DIFFERENCEFLOW: DYNAMIC IMPLICIT DIFFERENCES FOR ONE-STEP GENERATIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT



Figure 1: **One-step generation on ImageNet-1000 256×256.** DifferenceFlow achieves significantly better generation quality than previous state-of-the-art one-step baselines.

MeanFlow has emerged as a powerful framework for training one-step generative models from scratch, without relying on pretrained teachers or multi-stage distillation. However, its derivative-based training target, which involves temporal derivatives evaluated through Jacobian-vector products, can be difficult to optimize early in training. Recent approaches such as AlphaFlow alleviate this difficulty through target interpolation, but their interpolation weight is heuristic rather than derived from the MeanFlow identity, which biases the target toward the instantaneous velocity. To address these limitations, we propose **DifferenceFlow**, a discrete formulation of MeanFlow that replaces the temporal derivative with a finite difference between model predictions along the same trajectory. Derived directly from the discretized MeanFlow identity, the DifferenceFlow target removes this bias and is second-order accurate in the interval length, while shrinking the interval during training yields a coarse-to-fine curriculum toward the original MeanFlow objective. On class-conditional ImageNet-1000 at 256×256 resolution, our best model with fine-tuning outperforms state-of-the-art MeanFlow-based methods, achieving an FID of 2.29 with a single function evaluation.

1 INTRODUCTION

Diffusion and flow matching models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2020; Albergo & Vanden-Eijnden, 2022; Lipman et al., 2022) have become the dominant paradigms for generative modeling. However, their practical deployment is hindered by slow inference, as high-quality generation requires many denoising steps. This has motivated extensive research (Song & Dhariwal, 2024; Lu & Song, 2025; Frans et al., 2025) into few-step and one-step generative models that preserve sample quality while substantially reducing inference cost.

Recently, training few-step generators from scratch has emerged as a promising direction for accelerating sampling without relying on a pre-trained model or distillation. **MeanFlow** (MF) (Geng et al., 2026a) is a particularly effective approach that learns interval-averaged velocities through a differential identity, but its derivative-based target can be difficult to optimize. Building on MeanFlow, **AlphaFlow** (AF) (Zhang et al., 2026) alleviates this difficulty with a curriculum that mitigates conflicting gradients between trajectory flow matching and trajectory consistency. However, AlphaFlow's externally interpolated targets carry an additional bias proportional to the velocity gap

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

$v - u$ between instantaneous and average velocities. At the same time, its evolving training targets make training progress harder to monitor at early stage. This motivates a natural question: *can we retain the curriculum transition while constructing a target free of this velocity-gap bias and make the training progress faster?*

To address these limitations, we propose **DifferenceFlow** (DF), a dynamic curriculum derived directly from a finite-difference discretization of the MeanFlow identity. Evaluating the model at two trajectory points separated by the temporal interval $\Delta t = \alpha(t-r)$ yields a family of training targets in which α jointly controls the target composition and the temporal scale of the MeanFlow constraint, rather than externally interpolating between two targets as in AlphaFlow. We progressively reduce the temporal interval from 1 to 0 throughout training, retaining the curriculum transition from stronger explicit velocity supervision to the local MeanFlow target while maintaining consistency with the underlying MeanFlow identity. Crucially, because the DifferenceFlow target is derived directly from the finite-difference equation, it exactly satisfies the discretized MeanFlow identity and removes the additional $\alpha^2(v-u)$ bias carried by AlphaFlow's explicit interpolation: at every stage of the curriculum, its target is second-order accurate in the interval length, whereas AlphaFlow's is only first-order. This principled curriculum thus preserves the optimization benefits of gradual target transition while removing this bias.

Empirically, our experimental results demonstrate that the proposed dynamic curriculum consistently improves one-step generation across model scales and training configurations. On class-conditional ImageNet-1000 256×256 (Deng et al., 2009), our models trained from scratch outperform both MeanFlow and AlphaFlow under the 1-NFE setting. Moreover, the gains extend beyond FID and remain consistent across evaluation metrics computed using a diverse set of feature extractors, indicating that the improvements are not tied to a particular feature representation. These results further narrow the performance gap between one-step and multi-step generative models, highlighting the potential of one-step generation as a competitive stand-alone framework.

2 Related Work

Multi-step diffusion and flow models. Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) have become a dominant paradigm for generative modeling and can be formulated through stochastic differential equations (Song et al., 2020). Subsequent works have improved their generation quality, efficiency, and scalability (Dhariwal & Nichol, 2021; Karras et al., 2022; Rombach et al., 2022; Peebles & Xie, 2023). In parallel, flow matching and related formulations (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022a; Albergo et al., 2025; Esser et al., 2024; Ma et al., 2024) learn continuous velocity fields that transport noise to data through ordinary differential equation (ODE)-based dynamics, and have scaled successfully to image generation. Despite their performance, both paradigms typically require iterative integration with multiple model evaluations at inference, leading to substantial sampling cost. DifferenceFlow is motivated by this limitation and targets high-quality generation with a single function evaluation.

Few-step diffusion and flow models. To reduce the inference cost of multi-step models, few-step approaches mainly follow distillation- and consistency-based paradigms. Distillation methods compress a pre-trained multi-step model into a few-step generator (Salimans & Ho, 2022; Geng et al., 2023; Sauer et al., 2024; Luo et al., 2023), while Consistency Models enforce consistent predictions across time points or intervals (Song et al., 2023; Song & Dhariwal, 2024; Lu & Song, 2025; Geng et al., 2025; Frans et al., 2025; Kim et al., 2024; Wang et al., 2024; Yang et al., 2024; Peng et al., 2026). Recently, MeanFlow and its extensions (Geng et al., 2026a; Zhang et al., 2026; Geng et al., 2026b) learn interval-averaged velocities, enabling few- or one-step generation without multi-stage distillation. DifferenceFlow builds on this average-velocity formulation, but derives its curriculum directly from a finite-difference discretization of the MeanFlow identity. Unlike AlphaFlow's heuristic interpolation weight, the resulting weight is determined by the MeanFlow identity itself.

3 Preliminaries

In this section, we introduce the background of flow matching, MeanFlow and AlphaFlow.

