta \mathbb{E}\left[\mathbb{E}_{x|x_t,c}\left[\|F_\theta(x_t;t,c) - (F_\varnothing(x_t;t) + \omega(v_t^*(x_t;c) - F_\varnothing(x_t;t))) + \omega(v_t^*(x_t;c) - v_t(x_t|x))\|_2^2\right]\right] \\
&\approx \nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,c) - (v_t^*(x_t) + \omega(v_t^*(x_t;c) - v_t^*(x_t)))\|_2^2\right] + \omega^2\cdot\nabla_\theta \mathbb{E}\left[\mathrm{Var}_{x|x_t,c}\left[v_t(x_t|x)\right]\right] \\
&= \nabla_\theta \mathbb{E}\left[\|F_\theta(x_t;t,c) - \tilde{v}_t^*(x_t;c)\|_2^2\right],
\end{aligned}$$

where $F_\varnothing(x_t;t) = F_{\theta^-}(x_t;t,\varnothing)$ for brevity. Thus, the self-referential target $\tilde{v}_t(x_t|x)$ with label dropout ensures convergence to the ground-truth CFG velocity.

Consider the convergence of the flow map to the CFG velocity field. Following Sec. A.5, let $L_* f_{t,s} = \partial_t f_{t,s} + \tilde{v}_t^* \cdot \nabla_x f_{t,s}$ and $\Delta v = \tilde{v}_t(x_t|x) - \tilde{v}_t^*(x_t;c)$. It is sufficient to show that $\mathbb{E}_{x|x_t,c}[\Delta v] = 0$, and it is encouraged when label dropout is applied, and the flow matching objective is jointly optimized. Thus, the objective encourages convergence to the flow map induced by the CFG velocity. □

B Improving Self-Distillation

B.1 Compatibility of the Joint Objective

Revisit our objective

$$\mathcal{L} = \mathbb{E}_{x,z,t,s}\left[\|F_\theta(x_t;t,t) - v_t(x_t|x)\|_2^2 + \|\partial_t f_\theta(x_t;t,s) + F_{\theta^-}(x_t;t,t)\cdot\nabla_x f_\theta(x_t;t,s)\|_2^2\right].$$

The first term on the right-hand side trains $F_\theta(x_t;t,t)$ to approximate the marginal velocity via the flow matching objective, while the second term learns the flow map $f_{t,s}$ along the trajectory of $F_{\theta^-}(x_t;t,t)$ in a self-distillation manner.

Individually, each term is encouraged to converge to its desired optimum: the marginal velocity and the flow map by the Eulerian equation (Sec. A.3). From a joint perspective, we need to consider $t = s$, since the network is forced to optimize both terms simultaneously at this point. As Eulerian distillation collapses to the flow matching objective when $s \to t$ under the ν-condition, the second term trains the model to learn the instantaneous velocity of the trajectory (Sec. A.8). In this case, $F_\theta(x_t;t,t)$ learns from $F_{\theta^-}(x_t;t,t)$, and inductively approximates $v_t^*(x_t)$ through the first term. This naturally reduces to non-conflicting joint training. For $t \neq s$, the network is conditioned differently in the two terms, and it can learn the proper mapping provided that the network capacity is sufficient. Consequently, the overall objective naturally trains the network to follow the marginal velocity as the trajectory of the flow map. □

B.2 Deriving Final Objective

Recall the consistency training objective under the flow map representation (Sec. A.8):

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[f_\theta^T(x_t;t,s)\frac{df_{\theta^-}(x_t;t,s)}{dt}\right] \\
&\propto \nabla_\theta \mathbb{E}\left[-A_{t,s}\nu^{-2}F_\theta(x_t;t,s)\cdot\left(A''_{t,s}x_t + A'_{t,s}(v_t^*(x_t) - F_{\theta^-}(x_t;t,s)) - A_{t,s}\frac{dF_{\theta^-}(x_t;t,s)}{dt}\right)\right] \\
&= \nabla_\theta \mathbb{E}\left[A_{t,s}\nu^{-2}\|F_\theta(x_t;t,s) - \mathrm{sg}[F_{\mathrm{tgt}}(x_t;t,s)]\|_2^2\right],
\end{aligned}$$

where $F_{\mathrm{tgt}}(x_t;t,s) = F_\theta(x_t;t,s) + \left(A''_{t,s}x_t + A'_{t,s}(v_t^*(x_t;t) - F_\theta(x_t;t,s)) - A_{t,s}\frac{dF_\theta(x_t;t,s)}{dt}\right)$.

To follow the marginal velocity, we replace $v_t^*(x_t)$ with instantaneous velocity $v_\theta(x_t;t) = F_\theta(x_t;t,t)$ while jointly training with $\mathcal{L}_{\mathrm{CFM}}$. Particularly, for linear interpolation, we have $A_{t,s} = t-s$, $A'_{t,s} = 1$ and $A''_{t,s} = 0$. This simplifies the target to $F_{\mathrm{tgt}}^{\mathrm{lin}}(x_t;t,s) = v_\theta(x_t;t) - (t-s)F'_\theta(x_t;t,s)$ which coincides with the regression target of MeanFlow. For trigonometric interpolation, we have $A_{t,s} = \sin(t-s)$, $A'_{t,s} = \cos(t-s)$ and $A''_{t,s} = -\sin(t-s)$. Thus, the target becomes $F_{\mathrm{tgt}}^{\mathrm{tri}} = F_\theta(x_t;t,s) + \cos(t-s)\cdot(v_\theta(x_t;t) - F_\theta(x_t;t,s)) - \sin(t-s)\cdot(x_t + F'_\theta(x_t;t,s))$.

Although consistency training already guarantees the marginal flow map at its fixed point, the gradient in practice can exhibit a gap expressed as:

$$\mathbb{E}_{x,z,t,s}[f_{t,s}\cdot(\Delta v\cdot\nabla_x f_{t,s})].$$

When self-distillation is combined with flow matching, $\mathbb{E}\left[\|F_\theta(x_t;t,t) - v(x_t|x)\|_2^2\right]$, the velocity error $\Delta v = F_\theta(x_t;t,t) - v_t^*(x_t) \approx v_t^*(x_t) - v_t^*(x_t) = 0$ can be further reduced compared to $\Delta v = v_t(x_t|x) - v_t^*(x_t)$, thereby

stabilizing the training. In Sec. C.6, we demonstrate that self-distillation empirically reduces loss variance through loss landscape analysis.

For JVP approximation, to ensure that dx_t/dt follows the velocity $v_\theta(x_t;t) = F_\theta(x_t;t,t)$, we approximate

$$\frac{dx_t}{dt} \approx \frac{[x_t + \epsilon \cdot v_\theta(x_t;t)] - [x_t - \epsilon \cdot v_\theta(x_t;t)]}{2\epsilon} = v_\theta(x_t;t).$$

Thus, the full JVP approximation becomes

$$\frac{dF_\theta(x_t;t,s)}{dt} = \frac{F_\theta(x_t + \epsilon \cdot v_\theta(x_t;t), t+\epsilon, s) - F_\theta(x_t - \epsilon \cdot v_\theta(x_t;t), t-\epsilon, s)}{2\epsilon} + O(\epsilon^2).$$

Our final objective is

$$\begin{aligned}
\mathcal{L}_{\text{iSD}} &= \lambda_1 \mathcal{L}_{\text{CFM}} + \lambda_2 \mathcal{L}_{\text{SD-R}}, \\
\mathcal{L}_{\text{CFM}} &= \mathbb{E}\left[\|F_\theta(x_t;t,t) - v_t(x_t|x)\|_2^2\right], \quad \mathcal{L}_{\text{SD-R}} = \mathbb{E}\left[\|F_\theta(x_t;t,s) - \text{sg}\left[F_{\text{tgt}}(x_t;t,s)\right]\|_2^2\right].
\end{aligned}$$

We consider two weighting methods: (i) cosine weighting $\lambda_1 = \lambda_2 = \cos(\pi t/2)$ from Sun et al. (2025), and (ii) adaptive weighting $\lambda_1 = \lambda_2 = (\text{sg}[\tilde{\mathcal{L}}_{t,s}(x_t,x)] + \eta)^{-p}$ where $\tilde{\mathcal{L}}_{t,s}(x_t,x) = \|F_\theta(x_t;t,t) - v_t(x_t|x)\|_2^2 + \|F_\theta(x_t;t,s) - \text{sg}[F_{\text{tgt}}(x_t,t,s)]\|_2^2$ from Geng et al. (2025a). While our method is sufficiently stable to train flow maps without additional weighting, we found that adaptive weighting can further improve performance in the SD-VAE setting, and cosine weighting in the VA-VAE setting. We use adaptive weighting by default, except when using VA-VAE.

When using JVP approximation, the truncation error is $O(\epsilon^2)$. This error appears acceptable under BF16 precision, as the approximation shows better results than direct JVP computation. The complete training and sampling algorithms are provided in Alg. 1 and Alg. 2.

We note that the update in consistency training can be viewed as a coupled fixed-point iteration with a parametrized target. In this setting, contraction and global convergence properties are generally difficult to establish without additional assumptions. Instead, Sec. B.1 shows that the two objectives are aligned and do not introduce conflicts, which we find sufficient for stable training in practice. We leave deriving global convergence rates for future work.

Algorithm 1 (iSD Training) Training algorithm of vanilla iSD

Input: Noise distribution p_Z, data distribution p_X, model F_θ, learning rate μ, time distribution τ, adaptive weighting (η, p), JVP approximation step size ϵ, class labels c
repeat
  $z \sim p_Z, \quad x \sim p_X, \quad t, s \leftarrow \tau$
  $x_t \leftarrow \alpha_t x + \sigma_t z, \quad v_t \leftarrow \alpha_t' x + \sigma_t' z$
  $F_{t,s} \leftarrow F_\theta(x_t;t,s,c), \quad F_{t,t} \leftarrow F_\theta(x_t;t,t,c)$
  $F_{t,s}' \leftarrow [F_\theta(x_t + \epsilon F_{t,t}; t+\epsilon, s, c) - F_\theta(x_t - \epsilon F_{t,t}; t-\epsilon, s, c)]/(2\epsilon)$ ▷ JVP-Approx.
  $F_{\text{tgt}} \leftarrow F_{t,s} + \left(A_{t,s}'' x_t + A_{t,s}'(F_{t,t} - F_{t,s}) - A_{t,s} F_{t,s}'\right)$
  $\mathcal{L} \leftarrow \lambda_1 \|F_{t,t} - v_t\|_2^2 + \lambda_2 \|F_{t,s} - \text{sg}[F_{\text{tgt}}]\|_2^2$ ▷ Optimization Target
  $\theta \leftarrow \theta - \mu \nabla_\theta \mathcal{L}$ ▷ Model Update
until Convergence

Algorithm 2 (iSD Sampling) Sampling algorithm of vanilla iSD

Input: Initial noise $z \sim p_Z$, trained model F_θ, class labels c, sampling time steps $\{t_i\}_{i=1}^{N+1}$
$x \leftarrow z$
for $i \leftarrow 1$ to N **do**
  $x \leftarrow \nu^{-1}(A_{t_i,t_{i+1}}' x - A_{t_i,t_{i+1}} F_\theta(x; t_i, t_{i+1}, c))$
end for

B.3 Random initialization instead of Preconditioners

When constructing the iSD framework, we focus on random initializations rather than preconditioners. As shown in Tab. 2, pretrained networks with better FIDs do not necessarily yield better FIDs in the resulting consistency models. With random initialization, one can explore different random seeds and retrain the network to obtain improved results. There also remains potential to improve performance using alternative initialization strategies, such as leveraging preconditioners. However, when adopting preconditioners as baselines, it becomes difficult to systematically design experiments to discover *better preconditioners*: which data, architectures, and hyperparameters should be used, how to quantify the quality of preconditioners, and whether perturbing pretrained weights is necessary. Therefore, we conclude that random initialization makes the reproducibility problem more tractable, and we construct the iSD as a randomly initializable framework.

B.4 Classifier-free Guidance of Flow Map Models

Pre-CFG. By abstracting the guiding trajectory to v_θ, the flow map model can naturally be trained to follow the specific trajectory as long as it is Lipschitz continuous. To achieve this, we replace the guiding trajectory and refer to this training scheme as *Pre-CFG*. We consider three variants of objectives: iSD-C, iSD-U, and iSD-T.

iSD-U (Guidance-**U**nconditional). Given a label c and a null class label $\varnothing$, let the corresponding velocity fields be $F_{t,t}(x_t; c)$ and $F_{t,t}(x_t; \varnothing)$. If both are globally Lipschitz continuous, then the CFG trajectory $\tilde{v}_\theta(x_t; t, c)$ is also globally Lipschitz continuous, since any linear combination of Lipschitz continuous functions remains Lipschitz continuous.

$$\tilde{v}_\theta(x_t; t, c) = F_\theta(x_t; t, t, \varnothing) + \omega(F_\theta(x_t; t, t, c) - F_\theta(x_t; t, t, \varnothing)).$$

Thus, the flow map can be trained to follow the CFG velocity field. As a naive approach, we first train the network $F_\theta(x_t; t, s, c)$ to align with $\tilde{v}_\theta$, while leaving the remaining components unchanged. This shares a similar scheme with the Shortcut Model, which trains the marginal velocity and guides the shortcut mapping with a constructed CFG trajectory $\tilde{v}_\theta$.

In this case, however, we need to address a conflict: $\mathcal{L}_{\text{CFM}}$ enforces $F_{t,t} \approx v_t^*$ while $\mathcal{L}_{\text{SD-R}}$ enforces $F_{t,t} \approx \tilde{v}_\theta$.

iSD-C (Guidance-**C**onditional). To resolve this conflict, we append the guidance scale ω as an additional condition, $F_\theta(x_t; t, s, c, \omega)$. The modified objectives are given by:

$$\begin{aligned}
\mathcal{L}_{\text{CFM}} &= \mathbb{E}\left[\|F_\theta(x_t; t, t, c, 1.0) - v_t(x_t|x)\|_2^2\right], \\
\mathcal{L}_{\text{iSD-C}} &= \mathbb{E}\left[\|F_\theta(x_t; t, s, c, \omega) - \text{sg}[F_{\text{tgt}}(x_t; t, s, c, \omega)]\|_2^2\right],
\end{aligned}$$

where

$$\begin{aligned}
F_{\text{tgt}} &= F_\theta(x_t; t, s, c, \omega) + \left(A''_{t,s} x_t + A'_{t,s}\left(\tilde{v}_\theta(x_t; t, c) - F_\theta(x_t; t, s, c, \omega)\right) - A_{t,s} F'_\theta(x_t; t, s, c, \omega)\right), \\
\tilde{v}_\theta(x_t; t, c) &= F_\theta(x_t; t, t, \varnothing, 1.0) + \omega(F_\theta(x_t; t, t, c, 1.0) - F_\theta(x_t; t, t, \varnothing, 1.0)).
\end{aligned}$$

Hence, $\mathcal{L}_{\text{CFM}}$ ensures $F_\theta(x_t; t, t, c, 1.0) \approx v_t^*(x_t; c)$, while $\mathcal{L}_{\text{iSD-C}}$ ensures $F_\theta(x_t; t, t, c, \omega) \approx \tilde{v}_\theta(x_t; t, c)$. This choice is natural, as $\tilde{v}_\theta = v_t^*$ when $\omega = 1$.

iSD-T (**T**raining-time CFG). To resolve the conflict from another perspective, we adopt the training-time CFG described in Sec. A.9. As proven there, the training target $\tilde{v}_t(x_t|x) = (1-\omega)F_{\theta^-}(x_t; t, t, \varnothing) + \omega v_t(x_t|x)$ leads the flow matching objective toward the CFG velocity field. Accordingly, we modify the flow matching objective to follow the CFG velocity, distilling the approximated velocity into flow mappings:

$$\begin{aligned}
\mathcal{L}_{\text{CFM}} &= \mathbb{E}\left[\|F_\theta(x_t; t, t, c) - \tilde{v}_t(x_t|x)\|_2^2\right], \\
\mathcal{L}_{\text{iSD-T}} &= \mathbb{E}\left[\|F_\theta(x_t; t, s, c) - \text{sg}\left[F_{\text{tgt}}(x_t; t, s, c)\right]\|_2^2\right].
\end{aligned}$$

where $v_\theta(x_t; t) = F_\theta(x_t; t, t, c)$. In this case, the network approximates the CFG velocity, and the resulting flow mappings naturally follow the CFG trajectory. Unlike $\mathcal{L}_{\text{iSD-U}}$, the network does not compromise theoretical guarantees at $s = t$. We finalize our method by adopting iSD-T as the default configuration. The ablation studies are provided in Sec. C.3, and the training algorithm is described in Alg. 3.

Post-CFG. However, *Post-CFG*, defined as

$$\tilde{F}_\theta(x_t; t, s, c) = F_\theta(x_t; t, s, \varnothing) + \omega(F_\theta(x_t; t, s, c) - F_\theta(x_t; t, s, \varnothing)),$$

following Alg. 4, does not generally follow the exact CFG trajectory. This discrepancy arises from the definition of the flow map,

$$f_\theta(x_t; t, s) \approx x_t + \int_t^s v_\tau^*(x_\tau) d\tau,$$

which performs the path integral along a specific trajectory induced by v_τ^*. For a CFG trajectory, the path integral should be taken along $\tilde{v}_\tau^*$. In contrast, Post-CFG computes two separate forward passes, integrating along $v_\tau^*(x_\tau; c)$ and $v_\tau^*(x_\tau; \varnothing)$, rather than along $\tilde{v}_\tau^*$. As a result, the resulting update can differ from the expected CFG trajectory.