3.1 FLOW MATCHING

Flow matching (Lipman et al., 2022) learns a velocity field that transports a prior to the data distribution. Given $x \sim p_0$ and $e \sim p_1 = \mathcal{N}(0, I)$, it constructs the linear path $x_t = (1-t)x + te$, $t \in [0,1]$, with conditional velocity $v_t = e - x$, and minimizes $\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{x,e,t}\|v_\theta(x_t,t) - v_t\|_2^2$, whose minimizer is the marginal velocity $v_\theta^*(x_t,t) = \mathbb{E}[v_t \mid x_t, t]$. Sampling integrates this field from $t=1$ to $t=0$ with a multi-step ODE solver, which makes generation slow.

3.2 MEANFLOW

MeanFlow (Geng et al., 2026a) instead models the average velocity $u(x_t,r,t) = \frac{1}{t-r}\int_r^t v(x_\tau,\tau)\,\mathrm{d}\tau$ over $[r,t]$, so that the entire trajectory can be traversed with a single function evaluation. Differentiating along the trajectory gives the MeanFlow identity $u = v - (t-r)\frac{\mathrm{d}u}{\mathrm{d}t}$, where the total derivative is evaluated by the Jacobian-vector product $\mathrm{JVP}(u_\theta;(x_t,r,t);(v_t,0,1))$, leading to

$$\mathcal{L}_{\mathrm{MF}}(\theta) = \mathbb{E}_{x,e,r,t}\left[\left\|u_\theta(x_t,r,t) - \mathrm{sg}\left(v_t - (t-r)\,\mathrm{JVP}\left(u_\theta;(x_t,r,t);(v_t,0,1)\right)\right)\right\|_2^2\right],$$

where sg denotes stop-gradient and $0 \le r \le t \le 1$; at $r=t$, $u=v$ and the derivative term vanishes. However, this derivative-based target is difficult to optimize, especially early in training, and induces conflicting signals between trajectory flow matching and trajectory consistency.

3.3 ALPHAFLOW

AlphaFlow (Zhang et al., 2026) mitigates this conflict with a curriculum. It introduces an intermediate time $s = \alpha r + (1-\alpha)t$ and a shifted state $x_s = x_t - (t-s)\tilde{v}_{s,t}$, where $\tilde{v}_{s,t}$ is a shift velocity, and minimizes

$$\mathcal{L}_{\mathrm{AF}}(\theta) = \mathbb{E}_{x,e,r,t}\left[\alpha^{-1}\left\|u_\theta(x_t,r,t) - (\alpha\tilde{v}_{s,t} + (1-\alpha)u_{\theta^-}(x_s,r,s))\right\|_2^2\right],$$

where θ^- is a stop-gradient copy of the model. As α decreases from 1 to 0, the target moves from trajectory flow matching to MeanFlow. This interpolation, however, is externally imposed rather than derived from the MeanFlow identity: for finite α, its target does not satisfy the backward-difference discretization of the identity and carries an additional bias term. This motivates a curriculum that preserves the gradual transition while remaining consistent with the MeanFlow identity.

4 METHODOLOGY

Our main objective is to introduce a mathematically principled and interpretable framework for designing an effective curriculum for training one-step generation models. Based on the average velocity formulation proposed by MeanFlow, we leverage finite-difference approximations to progressively approach the original MeanFlow objective.

4.1 FINITE DIFFERENCE

Finite difference provides a standard way to express derivatives using function values at neighboring points. For a differentiable function $f(x)$, its derivative can be written through the backward difference as

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0}\frac{f(x) - f(x-\Delta x)}{\Delta x},$$

where Δx denotes a finite change in the input variable. As Δx approaches zero, the difference quotient converges to the continuous derivative.

Following the same principle, we apply finite difference to the temporal derivative in the MeanFlow target. The average velocity $u(x_t,r,t)$ depends on t through both the temporal variable itself and the corresponding trajectory state x_t. Therefore, when the time point moves from $t-\Delta t$ to t, the trajectory state changes accordingly from $x_{t-\Delta t}$ to x_t. The temporal derivative of the average velocity along the trajectory can then be expressed as

$$\frac{du(x_t,r,t)}{dt} = \lim_{\Delta t \to 0}\frac{u(x_t,r,t) - u(x_{t-\Delta t},r,t-\Delta t)}{\Delta t}.$$

This expression describes the temporal change of the average velocity through two neighboring points on the same trajectory. As Δt approaches zero, the time point $t - \Delta t$ approaches t, and the difference between the corresponding average velocities recovers the continuous temporal derivative in MeanFlow. Substituting this expression into the original MeanFlow identity gives

$$u_{\text{target}}(x_t, r, t) = v(x_t, t) - \lim_{\Delta t \to 0} \left[(t - r) \frac{u(x_t, r, t) - u(x_{t-\Delta t}, r, t - \Delta t)}{\Delta t} \right]. \tag{1}$$

Since the interval length $t - r$ varies with the sampled time pair (r, t), we further parameterize the temporal displacement relative to the entire interval. Specifically, we let $\Delta t = \alpha(t - r)$, where $\alpha \in (0, 1]$ represents the relative temporal displacement within $[r, t]$. We then define the intermediate time point as $s = t - \Delta t$, which can equivalently be written as $s = \alpha r + (1 - \alpha)t$. Under this definition, for a fixed pair (r, t), $\Delta t \to 0$ corresponds to $\alpha \to 0$, and consequently $s \to t$. Substituting $\Delta t = \alpha(t - r)$ and $s = \alpha r + (1 - \alpha)t$ into Eq. (1), the interval length $t - r$ is canceled, yielding

$$u_{\text{target}}(x_t, r, t) = \lim_{\alpha \to 0} \left[\frac{\alpha}{1 + \alpha} v(x_t, t) + \frac{1}{1 + \alpha} u(x_s, r, s) \right],$$

where $v(x_t, t) = e - x$. As $\alpha \to 0$, the intermediate time point s approaches t, while the above expression recovers the original continuous MeanFlow relation.

4.2 Dynamic Curriculum For Alpha

The parameter α controls the relative contribution of the instantaneous velocity $v(x_t, t) = e - x$ and the predicted average velocity $u(x_s, r, s)$ in the training target. A smaller α places more emphasis on the predicted average velocity rather than the instantaneous velocity while reducing the contribution of the explicit instantaneous velocity. Since the prediction of $u(x_s, r, s)$ may still be inaccurate at the early stage of training, relying too heavily on it can make the optimization more difficult. In contrast, a relatively large α preserves more supervision from the directly available instantaneous velocity and reduces the dependence on the current model prediction.