Algorithm 3 (iSD-T Training) Training algorithm of iSD-T

Input: Noise distribution p_Z, data distribution p_X, model F_θ, learning rate μ, time distribution τ, adaptive weighting (η, p), JVP approximation ϵ, Pre-CFG scale ω, class labels c.
repeat
  $z \sim p_Z, \quad x \sim p_X, \quad t, s \leftarrow \tau$
  $x_t \leftarrow \alpha_t x + \sigma_t z, \quad v_t \leftarrow \alpha_t' x + \sigma_t' z$
  $\tilde{v}_t = \omega v_t + (1 - \omega)\,\mathrm{sg}[F_\theta(x_t; t, t, \varnothing)]$
  $F_{t,s} \leftarrow F_\theta(x_t; t, s, c), \quad F_{t,t} \leftarrow F_\theta(x_t; t, t, c)$
  $F_{t,s}' \leftarrow [F_\theta(x_t + \epsilon F_{t,t}; t + \epsilon, s, c) - F_\theta(x_t - \epsilon F_{t,t}; t - \epsilon, s, c)]\,/(2\epsilon)$ ▷ JVP-Approx.
  $F_{\text{tgt}} \leftarrow F_{t,s} + \left(A_{t,s}'' x_t + A_{t,s}'(F_{t,t} - F_{t,s}) - A_{t,s} F_{t,s}'\right)$
  $\mathcal{L} \leftarrow \lambda_1 \|F_{t,t} - \tilde{v}_t\|_2^2 + \lambda_2 \|F_{t,s} - \mathrm{sg}[F_{\text{tgt}}]\|_2^2$ ▷ Optimization Target
  $\theta \leftarrow \theta - \mu \nabla_\theta \mathcal{L}$ ▷ Model Update
until Convergence

Algorithm 4 (Post-CFG Sampling) Sampling algorithm of iSD with Post-CFG

Input: Initial noise $z \sim p_Z$, model F_θ, Post-CFG scale ω, class labels c, sampling steps $\{t_i\}_{i=1}^{N+1}$.
$x \leftarrow z$
for $i \leftarrow 1$ to N **do**
  $\tilde{F}_{t,s} \leftarrow (1 - \omega) F_\theta(x; t_i, t_{i+1}, \varnothing) + \omega F_\theta(x; t_i, t_{i+1}, c)$ ▷ Post-CFG
  $x \leftarrow \nu^{-1}(A_{t_i,t_{i+1}}' x - A_{t_i,t_{i+1}} \tilde{F}_{t,s})$
end for

C Experimental Details

C.1 Reproducibility of Consistency Training

To evaluate the reproducibility of consistency training, we conducted experiments within the UCGM (Sun et al., 2025) framework. We compared the FID scores of several models trained under different initialization conditions. Following UCGM, we first extract latent representations of ImageNet-1K 256 × 256 using VA-VAE (Yao et al., 2025). All few-step models were trained with the same hyperparameters and settings: RAdam optimizer with a learning rate of 1e-4, weight decay of 0.0, $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch size of 1024, gradient clipping at 0.1, and timestep t sampled from Beta(0.8, 1.0). For enhancement, we applied a label

drop ratio of 0.1, an enhancement range of $(0, 0.75)$, and a ratio of 2.0. We also used the cosine function as the loss weighting function and trained all models with linear interpolation for 40K iterations.

Multistep Baseline. We trained the DiT-XL/1 architecture initialized from the publicly released multistep checkpoint of UCGM. This configuration achieved a 2-step FID of 2.52, which is reasonable but still falls short of the reported FID 1.42.

LightningDiT. We trained the LightningDiT-XL/1 architecture from its released pretrained model. In this setting, the model achieved a 2-step FID of 9.59, which is worse than the reported FID.

In-house Multistep Model. We trained the DiT-XL/2 architecture from scratch. For training, we used AdamW (Loshchilov & Hutter, 2019) with a learning rate of 0.0002, $\beta_1 = 0.9$, $\beta_2 = 0.95$, EMA decay weight of 0.999, and timestep t sampled from Beta(1, 1). We used an enhancement ratio of 0.47 and a cosine weighting function. After training the multistep model for 800k iterations, we conducted consistency training under the same few-step settings. This resulted in a 2-step FID 5.78.

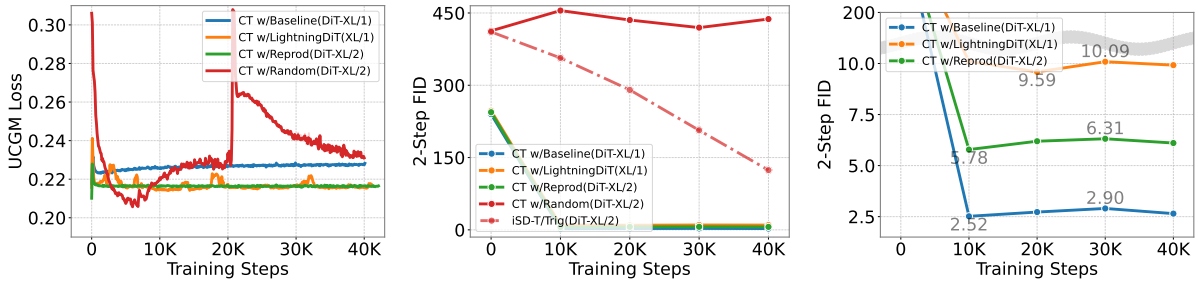


Figure 9: Training curves of consistency training. (a) Training loss over training steps. (b) Two-step FIDs over training steps. (c) Rescaled Y-axis for the consistency model with a preconditioner.

Without Preconditioner. We train a DiT-XL/2 architecture from randomly initialized weights without any preconditioner. In this case, training consistently failed, with the loss diverging and no meaningful samples being generated. While 40K steps may appear insufficient for scratch training, other scratch training methods already show a rapid decrease by 40K steps (Fig. 9). We consider that this is enough to check the unstable dynamics compared to other models.

These suggest that consistency training can be sensitive to initialization and the choice of preconditioner. It becomes unstable under random initialization and appears to substantially benefit from a well-trained multistep model. Even with pretrained initializations, consistency training demonstrates limited robustness and reproducibility across different architectures and setups.

C.2 Implementation

ImageNet 256×256. SD-VAE (Rombach et al., 2022) was used to encode images into a $32 \times 32 \times 4$ latent representation. For DiT (Peebles & Xie, 2023) models, we employed RMSNorm (Zhang & Sennrich, 2019), QK normalization (Henry et al., 2020), and RoPE (Su et al., 2023). Each model was scaled by depth and hidden dimension. We sampled t and s independently from Beta(0.8, 1.0), and set $t, s := \max(t, s), \min(t, s)$. For trigonometric interpolation, we additionally scale the timesteps as $t \leftarrow \frac{\pi}{2}t$ and $s \leftarrow \frac{\pi}{2}s$. For generations, we uniformly sampled the intermediate timesteps from the interval without additional engineering.

In ablation studies, DiT-B/4 was trained for 400K steps with a batch size of 256 and BF16 precision. For iSD-T, the corresponding wall-clock time is 32 GPU hours on a single A100 GPU. For large-scale experiments, DiT-XL/2 was trained for up to 800K steps, corresponding to 216 A100 GPU hours. We note that prior work (Sun et al., 2025; Zhou et al., 2025; Geng et al., 2025a; Peng et al., 2025) reports training from scratch for 800 or more epochs, corresponding to 1M or more steps with a larger batch size (e.g., 1024).

We also report results obtained with additional performance-oriented architectural choices. Specifically, we augment the input space for the guidance scale and the CFG range by sampling them during training rather

Table 9: Experimental Settings

Dataset	ImageNet-1K 256×256						CelebA-HQ 256×256
Preprocessor	SD-VAE (Rombach et al., 2022)				SD-VAE	VA-VAE (Yao et al., 2025)	SD-VAE
Input size	$32 \times 32 \times 4$				$32 \times 32 \times 4$	$16 \times 16 \times 32$	$32 \times 32 \times 4$
Condition	Class-conditional				Conditional	Conditional	Unconditional
Backbone	DiT-B/4	DiT-B/2	DiT-L/2	DiT-XL/2	DiT-XL/2†	DiT-XL/1†	DiT-B/2
Params (M)	131.33	131.33	459.42	676.78	682.02	681.77	131.33
Depth	12	12	24	28	28	28	12
Hidden dim	768	768	1024	1152	1152	1152	768
Heads	12	12	16	16	16	16	12
Patch size	4×4	2×2	2×2	2×2	2×2	1×1	2×2
Interpolation	Trigonometric				Linear	Linear	Trigonometric
Self-distillation	$\mathcal{L}_{\text{iSD-T}}$				$\mathcal{L}_{\text{iSD-T}}$	$\mathcal{L}_{\text{iSD-T}}$	$\mathcal{L}_{\text{iSD}}$
Joint training	Enabled				Enabled	Enabled	Enabled
JVP	Approximation				Approx.	Approx.	Approx.
Weighting	Adaptive				Adaptive	Cosine	Adaptive
ϵ	0.005				0.005	0.005	0.005
p	1.0				1.0	-	1.0
η	0.01				0.01	-	0.01
Pre-CFG ω	5.0				Randomly sampled	Randomly sampled	-
FP Precision	BF16				BF16	BF16	BF16
Batch size	256				256	256	256
Label dropout	0.1				0.1	0.1	-
Optimizer	AdamW (Loshchilov & Hutter, 2019)				NorMuon (Li et al., 2025)	NorMuon	AdamW
LR Scheduler	Constant				Constant	Constant	Constant
β_1	0.9				-	-	0.9
β_2	0.999				0.95	0.95	0.999
Momentum	-				0.95	0.95	-
Learning rate	1e-4				5e-4	5e-4	1e-4
Training steps	400K				800K	600K	200K
Weight decay	0				0	0	0
EMA decay	0.99995				0.99995	0.99995	0.99995

than using fixed values, following the design choices introduced in Geng et al. (2025b). The guidance scale is sampled from a truncated exponential distribution, and the CFG range is sampled from a uniform distribution. These modifications are orthogonal to our analysis and are used only to demonstrate that the proposed objective remains stable under commonly adopted performance-oriented settings. For conditioning, we concatenate conditions in the input token space. We denote this configuration with SD-VAE by DiT-XL/2† and VA-VAE by DiT-XL/1†. Detailed training parameters are provided in Tab. 9, and the corresponding results in Tab. 7 are reported under this setting.