Based on this observation, we treat α as a curriculum variable that changes dynamically during training. Instead of restricting α to the limiting case $\alpha \to 0$ throughout the entire training process, we relax this constraint and consider a finite $\alpha \in (0, 1]$. For a given α, we define the corresponding training target and objective

$$u^{\alpha}_{\text{target}}(x_t, r, t) = \frac{\alpha}{1 + \alpha} v(x_t, t) + \frac{1}{1 + \alpha} u(x_s, r, s), \tag{2}$$

$$\mathcal{L}_{\text{DF}}(\theta) = \mathbb{E}_{x,e,r,t} \left[\frac{(1 + \alpha)}{\alpha} \cdot \text{sg}(\omega) \left\| u_\theta(x_t, r, t) - \left(\frac{\alpha}{1 + \alpha} v(x_t, t) + \frac{1}{1 + \alpha} u_{\theta^-}(x_s, r, s) \right) \right\|_2^2 \right], \tag{3}$$

where $s = \alpha r + (1 - \alpha)t$. The detailed derivation is provided in Appendix C and the detailed algorithm is presented in Algorithm 1. As α decreases, the coefficient $\frac{\alpha}{1+\alpha}$ of the instantaneous velocity decreases, while the coefficient $\frac{1}{1+\alpha}$ of the average velocity increases. At the same time, the intermediate time point s moves continuously toward t. The prefactor $\frac{(1+\alpha)}{\alpha}$ preserves the gradient of the original stop-gradient finite-difference objective.

Therefore, varying α continuously changes both the composition of the training target and the temporal interval associated with $u(x_s, r, s)$. We use this property to construct a curriculum over the training process, where α is gradually adjusted to shift the target from stronger explicit velocity supervision toward the limiting MeanFlow objective.

4.3 Target Bias Analysis

The DifferenceFlow target in Eq. (2) exactly solves the discretized MeanFlow equation at finite α. This alone does not imply a more accurate target, since the exact average velocity itself has an $O(\alpha)$ residual in that equation (Section C.2). We therefore compare the interpolation targets themselves, rather than discrete residuals, network predictions, or training gradients, directly against the exact average velocity.

Fix $r < t$, let $L = t - r$, and let $s = t - \alpha L$ with $\alpha \in (0, 1]$. In this section, v denotes the marginal velocity at x_t, and all quantities are taken along the marginal flow trajectory through x_t: $w(\tau) = v(x_\tau, \tau)$ is the instantaneous velocity and $U(\tau) = u(x_\tau, r, \tau)$ is the average velocity over $[r, \tau]$, with $u_t = U(t)$, $u_s = U(s)$, and $v = w(t)$. We assume that w is twice continuously differentiable on $[0, t]$; then U is twice continuously differentiable on $[r, t]$ (Section C.2), and the MeanFlow identity reads $v - u_t = LU'(t)$, where U' denotes the total derivative along the trajectory.

Consider the general interpolation target

$$T_\beta = \beta v + (1 - \beta)u_s, \qquad \beta \in [0, 1]. \tag{4}$$

A first-order Taylor expansion, combined with the MeanFlow identity, gives

$$u_s = u_t - \alpha L U'(t) + \rho_\alpha = u_t - \alpha(v - u_t) + \rho_\alpha, \tag{5}$$

where the remainder satisfies

$$\|\rho_\alpha\| \le \frac{\alpha^2 L^2}{2} \sup_{\tau \in [s,t]} \|U''(\tau)\|. \tag{6}$$

Substituting Eq. (5) into Eq. (4) yields the exact error decomposition

$$T_\beta - u_t = \big[\beta(1 + \alpha) - \alpha\big](v - u_t) + (1 - \beta)\rho_\alpha. \tag{7}$$

If U is affine on $[s, t]$, then $\rho_\alpha = 0$, and for $v \neq u_t$ the target is exact if and only if $\beta = \alpha/(1 + \alpha)$, which is precisely the DifferenceFlow weight. In general,

$$T_{\mathrm{DF}} - u_t = \frac{\rho_\alpha}{1 + \alpha}, \tag{8}$$

whereas AlphaFlow, which uses $\beta = \alpha$ with the same v, u_s, and s, gives

$$T_{\mathrm{AF}} - u_t = \alpha^2(v - u_t) + (1 - \alpha)\rho_\alpha. \tag{9}$$

At a fixed stage of the curriculum, i.e., for a fixed α and a shrinking interval $L \to 0$, we have $v - u_t = LU'(t) = \frac{L}{2} w'(t) + O(L^2)$ while $\rho_\alpha = O(\alpha^2 L^2)$ (Section C.2), so that

$$\|T_{\mathrm{DF}} - u_t\| \le \frac{\alpha^2 L^2}{6(1 + \alpha)} \sup_{\tau \in [r,t]} \|w''(\tau)\|, \qquad T_{\mathrm{AF}} - u_t = \frac{\alpha^2 L}{2}\, w'(t) + O(\alpha^2 L^2). \tag{10}$$

Thus, at every curriculum stage, the DifferenceFlow target is second-order accurate in the interval length, whereas the AlphaFlow target is first-order whenever $w'(t) \neq 0$; for $\alpha = 1$, they reduce to the trapezoidal and right-endpoint rules for $u_t = \frac{1}{L} \int_r^t w(\tau)\, d\tau$. In the complementary limit $\alpha \to 0$ with L fixed, both errors are $O(\alpha^2)$ (Section C.2). These results also hold for the stop-gradient regression target $\mathbb{E}[T_\beta \mid x_t]$ under the conditional velocity and path of Algorithm 1, with v and ρ_α replaced by their conditional expectations.

4.4 Design Decisions

DifferenceFlow with guidance. We apply a similar classifier-free guidance (CFG) training strategy as MeanFlow and AlphaFlow (Geng et al., 2026a; Zhang et al., 2026), by setting $v(x_t, t)$ in Eq. (3) with

$$v(x_t, t) = w \cdot v(x_t, t|x) + \kappa \cdot u_{\theta^-}(x_t, t, t|c) + (1 - w - \kappa) \cdot u_{\theta^-}(x_t, t, t|\varnothing),$$

where w and κ denote the guidance scales, and $u_{\theta^-}(\cdot \mid c)$ and $u_{\theta^-}(\cdot \mid \varnothing)$ represent the class-conditional prediction for class c and the class-unconditional prediction, respectively. Further details on the settings of w and κ are provided in Appendix A.

Loss metrics. To improve training stability, we apply adaptive weighting to the loss function. Let Δ denote the prediction residual, for which the standard ℓ_2 loss is $\mathcal{L} = \|\Delta\|_2^2$. We define the adaptive weight as $\omega = \frac{\alpha}{1+\alpha}/(\|\Delta\|_2^2 + c)^p$, where $\frac{\alpha}{1+\alpha}$ accounts for the α-dependent interpolation scale and $(\|\Delta\|_2^2 + c)^{-p}$ adaptively rescales samples according to their residual magnitude. The final objective is $\mathrm{sg}(\omega)\frac{1+\alpha}{\alpha}\|\Delta\|_2^2$, where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. This design jointly incorporates the interpolation scale and residual-dependent normalization, reducing the influence of samples with disproportionately large residuals and improving optimization stability. We further study the effect of the loss metric and the hyperparameter p in Section 5.2.

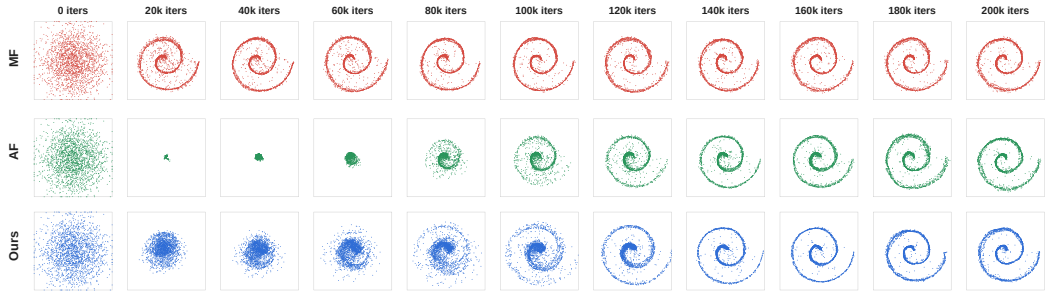


Figure 2: **Toy experiment on the Swiss-roll distribution.** We visualize the training dynamics of three different methods over 200K iterations, comparing how their generated distributions progressively approach the target Swiss-roll distribution throughout training. Red represents MeanFlow, green represents AlphaFlow, and blue represents DifferenceFlow.

5 Experiments

5.1 Toy Experiment

In this toy experiment, we first evaluate DifferenceFlow on the synthetic **Swiss-roll** dataset to provide an intuitive illustration of its effectiveness in capturing complex, nonlinear data distributions. We compare DifferenceFlow with **AlphaFlow** and **Meanflow** under the same experimental setting and use the **maximum mean discrepancy** (MMD) and **Gaussian Fréchet Distance** (Gaussian-FD) as the quantitative metric. A lower MMD and Gaussian-FD indicates better agreement between the two distributions, and the detailed computation of these metrics is provided in Appendix B.1.

As shown in Fig. 2 and Table 1, DifferenceFlow accurately recovers the Swiss-roll distribution. It achieves an MMD of 0.0113 and a Gaussian-FD of 0.0063, outperforming all competing methods on both metrics. Compared with the best-performing baseline, these results correspond to relative reductions of 35.4% in MMD compared to the MeanFlow. Visually, the samples generated by DifferenceFlow follow the Swiss-roll manifold more faithfully, providing more uniform coverage while producing fewer off-manifold samples. Together, these results suggest that DifferenceFlow effectively preserves the global geometry of the target distribution while reducing spurious probability mass outside its underlying manifold.

Table 1: **Quantitative evaluation on the Swiss-roll experiment.**

Method	MMD ↓	Gaussian-FD ↓
MF	0.0175	0.0064
AF	0.0487	0.0065
Ours	**0.0113**	**0.0063**

5.2 Ablation Study

In the ablation study, we investigate the sensitivity of DifferenceFlow to flow matching ratio, loss metric, time sampler and alpha schedule. We adopt DiT-B/2 developed in Peebles & Xie (2023) as the backbone and train our model without CFG. All models are trained from scratch for 80 epochs with a batch size of 256 and we evaluate the generative performance on FID, FDr6 and IS.

Ratio of flow matching. We investigate the impact of the flow matching ratio in Table 2a. The results reveal a trade-off between instantaneous velocity learning and trajectory-level consistency. DifferenceFlow achieves the best 1-NFE performance with a media flow matching ratio, demonstrating an appropriate flow matching ratio can improve generation performance.

Loss metric. Song & Dhariwal (2024) reports that the loss formulation plays an important role in optimizing the network. We compare different loss metrics by varying the weighting coefficient p in our adaptive loss weighting strategy. As shown in Table 2b, the standard squared L2 loss ($p = 0$) leads to inferior one-step generation performance, while adaptive weighting with $p = 0.5$ or $p = 1.0$ consistently improves the results. The best performance is achieved with $p = 1.0$, demonstrating that adaptive loss weighting provides a more effective optimization objective for one-step generation.

Table 2: **Ablation study on 1-NFE ImageNet 256 × 256 generation for DiT-B/2 w/o CFG.** FD and IS are computed over 50K generated samples under the 1-NFE sampling setting.

(a) **Ratio of flow matching $r \neq t$.** The 0% entry reduces to the standard Flow Matching baseline.

% of $r \neq t$	FID ↓	FDr-6 ↓	IS ↑
0% (= FM)	303.80	237.90	1.96
25%	43.63	53.48	34.35
50%	**41.12**	**52.78**	**37.24**
100%	59.50	64.44	21.99

(b) **Loss metric.** p=0 is squared L2 loss. p=0.5 is Pseudo-Huber loss.

p	FID ↓	FDr-6 ↓	IS ↑
0.0	51.54	59.36	28.06
0.5	46.20	55.33	31.29
1.0	**41.12**	**52.78**	**37.24**
1.5	46.36	56.01	32.99

(c) **Time sampler.** t and r are sampled from the specific sampler.

t, r sampler	FID ↓	FDr-6 ↓	IS ↑
uniform$(0, 1)$	53.41	53.26	26.88
logitnorm$(-0.2, 1.0)$	43.36	53.62	35.31
logitnorm$(-0.2, 1.2)$	45.31	55.56	32.86
logitnorm$(-0.4, 1.0)$	**41.12**	**52.78**	**37.24**

(d) **Alpha schedule.** Comparison of different scheduling strategies for α.

α	FID ↓	FDr-6 ↓	IS ↑
0.01	126.38	104.66	15.06
Sigmoid	**41.12**	**52.78**	**37.24**
Log	47.73	59.15	35.82
Linear	59.28	69.11	31.35

Time sampler. Prior works (Esser et al., 2024; Geng et al., 2026a) have shown that the sampling distributions of t and r strongly affect generation quality. As shown in Table 2c, logit-normal sampling generally outperforms uniform sampling. Time pairs (r, t) are sampled independently and then post-processed by swapping the two values when necessary to ensure $t > r$, while restricting the proportion of pairs with $r \neq t$ to a predefined ratio.