CelebA-HQ. For CelebA-HQ, we use SD-VAE to encode images into 32×32×4 latent representations. The model is based on DiT-B/2 in an unconditional setting, as in the Shortcut Model. It is trained on four RTX 3090 GPUs with a global batch size of 256. It takes 2 days for 200K steps (176 RTX 3090 GPU hours). Detailed settings are provided in Tab. 9.

The choice of interpolation. This reflects different geometric assumptions. Linear interpolation is aligned with the optimal transport path under a Euclidean cost. This is well-suited for tasks such as Push-T (Sec. C.5), where the end-effector coordinates are defined in Euclidean space. In contrast, trigonometric interpolation is associated with a geodesic on a spherical manifold. It is more appropriate for tasks involving normalized representations (Chen & Lipman, 2024), such as image generation in certain latent spaces.

We note that these choices are not strict requirements. In practice, sufficiently expressive networks can compensate for geometric mismatch (Zheng et al., 2026). Therefore, we view the choice of interpolation primarily as an inductive bias, rather than a hard constraint.

C.3 Additional Ablation Study

Direct Intervention. To demonstrate the role of guiding velocities, we directly perturb the guiding velocity and observe the resulting flow maps. We use marginal velocity as the baseline guidance, and control a proxy

Table 10: Direct intervention on guiding velocity by interpolating marginal velocity ($\gamma = 0$) and conditional velocity ($\gamma = 1$). The marginal velocity guidance shows lower FID and lower loss variance. (Left) Toy experiments following Fig. 2. We measure the optimality of flow maps using the proxy measure $\mathcal{L}_{\text{ED}}$. (Right) Experiments in the ImageNet ablation setting. We report FIDs and the standard deviations σ of each loss landscape, following Fig. 4.

(a) $\mathcal{L}_{\text{ED}}$ in toy experiments

γ	1.00	0.75	0.50	0.25	0.00
$\mathcal{L}_{\text{ED}}$	0.34	0.24	0.17	0.12	0.10
$\mathcal{L}_{\text{CT}}$	0.16	0.15	0.15	0.12	0.10

(b) FID and σ in ablation settings

γ	1.0	0.5	0.0
FID	124.5	112.42	110.38
σ	204.07	92.43	61.94

Table 11: Ablation study for hyperparameters. We set $\epsilon = 0.005$, $\eta = 0.01$, $p = 1.0$, and NFE=2 by default.

(a) JVP approximation step size ϵ

ϵ	0.01	0.005	0.001	0.0001
FID	14.74	14.62	15.03	29.29

(b) Adaptive weighting bias η

η	1.0	0.1	0.01	0.001	0.0001
FID	14.68	14.58	14.62	14.34	14.53

(c) Adaptive weighting exponential p

p	0.0	0.5	1.0	1.5	2.0
FID	31.85	18.03	14.62	13.66	13.49

(d) Sampling steps (NFE)

NFE	1	2	4	8
FID	24.18	14.62	14.31	14.43

for the noise level by defining $v_t^\gamma(x_t) = (1-\gamma)v_t^*(x_t) + \gamma v_t(x_t|x)$, interpolating between marginal velocity v_t^* and noisy conditional velocity $v_t(x_t|x)$.

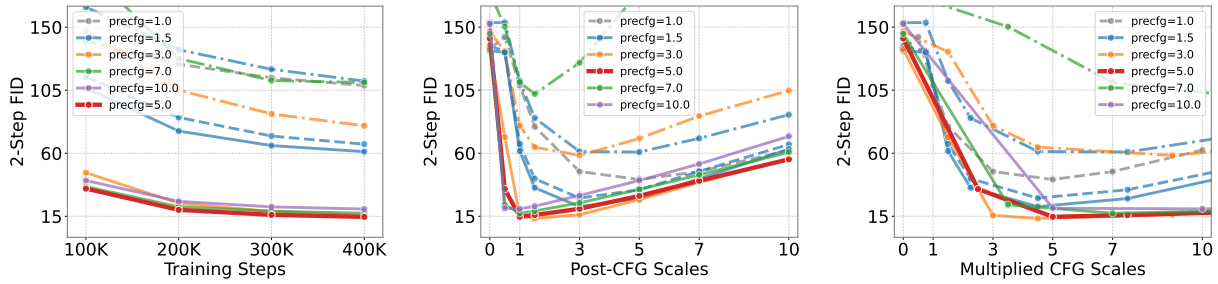
In a toy setting, we measure optimality as $\mathcal{L}_{\text{ED}}$ with v_t^* following Fig. 3d, as shown in Tab. 10a. For Eulerian distillation $\mathcal{L}_{\text{ED}}$ and consistency training $\mathcal{L}_{\text{CT}}$ along v_t^γ, optimality improves monotonically as γ decreases, i.e., as the guiding velocity becomes closer to the marginal velocity. On ImageNet-1K with DiT-B/4, reducing γ improves FID. We measure instability via the standard deviation of each loss landscape, and it also decreases as γ decreases.

These results provide empirical evidence consistent with that the gradient difference induced by the guiding velocity can affect optimization and is associated with suboptimality (Prop. 4.2) and instability (Prop. 4.3).

Hyperparameters. In Tab. 11, we explore hyperparameters in the ablation setting: iSD-T on DiT-B/4 with trigonometric interpolation and $\omega = 5.0$. The default value of $\epsilon = 0.005$ follows Sun et al. (2025), and default values of $\eta = 0.01$ with $p = 1.0$ follow Geng et al. (2025a). We observe that FID remains stable for $\epsilon \in [0.001, 0.01]$, while extremely small values can degrade performance, likely due to numerical instability in lower-precision settings. For adaptive weighting parameters, performance also varies smoothly across η and for nonzero p. We also observe that performance improves from 1 to 2 steps, and remains nearly plateaued.

This suggests that the method may not be highly sensitive within a reasonable range of hyperparameters. We use a single default setting across all experiments without additional tuning, including both image and policy generation. This supports that the method may not require extensive hyperparameter search.

JVP Approximation. To assess the generality of JVP approximation, we additionally apply it to MeanFlow, beyond its use in iSD. Under the ablation setting, training MeanFlow with the exact JVP requires approximately 25GB of VRAM and 58 A100 GPU hours. With the approximation, the VRAM usage decreases to 11GB and the wall-clock time to 21.5 GPU hours. Under this setting, the FID decreases to 10.74. Note that in Fully Sharded Data Parallel setups, while integrating exact JVP may require additional engineering effort, this approximation can be incorporated with lower implementation overhead.



(a) Variants of Pre-CFG methods (b) Post-CFG scales (c) Multiplied CFG scales

Figure 10: Variants of Pre-CFG. Dash, dash-dotted, and solid lines indicate iSD-U, iSD-C, and iSD-T, respectively. Each color represents a different Pre-CFG scale. Trigonometric interpolation and the JVP approximation are used by default on DiT-B/4. (a) FIDs of Pre-CFG variants over training steps. (b) FIDs of Post-CFG over guidance scales. (c) FIDs over multiplied guidance scales.

Time-condition Annealing. Since our training dynamics can be divided into two stages: learning marginal velocity and flow mapping guided by itself, we also consider annealing of the time condition s, sampled near t for the initial stage and then warmed up to $0 \leq s < t$ as training continues. To linearly warm up the distribution of s, we first sample s and t following the baseline and reformulate s as:

$$s = t - (t - s) \times \min\left(1.0, \frac{\text{steps}}{\text{max_warmup}}\right).$$

We set `max_warmup` to 10K steps, corresponding to the first knee point of the flow matching term. However, time-condition annealing does not noticeably affect the training dynamics, and the resulting FID of 29.65 is worse than the baseline.

To interpret this phenomenon, we consider two points: (i) we adopt adaptive weighting, which covers both flow matching and self-distillation objectives, and (ii) in the initial training stages, the loss of flow matching remains around 15, while that of self-distillation is near 0.5. This suggests that the flow matching objective already dominates the training signal, and thus the intended effect of annealing becomes diluted.

Variants of Pre-CFG. In the ablation setting, as shown in Fig. 10a and Tab. 14, applying Pre-CFG $\mathcal{L}_{\text{iSD-U}}$ with $\omega = 1.5$ improves FIDs compared to vanilla $\mathcal{L}_{\text{iSD}}$. However, training diverges when $\omega > 3.0$. This instability is consistent with the conflict discussed in Sec. B.4, where increasing the guidance scale can make the self-distillation target less aligned with the flow matching objective in iSD-U settings.