Alpha schedule. It is reported in Zhang et al. (2026) that different alpha schedules influence the generation performance of 1-NFE generation. We use 4 different alpha schedules, including constant (0.01), sigmoid, log and linear, and their corresponding algorithms are shown in Algorithm 2. As demonstrated in Table 2d, different schedules lead to noticeable differences in generation quality, demonstrating that the evolution of α during training plays an important role in controlling the training objective.

5.3 Comparison with Baselines

In this section, we evaluate DifferenceFlow on **ImageNet-1000 (256×256)** (Deng et al., 2009). We mainly compare the generation quality of DifferenceFlow with existing methods including **MeanFlow** (Geng et al., 2026a), **AlphaFlow** (Zhang et al., 2026), **Shortcut**(Frans et al., 2025), **GANs** (Brock et al., 2018; Kang et al., 2023; Sauer et al., 2022), **Autoregressive models** (Esser et al., 2021; Chang et al., 2022; Tian et al., 2024; Li et al., 2024) and **diffusion models** (Dhariwal & Nichol, 2021; Rombach et al., 2022; Hoogeboom et al., 2023; Peebles & Xie, 2023; Ma et al., 2024; Yu et al., 2024). The generated images are evaluated by using the **Fréchet distance** (FD), **Inception Score** (IS) and **Representation Fréchet Distance over 6 representations** (FDr6), computed in multiple feature spaces defined by different backbones, such as **Inception** (Szegedy et al., 2016), **ConvNeXt** (Liu et al., 2022b), **MAE** (He et al., 2022), **DINOv2** (Oquab et al., 2023), and **SigLIP** (Zhai et al., 2023). For each feature space, we evaluate the performance for only 1-NFE. Detailed computation procedures are provided in Appendix D.

ImageNet Comparison We train our models in the latent space of a pre-trained encoder (Rombach et al., 2022), which encodes each image into a $32 \times 32 \times 4$ latent representation that serves as the input to the model. More details on the experiments settings are provided in Appendix A.

As shown in Table 3, our model DF-B/2 trained from scratch achieves a competitive performance compared with existing methods. Specifically, it obtains 5.41 on FID and 210.60 on IS, outperforming the strongest baseline, AlphaFlow. Similar improvements can also be observed in terms of most other metrics such as CLIP FD and SigLIP FD, demonstrating the effectiveness of Difference-

Table 3: **Class-conditional generation on ImageNet.** †: Results for iCT-XL/2 are reported by Zhou et al. (2025). The best results are highlighted in **bold**.

Method	Source	Params	NFE	FID↓
1-NFE diffusion/flow from scratch				
iCT-XL/2†	(Song & Dhariwal, 2024)	675M	1	34.24
Shortcut-XL/2	(Frans et al., 2025)	675M	1	10.60
IMM-XL/2	(Zhou et al., 2025)	675M	1	8.05
TiM-XL/2	(Wang et al., 2026)	664M	1	3.26
MeanFlow-B/2	Our Reproduction	131M	1	6.14
MeanFlow-XL/2	(Geng et al., 2026a)	676M	1	3.46
AlphaFlow-B/2	Our Reproduction	131M	1	5.60
AlphaFlow-XL/2	(Zhang et al., 2026)	676M	1	2.95
AlphaFlow-XL/2+	(Zhang et al., 2026)	676M	1	2.58
DifferenceFlow-B/2	Ours	131M	1	5.41
DifferenceFlow-B/2+	Ours	131M	1	**2.29**

Method	Source	Params	NFE	FID↓
GANs				
BigGAN	(Brock et al., 2018)	112M	1	6.95
GigaGAN	(Kang et al., 2023)	569M	1	3.45
StyleGAN-XL	(Sauer et al., 2022)	166M	1	2.30
autoregressive/masking				
AR w/ VQGAN	(Esser et al., 2021)	227M	1024	26.52
MaskGIT	(Chang et al., 2022)	227M	8	6.18
VAR-$d30$	(Tian et al., 2024)	2B	10×2	1.92
MAR-H	(Li et al., 2024)	943M	256×2	1.55
diffusion/flow				
LDM-4-G	(Rombach et al., 2022)	400M	250×2	3.60
SimDiff	(Hoogeboom et al., 2023)	2B	512×2	2.77
DiT-XL/2	(Peebles & Xie, 2023)	675M	250×2	2.27
SiT-XL/2	(Ma et al., 2024)	675M	250×2	2.06
SiT-XL/2+REPA	(Yu et al., 2024)	675M	250×2	1.42

Table 4: **ImageNet generation results under different model scales and classifier-free guidance settings.** The best results are highlighted in **bold**.

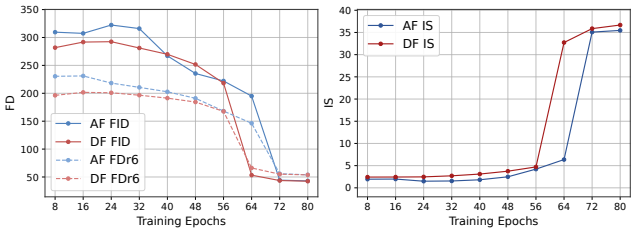
Metric	B/2 w/o CFG			B/2 w/ CFG			
	Ours	**AF**	**MF**	**Ours**	**AF**	**MF**	**Ours(B/2+)**
FDr-6 ↓	**52.78**	53.55	54.16	18.22	19.79	22.17	**15.00**
IS ↑	37.24	**39.61**	34.27	210.60	208.44	201.82	**251.44**
FID(JiT) FD ↓	**40.86**	42.12	43.34	5.42	5.49	6.06	**2.37**
FID(ADM) FD ↓	**41.12**	41.89	43.62	5.41	5.60	6.14	**2.29**
FID(ADM) FDr ↓	**24.47**	24.93	25.96	3.22	3.33	3.65	**1.36**
MAE cls FD ↓	**1.04**	1.15	**1.04**	0.55	0.58	0.64	**0.42**
MAE cls FDr ↓	26.19	28.81	**26.17**	13.94	14.59	16.24	**10.63**
MAE avg FD ↓	**2.80**	3.95	2.83	3.75	2.13	2.62	**1.20**
DINOv2 cls FD ↓	822.51	**805.06**	845.49	246.46	295.54	323.38	**207.55**
DINOv2 cls FDr ↓	57.96	**56.73**	59.58	17.36	20.82	22.78	**14.62**
DINOv2 avg FD ↓	200.13	**196.02**	205.06	73.28	88.05	96.93	**65.13**
CLIP cls FD ↓	**380.29**	385.79	382.02	163.08	189.42	214.04	**156.97**
CLIP cls FDr ↓	**67.91**	68.89	68.21	29.12	33.82	38.22	**28.03**
SigLIP cls FD ↓	**60.34**	61.57	62.01	21.80	24.75	26.68	**19.52**
SigLIP cls FDr ↓	**100.57**	102.63	103.35	36.33	41.25	44.47	**32.53**
SigLIP avg FD ↓	**120.01**	122.01	122.94	45.64	49.50	52.55	**39.66**

Flow. This improvement may be partly attributed to Section 4.3: at every finite α of the curriculum, DifferenceFlow removes the $\alpha^2(v-u)$ bias present in AlphaFlow.

To further compare DifferenceFlow with two closely related methods: Meanflow and AlphaFlow, Table 4 demonstrates more results evaluated under more metrics of these 3 methods. Under B/2 w/o CFG, DifferenceFlow achieves the best performance on 11 out of 16 evaluation metrics, demonstrating its strong and well-rounded performance across diverse evaluation dimensions. Under B/2 w/ CFG and w/o fine-tuning, our model ranks first on 15 out of 16 reported metrics, indicating improvements across multiple feature representations.

5.4 Fréchet Distance-based Fine-tuning

In this section, we adopt the fine-tuning strategy proposed in Yang et al. (2026) to the pre-trained model DF-B/2. This fine-tuning process enables the model to better capture the target data distribution while keeping its basic generative capabilities. Given a pretrained generator G_θ, a frozen VAE decoder D, and a frozen feature extractor ϕ, we sample noise $z_i \sim \mathcal{N}(0, I)$ and class condition $y_i \sim p(y)$. The generated images and their feature representations are given by $x_i = D(G_\theta(z_i, y_i))$,



(a) **Convergence comparison between DifferenceFlow and AlphaFlow on ImageNet.** Red represents DifferenceFlow and blue represents AlphaFlow.



(b) **Samples generated before and after fine-tuning.** The top row shows samples before fine-tuning, while the bottom row shows samples after fine-tuning.

Figure 3: **Convergence and Fine-tuning Experiments.** In Fig. 3a, for the left plot, curves closer to the lower-left corner indicate better performance, while for the right plot, curves closer to the upper-left are preferred.

$f_i = \phi(x_i)$. Let (μ_r, Σ_r) denote the mean and covariance of the real-data distribution in the feature space. The corresponding statistics of the generated samples are computed as $\mu_g = \frac{1}{N}\sum_{i=1}^{N} f_i$, $\Sigma_g = \frac{1}{N-1}\sum_{i=1}^{N}(f_i - \mu_g)(f_i - \mu_g)^\top$. The Fréchet distance between the generated and real feature distributions is defined as

$$\mathcal{L}_{\mathrm{FD}}(\theta) = \|\mu_g - \mu_r\|_2^2 + \mathrm{Tr}\left(\Sigma_g + \Sigma_r - 2\left(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2}\right)^{1/2}\right).$$

The mean term aligns the centers of the two distributions, while the covariance term matches their variation and correlation structures. The generator is fine-tuned by minimizing this distance, i.e., $\theta^* = \arg\min_\theta \mathcal{L}_{\mathrm{FD}}(\theta)$. During optimization, gradients are propagated through ϕ and D to G_θ, while only the generator parameters are updated.

Figure 3b presents a controlled comparison of images generated before and after fine-tuning, where each pair is produced from the same initial noise. Before fine-tuning, the images exhibit structural and semantic artifacts, including duplicated bird beaks, malformed buildings and incorrectly overlapping flowers. Fine-tuning substantially reduces these artifacts, resulting in more coherent structures. This demonstrates that our fine-tuning strategy improves generation quality.

5.5 Convergence Comparison

In this section, we compare the convergence rates of DifferenceFlow and AlphaFlow. We follow the same implementation details as in the ablation study and adopt a fair alpha schedule for both methods. The detailed results are provided in Appendix B.2. As presented in Fig. 3a, DifferenceFlow exhibits a sharp improvement in FID, FDr-6, and IS around epoch 64, whereas AlphaFlow shows a comparable transition around epoch 72. In particular, AlphaFlow exhibits a degradation in performance during the early stage of training, whereas this phenomenon is much less pronounced in DifferenceFlow. Moreover, the convergence rate comparison would be more obvious in the toy experiments shown in Fig. 2. A possible explanation for this behavior is the different weight assigned to the bootstrapped prediction $u_{\theta^-}(x_s, r, s)$. DifferenceFlow assigns it a coefficient of $1/(1+\alpha)$, which is no smaller than the coefficient $1-\alpha$ used by AlphaFlow. This gives greater relative weight to the bootstrapped prediction at the same α and may contribute to the observed training dynamics.

6 Conclusions

This paper proposes DifferenceFlow, a finite-difference framework for training one-step generative models that improves the optimization of the derivative-based MeanFlow objective. By discretizing the MeanFlow identity, we derive a principled curriculum in which the interval parameter α jointly controls target composition and temporal scale. The resulting DifferenceFlow target exactly satisfies the discretized identity and removes the velocity-gap bias of AlphaFlow's interpolation. Experiments across model scales and evaluation metrics show that DifferenceFlow consistently accelerates convergence and improves generation quality, providing a principled foundation for more efficient one-step generative modeling.

AI USE STATEMENT

AI-based tools were used during the preparation of this manuscript for limited purposes, including language editing, proofreading, and assisting with code implementation. All mathematical derivations, theoretical results, algorithmic designs, and scientific conclusions were developed by the authors. All AI-assisted code was manually reviewed, verified, and validated by the authors, and all experimental implementations and empirical analyses were independently checked by the authors.

REFERENCES

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

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

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. Ieee, 2009.

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

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

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

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems*, 36:41914–41931, 2023.

Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and Zico Kolter. Consistency models made easy. In *International Conference on Learning Representations*, volume 2025, pp. 75158–75185, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026b.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

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

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10124–10134. IEEE, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Dongjun Kim, Chieh-Hsin Lai, WeiHsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44493–44525, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022b.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, volume 2025, pp. 50611–50649, 2025.