When the guidance scale is appended as a condition, $\mathcal{L}_{\text{iSD-C}}$ enables training at higher guidance scales, outperforming $\mathcal{L}_{\text{iSD}}$. However, $\mathcal{L}_{\text{iSD-C}}$ consistently underperforms $\mathcal{L}_{\text{iSD-U}}$. Intuitively, the additional condition forces the network to learn both CFG and non-CFG mappings, which may impose an extra burden on the network. Even though $\mathcal{L}_{\text{iSD-U}}$ compromises the theoretical guarantees at $s = t$, few-step generation commonly assumes $s \ll t$, resulting in comparable performance to iSD-T and making the issue less critical in practice. On the other hand, iSD-T neither introduces the conflict nor imposes additional burden on the network. It consistently outperforms both variants. Therefore, we adopt iSD-T as the default.

We also examined Post-CFG, as illustrated in Fig. 10b. iSD-U shows improved results when Post-CFG is applied, particularly at $\omega = 3.0$. Applying Post-CFG on top of Pre-CFG can be interpreted as applying CFG twice, leading to a multiplied guidance scale $4.5 = 1.5 \times 3.0$. Similarly, iSD-T achieves its lowest FIDs around a Pre-CFG scale of $\omega = 5.0$, while Post-CFG on it underperforms even when the scale is close to one. When applying iSD-T with $\omega = 3.0$, Post-CFG outperforms at $\omega = 1.5$, corresponding to a total scale of 4.5. *These consistent observations suggest that the effective CFG scale lies near 5.0 in this ablation setting.* We refer to this empirical tendency as effective-scale behavior.

We note that iSD-C demonstrates effective CFG scales between 7.5-10.5, distinct from the ω-unconditional settings (iSD-U and iSD-T). However, even within its optimal range, iSD-C underperforms them.

Pre-CFG Scale for larger networks. For iSD-T, we investigate the effective guidance scale for the larger network, DiT-XL/2, as shown in Tab. 14. We train DiT-XL/2 with $\omega = 5.0$ and measure metrics across Post-CFG scales (Tab. 12).

Table 12: FIDs and ISs across Post-CFG scales for DiT-XL/2 with iSD-T ($\omega = 5.0$, 600K steps).

Post-CFG ω	0.0	0.5	1.0	1.5	3.0	5.0	7.0	10.0
FID	100.89	**11.19**	16.96	18.26	19.33	25.83	35.59	48.50
IS	13.53	164.48	**234.85**	229.35	181.94	116.25	85.75	65.36

DiT-XL/2 shows better results at the Post-CFG scale of $\omega = 0.5$, suggesting an effective Pre-CFG scale near 2.5. As reported in Tab. 14, the Pre-CFG scale of $\omega = 3.0$ outperforms $\omega = 5.0$, supporting the effective-scale behavior. This indicates that *the effective Pre-CFG scale may not generalize to larger networks.* A similar trend has been observed in prior work, where different CFG scales were used across architectures.

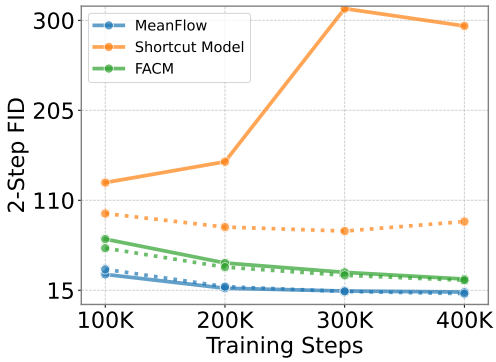


Figure 11: FIDs on ImageNet. Solid lines indicate $\omega = 5.0$, and dotted lines indicate the ω values in Tab. 5.

Pre-CFG Scale for prior work. In Tab. 5, we present results using the CFG scale reported as optimal in each work. For direct comparison, we train these methods with a fixed scale of $\omega = 5.0$, as shown in Tab. 13 and Fig. 11. Under this setting, the FIDs of MeanFlow and FACM increase relative to their reported scales. For the Shortcut Model, we did not observe stable convergence at larger guidance scales in our setup, and therefore report results using the recommended scale.

Table 13: Comparison at $\omega = 5.0$.

Case	FID↓	IS↑
MeanFlow	13.12	220.9
Shortcut Model	128.7	6.57
FACM	26.85	85.05

This behavior is consistent with our observations for iSD-U. As discussed in Sec. B.4, both CFG formulations share a common structure: they replace the guidance velocity with its CFG counterpart, while leaving the flow matching objective unchanged. At large guidance scales, this structural discrepancy can introduce conflicts between the consistency and flow matching objectives, potentially degrading training stability.

C.4 Comparison with Other Work

CelebA-HQ. We conduct additional comparisons on CelebA-HQ to validate our method in an unconditional setting. For fair comparison, all methods in Tab. 6 are trained with the DiT-B/2 architecture. As shown in Tab. 6, iSD achieves competitive results on both few-step (4-step) and multi-step (128-step) generation. Since we train for 200K steps in this case, iSD provides an efficient training configuration, while the Shortcut Model uses 800K steps.

Shortcut Model. To compare the reproducibility with shortcut models, we train the network under the same ablation settings. We utilize an EMA network as a teacher with its CFG approach at $\omega = 1.5$. Note that its guidance method shares an idea with iSD-U (Sec. B.4). Under this setup, iSD-U attains stronger quantitative results, as shown in Tab. 14.

Figure 12: 4-step samples from iSD on CelebA-HQ 256×256.

We hypothesize that the performance gap between the Shortcut Model and iSD-U may arise from the step-size discretization in the Shortcut Model, which introduces an asymptotic gap relative to continuous-time formulations. Given the behaviors seen in iSD-U under different ω values, it is plausible that the Shortcut Model may experience increased training sensitivity as ω becomes larger. As iSD-T shows the best results and provides solid theoretical grounding, applying the iSD-T Pre-CFG scheme to shortcut approaches provides a promising future direction.

FACM. As a concurrent work, FACM (Peng et al., 2025) suggests *flow-anchoring*, which jointly trains the network with the flow matching objective to provide an anchored training signal. This anchored signal is introduced to prevent degenerate model collapse and enable training from scratch. To compare with our method, we train FACM under our ablation settings. As shown in Tab. 14, FACM also reduces the variance of the resulting metrics compared to the baseline, and is on par with our methods in reproducibility. It can be viewed as a weak version of the time-condition relaxation, which relaxes the condition to $s \in \{0, t\}$ rather than $s = 0$. Not only does this align with our hypothesis, but Fig. 14 also demonstrates that the joint training of flow matching exhibits lower loss spikes and variance, making the training from scratch easier. This supports the effect of flow-anchoring in the landscape perspective, and further supports our strong version of the relaxation.

improved MeanFlow. In line with recent work, improved MeanFlow (iMF; Geng et al., 2025b) reorganizes the MeanFlow identity into a v-loss formulation and highlights that conditional velocity leakage can arise when formalizing instantaneous velocities from average velocity (u-prediction). They note that this leakage contributes to training instability, and that replacing conditional velocity guidance with the model's own marginal velocity prediction (v_ψ) can improve stability. Under linear interpolation, the gradient of the resulting objective can be aligned with our formulation at the expectation level:

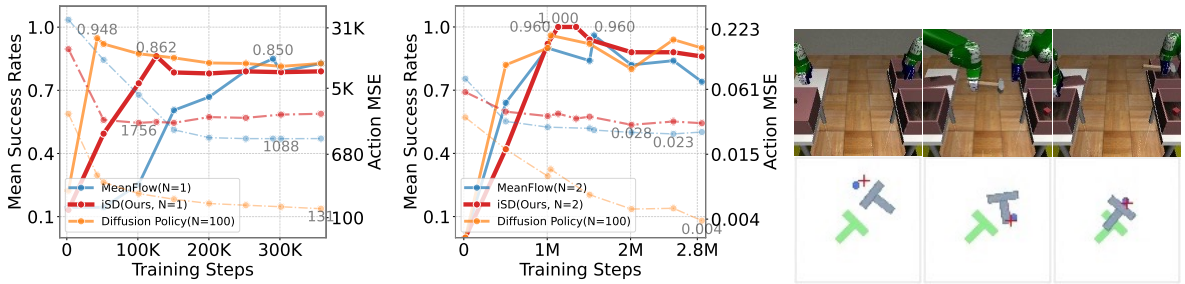
$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{iMF}} &\propto \mathbb{E}\left[\nabla_\theta F_\theta(x_t;t,s)\cdot\left(F_\theta(x_t;t,s) - v_t(x_t|x) + (t-s)\cdot\left(\partial_t F_\theta + v_\psi(x_t;t)\cdot\nabla_x F_\theta\right)\right)\right] \\
&\approx \mathbb{E}\left[\nabla_\theta F_\theta(x_t;t,s)\cdot\left(F_\theta(x_t;t,s) - v_t^*(x_t) + (t-s)\cdot\left(\partial_t F_\theta + v_t^*(x_t)\cdot\nabla_x F_\theta\right)\right)\right], \\
&\approx \mathbb{E}\left[\nabla_\theta F_\theta(x_t;t,s)\cdot\left(F_\theta(x_t;t,s) - F_{\theta^-}(x_t;t,t) + (t-s)\cdot\left(\partial_t F_\theta + F_{\theta^-}(x_t;t,t)\cdot\nabla_x F_\theta\right)\right)\right] \\
&\propto \nabla_\theta \mathcal{L}_{\text{SD-R}},
\end{aligned}$$

where $v_\psi(x_t;t) \approx v_t^*(x_t)$ is a velocity prediction head of the iMF network. We note that the training target of iMF is based on the conditional velocity, while the training target of our $\mathcal{L}_{\text{SD-R}}$ is based on the marginal velocity. This makes the relation hold only at the expectation level. While the motivation and analytical perspective of iMF differ from ours, both approaches arrive at marginal velocity guidance through independent reasoning.