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

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. Facm: Flow-anchored consistency models. In *International Conference on Learning Representations*, volume 2026, pp. 7520–7543, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

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

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

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

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

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

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

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*, pp. 29178–29189, 2026.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fr\'echet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026.

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. *arXiv preprint arXiv:2407.02398*, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. In *International Conference on Learning Representations*, volume 2026, pp. 144056–144094, 2026.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

A Training Details

Table 5: **Model architectures and training configurations.**

Configs	DiT-B/2-non-cfg	DiT-B/2	DiT-B/2+
Network Architectures			
Params (M)	131	131	131
Depth	12	12	12
Hidden dim	768	768	768
Heads	12	12	12
Patch size	2×2	2×2	2×2
Training Hyperparameters			
Training steps	400K	1.2M	–
Batch size for training	256	512	–
Fine-tuning steps	–	–	0.02M
Batch size for fine-tuning	–	–	128
Dropout	0.0	0.0	0.0
Optimizer		Adam (Kingma & Ba, 2014)	
LR schedule	constant	constant	cosine
LR	1×10^{-4}	1×10^{-4}	1×10^{-6}
Adam (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$	$(0.9, 0.95)$
Weight decay	0.0	0.0	0.0
EMA	0.999	0.999	0.999
Autoencoder used	sd-vae-ft-ema	sd-vae-ft-ema	sd-vae-ft-ema
α-Flow Hyperparameters			
Ratio of $r = t$	Table 2a	25%	–
(r, t) sampler	Table 2c	logitnorm(-0.4,1.0)	–
Adaptive weight	Table 2b	$\frac{\alpha}{1+\alpha}\left(\|\Delta\|_2^2 + c\right)^{-p}$	–
Schedule of α			
γ	Table 2d	25	–
k_s	Table 2d	0	–
k_e	Table 2d	1M	–
η	Table 2d	5×10^{-3}	–
CFG Training			
w	–	1.0	–
κ	–	1.0	–
CFG triggered if t is in	–	$[0.0, 1.0]$	–

B Additional Experiments

B.1 Toy Experiment Metrics

Maximum Mean Discrepancy. We quantify the distribution discrepancy between the real samples $X = \{x_i\}_{i=1}^n$ and the generated samples $Y = \{y_j\}_{j=1}^m$ using the squared Maximum Mean Discrepancy (MMD). Specifically, we adopt a multi-scale radial basis function (RBF) kernel:

$$k(a, b) = \sum_{\sigma \in \mathcal{S}} \exp\left(-\frac{\|a - b\|_2^2}{2\sigma^2}\right), \qquad \mathcal{S} = \{0.05, 0.1, 0.2, 0.5, 1.0\}.$$

The unbiased empirical estimator of the squared MMD is defined as

$$\widehat{\mathrm{MMD}}_u^2(X,Y) = \frac{1}{n(n-1)}\sum_{i\neq i'} k(x_i, x_{i'}) + \frac{1}{m(m-1)}\sum_{j\neq j'} k(y_j, y_{j'}) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i, y_j). \tag{11}$$

Gaussian Fréchet Distance. We further evaluate the distribution discrepancy by comparing the first- and second-order statistics of the two sample distributions using Gaussian Fréchet Distance (GFD). Let μ_X and Σ_X denote the empirical mean and covariance of the real samples, while μ_Y and Σ_Y represent those of the generated samples. The Gaussian Fréchet Distance is defined as

$$D_{\mathrm{GFD}}(X,Y) = \|\mu_X - \mu_Y\|_2^2 + \mathrm{Tr}\left(\Sigma_X + \Sigma_Y - 2\left(\Sigma_X^{1/2}\Sigma_Y\Sigma_X^{1/2}\right)^{1/2}\right).$$

B.2 Convergence between DifferenceFlow and AlphaFlow

Table 6: Evaluation of AlphaFlow checkpoints using 50K generated samples. ↑ indicates higher is better, while ↓ indicates lower is better.

Feature	Metric	AF-1	AF-2	AF-3	AF-4	AF-5	AF-6	AF-7	AF-8	AF-9	AF-10
General	FID (JiT) ↓	310.47	308.35	323.80	317.50	268.73	236.87	222.99	195.59	43.96	42.62
	FID (ADM) ↓	309.51	307.35	322.32	315.97	267.06	235.46	222.10	194.96	44.34	43.01
	IS ↑	1.94	1.97	1.49	1.54	1.81	2.48	4.21	6.36	35.08	35.46
	FDr-6 ↓	230.43	231.10	218.31	210.58	202.71	190.97	168.14	146.26	55.84	53.91
	FDr (FID/ADM) ↓	184.23	182.95	191.86	188.07	158.96	140.15	132.20	116.05	26.39	25.60
MAE	FD (CLS) ↓	10.15	10.14	9.05	8.74	8.42	7.52	5.80	4.47	1.10	1.07
	FDr (CLS) ↓	253.90	253.74	226.34	218.58	210.52	188.16	145.04	111.85	27.65	26.91
	FD (AVG) ↓	176.00	177.22	81.65	55.09	40.42	34.38	21.68	17.56	3.00	3.06
DINOv2	FD (CLS) ↓	2739.99	2745.12	2570.41	2543.82	2563.29	2547.66	2296.01	1956.38	862.74	836.60
	FDr (CLS) ↓	193.09	193.45	181.14	179.26	180.64	179.53	161.80	137.87	60.79	58.95
	FD (AVG) ↓	786.18	787.88	833.49	869.41	870.05	792.83	633.66	527.35	215.88	208.98
CLIP	FD (CLS) ↓	1016.88	1016.66	996.79	1005.38	1080.30	1039.34	918.79	839.91	395.06	378.57
	FDr (CLS) ↓	181.58	181.54	177.99	179.53	192.91	185.59	164.07	149.98	70.54	67.60
	FD (AVG) ↓	95.26	94.80	90.84	91.61	91.97	91.78	90.17	87.36	48.46	46.81
SigLIP	FD (CLS) ↓	248.19	251.57	227.53	214.85	200.70	187.99	167.70	151.90	64.63	62.22
	FDr (CLS) ↓	413.65	419.28	379.21	358.08	334.50	313.33	279.50	253.17	107.72	103.70
	FD (AVG) ↓	438.42	439.72	412.83	386.33	344.23	321.02	284.12	255.16	126.48	122.09
ConvNeXt	FD ↓	8879.19	8851.49	8719.96	7960.36	7891.63	7909.69	7178.25	6177.89	2385.34	2315.17
	FDr ↓	156.13	155.64	153.33	139.97	138.76	139.08	126.22	108.63	41.94	40.70

Table 7: Evaluation of DifferenceFlow checkpoints using 50K generated samples. ↑ indicates higher is better, while ↓ indicates lower is better.