C.5 Policy Generation

To assess the applicability of iSD to diffusion-based policy learning, we train MeanFlow and iSD to imitate proficient human controls on two tasks, Push-T and Transport, using transformer- and state-based policies in simulation environments.

Unlike diffusion policy, iSD and MeanFlow require additional conditioning on s. Following prior work, which concatenates t to the observation context, we extend it by additionally appending s. Other settings follow prior work (Chi et al., 2023).



(a) Push-T (b) Transport (c) Simulation Results

Figure 13: Results of policy learning. The left two figures show average success rates (solid lines) and action mean-squared errors (dashed-dot lines) over training steps. Numeric entries indicate the optimal values across training steps. The rightmost figure shows the simulation results of iSD with NFE=2.

As shown in Tab. 8, iSD with linear interpolation achieves results comparable to those of other few-step methods, and performs particularly well on the Transport task. With 2-step sampling, iSD achieves performance comparable to that of the diffusion policy, which uses 100 sampling steps. Even with 2-step sampling, MeanFlow achieves an average success rate of 0.96, which is lower than iSD's. This suggests that iSD can be a reasonable candidate for policy generation.

In Fig. 13, we observe that iSD consistently exhibits higher action error than the other methods. However, it achieves success rates comparable to or better than those of other approaches. Even though the diffusion policy achieves a low action error of 0.004 on the Transport task, its peak success rate occurs earlier in training. In this case, the action error does not fully reflect the success rate, and iSD can achieve higher success rates despite yielding higher action errors than other methods.

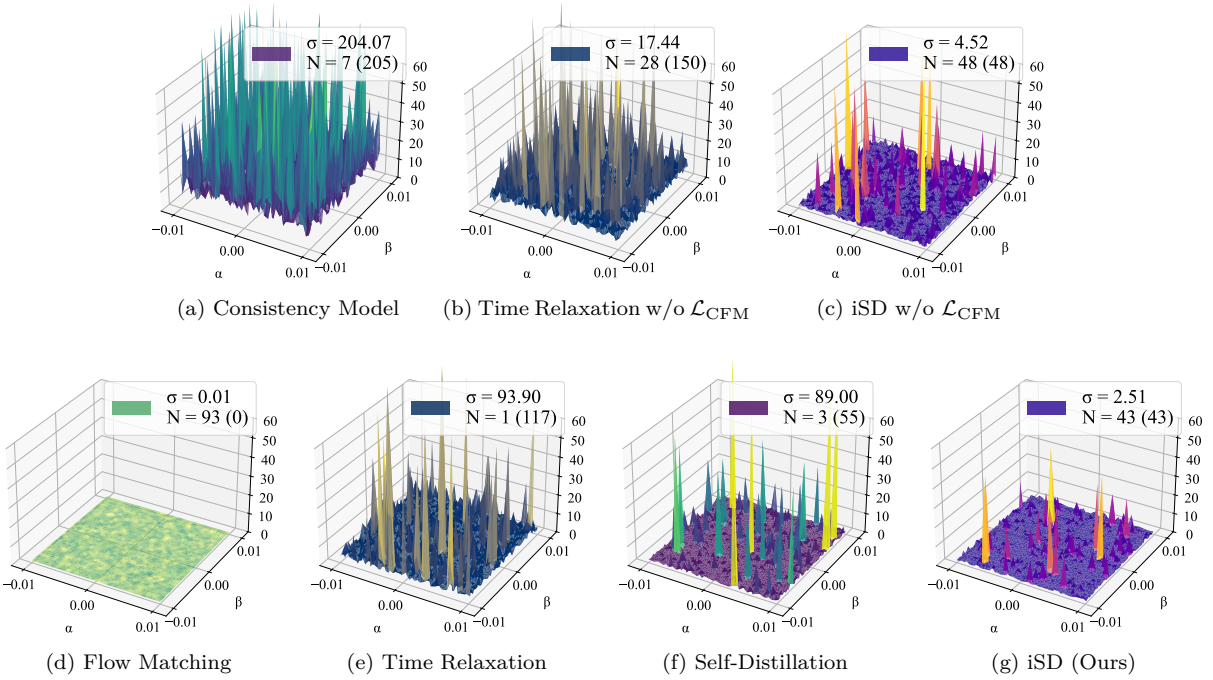
We note that trigonometric interpolation underperforms linear interpolation under iSD settings, achieving an average success rate of 0.605 on Push-T with NFE=2. This is consistent with the observation that interpolation exhibits varying performance across settings, and that linear interpolation appears to be a more suitable choice for policy generation in this setting.

C.6 Loss Landscape

To assess the stability of iSD, we conduct loss landscape analysis using PyHessian (Yao et al., 2020) on ImageNet-trained DiT-B/4. For each objective, we compute the top-2 eigenvectors of its Hessian to explore the landscape along the maximum curvature. To probe different training stages, we compute the Hessian at 0K and 200K training steps and visualize the landscapes in Fig. 14 and Fig. 15, respectively. We measure the loss variance and count spikes by identifying outliers beyond an empirical 95% range, providing empirical support for the linearization cost hypothesis.

In the upper rows, we first examine each objective without joint training with flow matching. As shown in Fig. 14 and Fig. 15, time-condition relaxation reduces loss spikes and variance, supporting the linearization cost hypothesis. In Fig. 14c, iSD shows even lower variance. These observations suggest that, independent of joint training with flow matching, both reduced linearization cost and marginal velocity guidance are associated with a more stable optimization landscape.

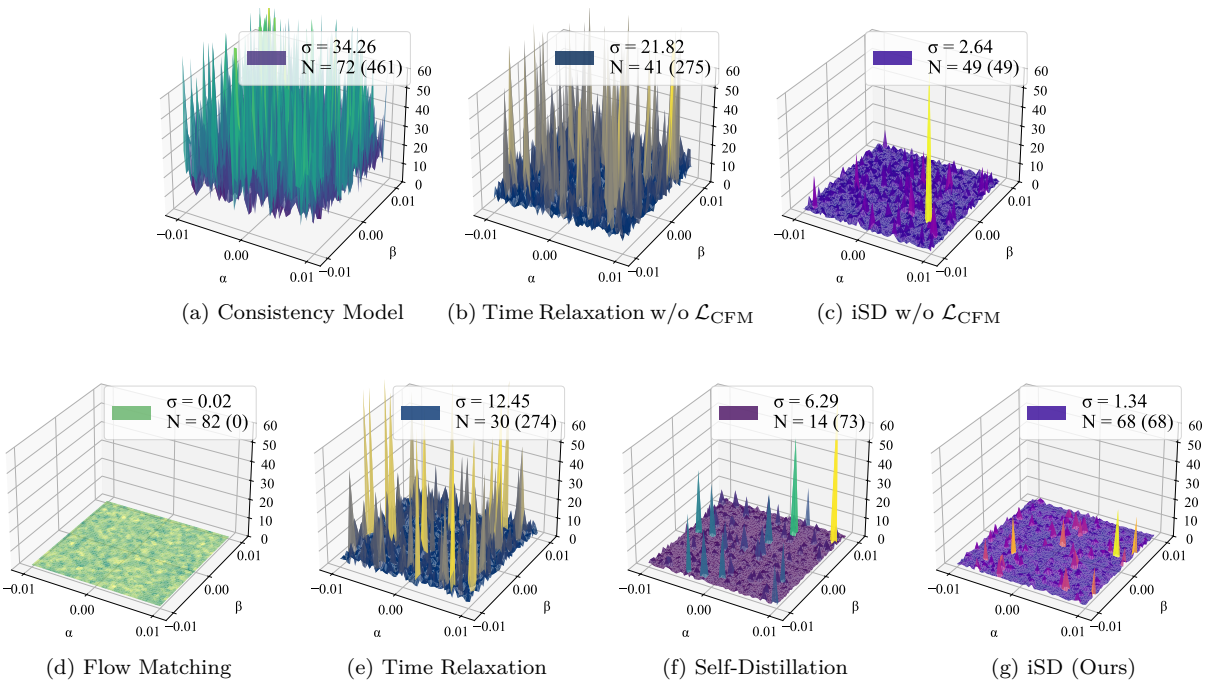
After joint training with flow matching, as shown in the lower row, variances and spikes decrease, leading to more stable training and potentially improved reproducibility. Across both training steps, the empirical tendencies remain consistent: time-condition relaxation and self-distillation contribute to more stable consistency training.



(a) Consistency Model (b) Time Relaxation w/o $\mathcal{L}_{\text{CFM}}$ (c) iSD w/o $\mathcal{L}_{\text{CFM}}$

(d) Flow Matching (e) Time Relaxation (f) Self-Distillation (g) iSD (Ours)

Figure 14: Loss landscapes of four methods at 0K training steps. α and β denote the top-2 eigenvectors of the Hessian on ImageNet-1K with DiT-B/4. σ denotes the standard deviations of the landscape fields, and N the number of samples outside each method's own empirical 95% range. Values in parentheses report the number of samples exceeding the 95% bound defined by iSD, as a common reference.



(a) Consistency Model (b) Time Relaxation w/o $\mathcal{L}_{\text{CFM}}$ (c) iSD w/o $\mathcal{L}_{\text{CFM}}$

(d) Flow Matching (e) Time Relaxation (f) Self-Distillation (g) iSD (Ours)

Figure 15: Loss landscapes at 200K training steps. Other settings are identical to those in Fig. 14.

Table 14: Quantitative results (2-NFE). The numeric entries in the header denote training steps. Reported results are averaged over three random seeds unless otherwise specified; entries marked with an underline are averaged over five seeds. *Std.* denotes the standard deviation at 400K steps (and at 800K steps for the bottom table). Hyphens in *Std.* indicate cases evaluated with a single run, where standard deviation was not computed. *HALTED* denotes cases where training diverged and was stopped earlier than the baseline.

(a) Frechet Inception Distance

Loss	Interp.	JVP	Arch.	Pre-CFG ω	100K	200K	300K	400K	Std.
Baseline	Linear	Exact	DiT-B/4	-	137.84	127.23	122.76	120.49	<u>3.71</u>
Joint-training	Linear	Exact	DiT-B/4	-	132.84	105.90	94.94	96.28	<u>15.86</u>
$\mathcal{L}_{\text{CT}}$	Linear	Exact	DiT-B/4	-	194.95	142.92	73.32	69.18	<u>2.84</u>
$\mathcal{L}_{\text{CT}}$	Trig	Exact	DiT-B/4	-	131.12	110.28	102.79	97.84	3.40
$\mathcal{L}_{\text{iSD}}$	Linear	Exact	DiT-B/4	-	155.30	128.15	119.98	114.56	<u>2.80</u>
$\mathcal{L}_{\text{iSD}}$	Trig	Exact	DiT-B/4	-	147.01	125.83	117.19	110.18	4.04
$\mathcal{L}_{\text{CT}}$	Linear	Approx	DiT-B/4	-	94.53	74.23	67.64	63.02	<u>0.42</u>
$\mathcal{L}_{\text{CT}}$	Trig	Approx	DiT-B/4	-	223.10	227.90	228.70	Halted	-
$\mathcal{L}_{\text{iSD}}$	Linear	Approx	DiT-B/4	-	146.37	122.32	113.79	108.70	<u>2.83</u>
$\mathcal{L}_{\text{iSD}}$	Trig	Approx	DiT-B/4	-	139.95	123.56	113.88	108.33	-
$\mathcal{L}_{\text{iSD-U}}$	Linear	Approx	DiT-B/4	1.5	124.94	93.20	82.15	75.53	1.48
$\mathcal{L}_{\text{iSD-U}}$	Trig	Approx	DiT-B/4	1.5	119.86	86.21	74.48	67.66	0.83
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	1.5	166.93	128.24	110.30	100.47	-
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	3.0	200.17	120.56	98.73	88.93	-
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	7.0	270.50	Halted	-		
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	1.5	164.34	133.90	119.94	111.56	-
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	3.0	147.15	105.09	88.08	79.64	-
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	7.0	192.14	127.65	111.93	110.49	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	1.5	115.18	83.02	70.88	63.81	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	3.0	61.77	39.68	31.80	27.62	0.40
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	5.0	51.77	35.20	29.81	26.91	<u>0.50</u>
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	7.0	51.88	36.85	31.46	28.61	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	10.0	54.90	41.77	37.07	33.86	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	1.5	106.54	75.85	65.50	61.19	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	3.0	46.01	24.35	18.42	15.56	0.12
$\mathcal{L}_{\text{iSD-T}}$	Trig(Ours)	Approx	DiT-B/4	5.0	34.97	19.56	16.02	14.62	<u>0.06</u>
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	7.0	36.17	21.83	18.72	17.17	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	10.0	40.54	25.69	21.75	20.18	-
MeanFlow	Linear	Exact	DiT-B/4	3.0	37.11	18.84	13.73	11.48	<u>1.68</u>
MeanFlow	Trig	Exact	DiT-B/4	3.0	48.36	24.73	18.19	15.08	3.11
Shortcut Model	Linear	Exact	DiT-B/4	1.5	96.02	81.74	77.51	87.52	<u>24.34</u>
FACM	Linear	Exact	DiT-B/4	1.75	59.45	39.41	30.79	25.52	<u>1.73</u>
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/2	3.0	36.40	19.20	13.83	11.92	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/2	5.0	31.00	18.96	16.75	15.93	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-L/2	3.0	26.02	14.86	12.11	11.17	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	3.0	24.69	14.33	11.82	11.08	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	5.0	28.30	18.89	17.33	17.03	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2†	6.0 (Eval)	14.53	5.82	4.30	3.79	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/1†	6.0 (Eval)	7.15	3.01	2.64	2.52	-

Loss	Interp.	JVP	Arch.	Pre-CFG ω	500K	600K	700K	800K
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	3.0	10.66	10.48	10.29	10.30
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	5.0	16.87	16.85	16.96	
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2†	6.0 (Eval)	3.47	3.28	3.09	2.98
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/1†	6.0 (Eval)	2.40	2.30		

(b) Inception Score

Loss	Interp.	JVP	Arch.	Pre-CFG ω	100K	200K	300K	400K	Std.
Baseline	Linear	Exact	DiT-B/4	-	7.90	8.88	9.40	9.69	<u>0.47</u>
Joint-training	Linear	Exact	DiT-B/4	-	10.95	13.01	14.32	14.11	<u>4.15</u>
$\mathcal{L}_{\text{CT}}$	Linear	Exact	DiT-B/4	-	8.76	11.34	19.78	21.45	<u>1.21</u>
$\mathcal{L}_{\text{CT}}$	Trig	Exact	DiT-B/4	-	11.77	15.04	16.79	18.11	0.98
$\mathcal{L}_{\text{iSD}}$	Linear	Exact	DiT-B/4	-	7.21	9.60	10.43	11.04	<u>0.41</u>
$\mathcal{L}_{\text{iSD}}$	Trig	Exact	DiT-B/4	-	9.37	12.02	13.53	14.75	0.70
$\mathcal{L}_{\text{CT}}$	Linear	Approx	DiT-B/4	-	14.86	19.50	21.92	24.08	<u>0.29</u>
$\mathcal{L}_{\text{CT}}$	Trig	Approx	DiT-B/4	-	3.76	3.85	3.89	Halted	-
$\mathcal{L}_{\text{iSD}}$	Linear	Approx	DiT-B/4	-	8.18	10.48	11.21	11.80	<u>0.36</u>
$\mathcal{L}_{\text{iSD}}$	Trig	Approx	DiT-B/4	-	10.02	11.80	13.68	14.93	-
$\mathcal{L}_{\text{iSD-U}}$	Linear	Approx	DiT-B/4	1.5	10.60	15.38	17.70	19.63	0.45
$\mathcal{L}_{\text{iSD-U}}$	Trig	Approx	DiT-B/4	1.5	12.72	20.71	25.57	29.26	0.52
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	1.5	6.45	9.33	11.18	12.54	-
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	3.0	4.88	10.16	12.55	13.98	-
$\mathcal{L}_{\text{iSD-C}}$	Linear	Approx	DiT-B/4	7.0	1.83	Halted	-		
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	1.5	8.04	10.95	12.58	13.88	-
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	3.0	9.14	14.30	17.17	18.84	-
$\mathcal{L}_{\text{iSD-C}}$	Trig	Approx	DiT-B/4	7.0	4.69	10.79	12.42	13.12	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	1.5	11.87	17.14	20.84	23.85	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	3.0	24.94	41.33	53.74	63.90	0.99
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	5.0	30.43	50.69	63.82	74.70	<u>2.43</u>
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	7.0	31.42	50.69	63.81	75.09	-
$\mathcal{L}_{\text{iSD-T}}$	Linear	Approx	DiT-B/4	10.0	28.87	42.85	53.79	63.18	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	1.5	15.68	25.63	31.76	35.41	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	3.0	39.45	84.45	114.30	135.60	0.19
$\mathcal{L}_{\text{iSD-T}}$	Trig(Ours)	Approx	DiT-B/4	5.0	53.20	117.46	157.83	184.53	<u>0.77</u>
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	7.0	52.79	117.45	157.02	186.79	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/4	10.0	46.28	100.12	137.87	164.59	-
MeanFlow	Linear	Exact	DiT-B/4	3.0	45.38	100.49	140.32	167.60	<u>20.39</u>
MeanFlow	Trig	Exact	DiT-B/4	3.0	39.22	89.40	123.32	147.18	22.16
Shortcut Model	Linear	Exact	DiT-B/4	1.5	13.58	15.75	16.25	14.94	<u>3.68</u>
FACM	Linear	Exact	DiT-B/4	1.75	24.05	39.32	52.87	65.42	<u>4.84</u>
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/2	3.0	44.96	93.25	132.21	156.90	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-B/2	5.0	56.63	122.93	153.30	170.99	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-L/2	3.0	61.29	122.22	159.68	178.42	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	3.0	64.95	127.97	165.02	185.29	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	5.0	66.44	140.28	179.87	201.29	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2†	6.0 (Eval)	97.80	172.30	204.08	219.71	-
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/1†	6.0 (Eval)	200.26	278.90	295.43	304.74	-

Loss	Interp.	JVP	Arch.	Pre-CFG ω	500K	600K	700K	800K
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	3.0	199.70	212.66	225.43	233.06
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2	5.0	220.07	233.32	234.85	
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/2†	6.0 (Eval)	235.11	245.51	251.24	257.96
$\mathcal{L}_{\text{iSD-T}}$	Trig	Approx	DiT-XL/1†	6.0 (Eval)	307.30	313.50		

D Qualitative Results

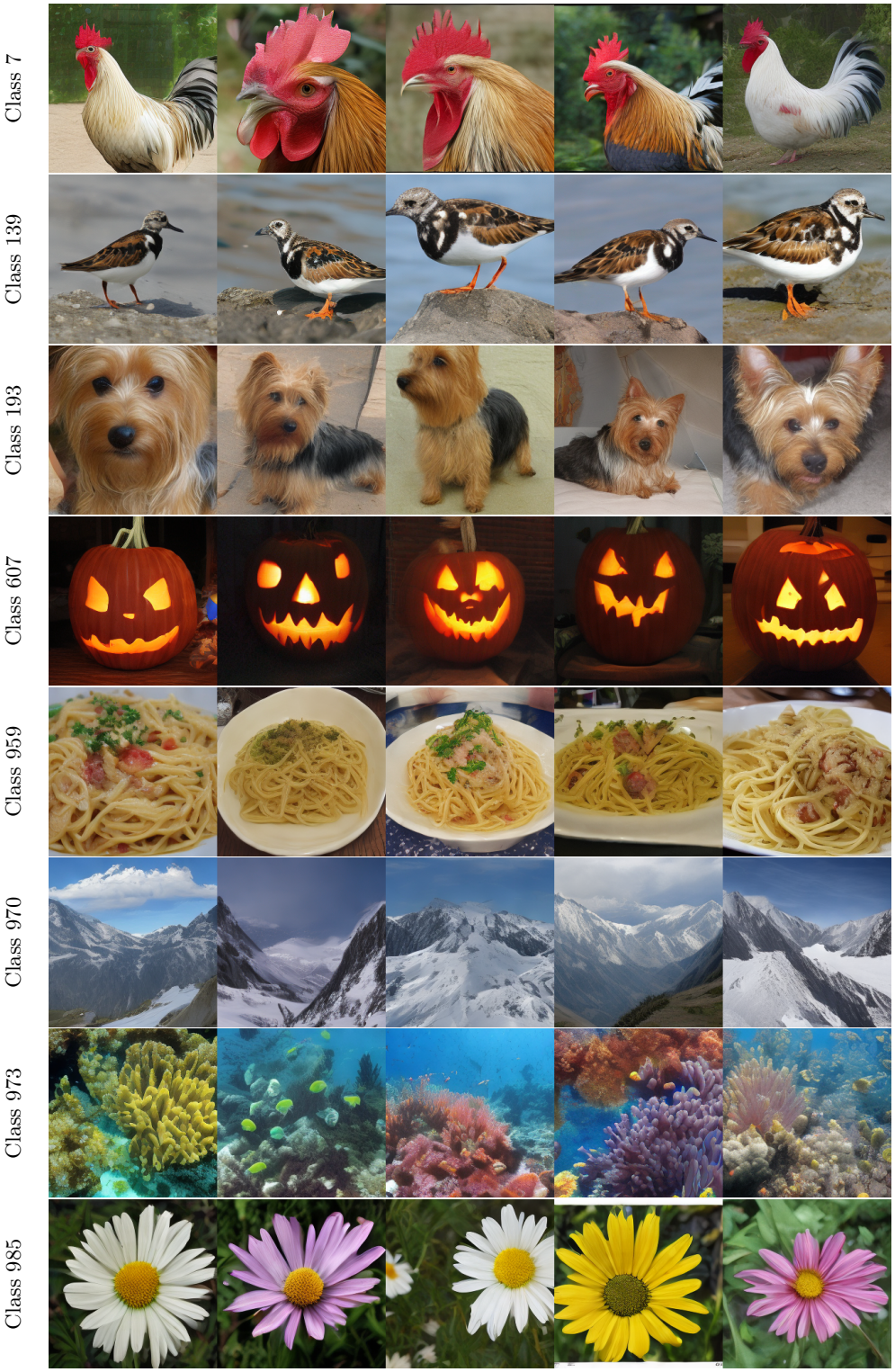


Figure 16: Class-level samples generated by iSD-T with two-step sampling on ImageNet 256×256.

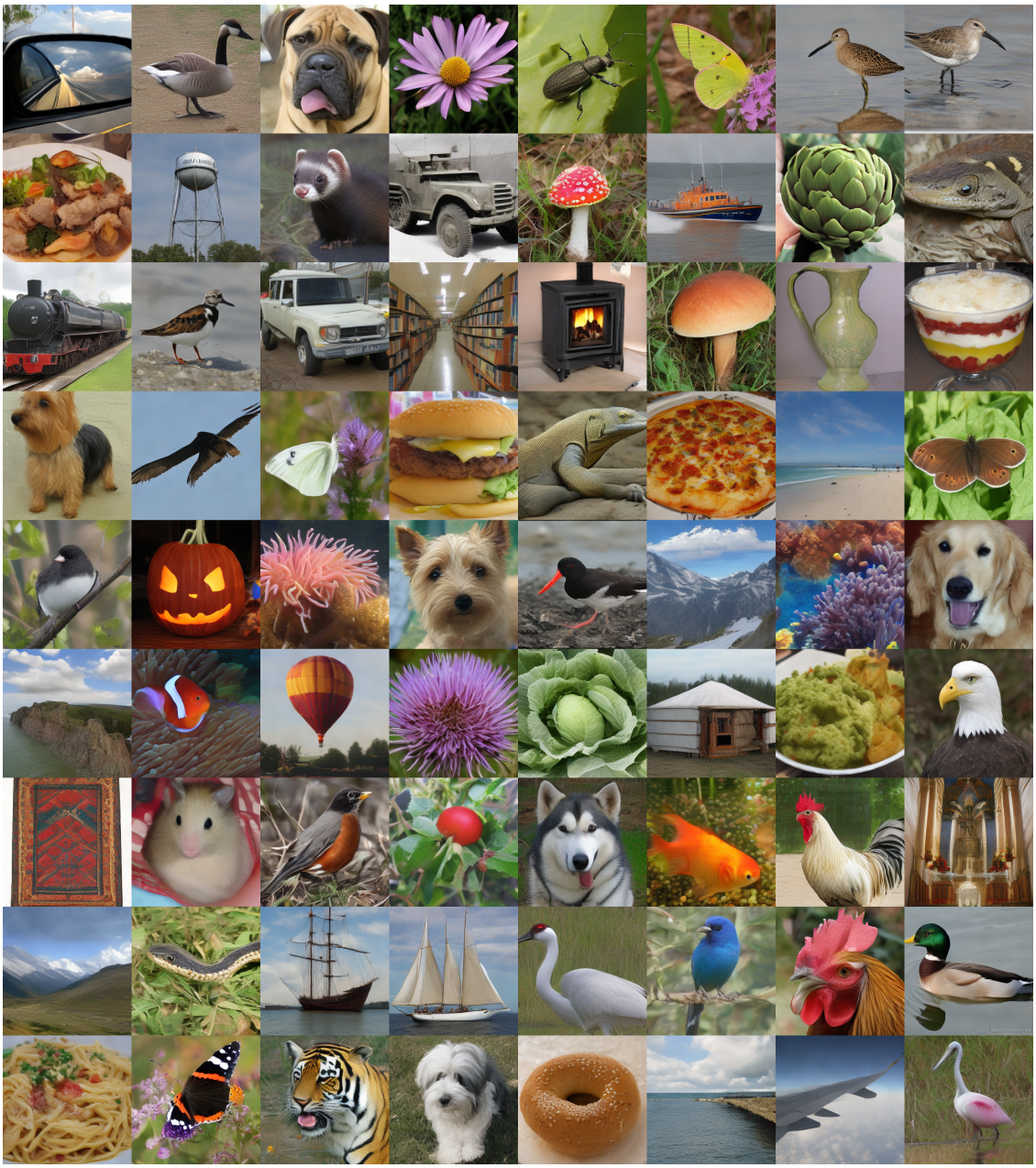


Figure 17: Two-step samples from the iSD-T on ImageNet 256×256.