Feature	Metric	Ours-1	Ours-2	Ours-3	Ours-4	Ours-5	Ours-6	Ours-7	Ours-8	Ours-9	Ours-10
General	FID (JiT) ↓	282.22	292.14	292.75	281.29	269.85	251.54	217.98	53.02	43.33	42.14
	FID (ADM) ↓	281.77	291.76	292.46	281.15	269.86	251.64	218.17	53.28	43.62	42.45
	IS ↑	2.42	2.43	2.46	2.70	3.10	3.74	4.70	32.73	35.86	36.67
	FDr-6 ↓	196.28	201.65	200.92	196.57	191.29	184.30	167.53	66.17	55.04	53.78
	FDr (FID/ADM) ↓	167.72	173.66	174.08	167.35	160.63	149.79	129.86	31.72	25.96	25.27
MAE	FD (CLS) ↓	6.97	7.23	7.06	6.69	6.21	5.83	4.95	1.34	1.08	1.05
	FDr (CLS) ↓	174.29	180.91	176.65	167.27	155.27	145.75	123.89	33.60	27.09	26.45
	FD (AVG) ↓	27.92	33.01	32.31	29.46	28.06	26.66	23.63	4.36	3.14	2.99
DINOv2	FD (CLS) ↓	2658.93	2684.54	2680.19	2639.28	2563.79	2466.27	2260.06	994.65	854.45	838.11
	FDr (CLS) ↓	187.38	189.18	188.87	185.99	180.67	173.80	159.27	70.09	60.21	59.06
	FD (AVG) ↓	783.23	790.59	779.01	747.63	709.75	666.74	595.98	254.55	211.40	205.21
CLIP	FD (CLS) ↓	967.23	969.48	967.56	957.88	947.01	930.66	901.35	476.25	392.85	385.38
	FDr (CLS) ↓	172.72	173.12	172.77	171.05	169.10	166.18	160.95	85.04	70.15	68.81
	FD (AVG) ↓	99.05	99.73	100.48	100.43	100.11	99.10	96.85	56.89	47.87	46.74
SigLIP	FD (CLS) ↓	196.82	204.70	205.12	203.80	203.27	199.69	184.52	77.08	63.35	61.71
	FDr (CLS) ↓	328.03	341.16	341.86	339.68	338.78	332.82	307.54	128.46	105.58	102.85
	FD (AVG) ↓	317.58	323.06	323.48	320.14	316.23	309.53	289.34	143.25	125.02	121.56
ConvNeXt	FD ↓	8393.10	8637.45	8602.62	8421.01	8147.95	7818.82	7034.30	2735.65	2347.39	2289.97
	FDr ↓	147.58	151.88	151.26	148.07	143.27	137.48	123.69	48.10	41.27	40.26

C PROOFS

C.1 DF-LOSS

Proof. The DifferenceFlow Loss is given by:

$$\begin{aligned}
\mathcal{L}_{DF}(\theta) &= \mathbb{E}_{x,e,t,r}\left[\left|u_\theta(x_t,r,t) - v(x_t,t) + (t-r)\frac{du_{\theta^-}(x_t,r,t)}{dt}\right|^2\right] \\
&= \mathbb{E}_{x,e,t,r}\left[\left|u_\theta(x_t,r,t) - v(x_t,t) + (t-r)\frac{u_{\theta^-}(x_t,r,t) - u_{\theta^-}(x_s,r,s)}{\Delta t}\right|^2\right] \\
&= \mathbb{E}_{x,e,t,r}\left[\left|u_\theta(x_t,r,t) - v(x_t,t) + (t-r)\frac{u_{\theta^-}(x_t,r,t) - u_{\theta^-}(x_s,r,s)}{\alpha(t-r)}\right|^2\right] \\
&= \mathbb{E}_{x,e,t,r}\left[\left|\left(1+\frac{1}{\alpha}\right)u_\theta(x_t,r,t) - v(x_t,t) - \frac{1}{\alpha}u_{\theta^-}(x_s,r,s)\right|^2\right] \\
&= \mathbb{E}_{x,e,t,r}\left[\left|\frac{1+\alpha}{\alpha}\left(u_\theta(x_t,r,t) - \frac{\alpha}{1+\alpha}v(x_t,t) - \frac{1}{1+\alpha}u_{\theta^-}(x_s,r,s)\right)\right|^2\right] \\
&= \mathbb{E}_{x,e,t,r}\left[\frac{(1+\alpha)^2}{\alpha^2}\left|u_\theta(x_t,r,t) - \frac{\alpha}{1+\alpha}v(x_t,t) - \frac{1}{1+\alpha}u_{\theta^-}(x_s,r,s)\right|^2\right]. \qquad (12)
\end{aligned}$$

Although the original stop-gradient residual equals $\frac{1+\alpha}{\alpha}\Delta$ in its forward value, its derivative with respect to θ is $\nabla_\theta u_\theta(x_t,r,t)$. Therefore, to preserve the gradient of the original stop-gradient squared-residual objective, we use $\frac{1+\alpha}{\alpha}\|\Delta\|_2^2$ rather than $\frac{(1+\alpha)^2}{\alpha^2}\|\Delta\|_2^2$. We then apply the additional detached adaptive weight $\mathrm{sg}(\omega)$ for training stability.

C.2 TARGET BIAS ANALYSIS

This section provides the derivations for Section 4.3. Throughout, $r < t$ and $\alpha \in (0,1]$ are fixed unless stated otherwise, $L = t - r$, and $s = t - \alpha L$. We follow the notation of Section 4.3. Until the paragraph on the conditional path, x_τ lies on the marginal flow trajectory through x_t, and $v(x_\tau,\tau)$ denotes the corresponding marginal velocity. For compactness, define $w(\tau) = v(x_\tau,\tau)$, $U(\tau) = u(x_\tau,r,\tau)$, $u_t = U(t)$, $u_s = U(s)$, and $v = v(x_t,t) = w(t)$. Assume that w is twice continuously differentiable on $[0,t]$.

Average velocity and MeanFlow identity. By definition, $(\tau - r)\,U(\tau) = \int_r^\tau w(\sigma)\,d\sigma$. Differentiating in τ gives $U(\tau) + (\tau - r)U'(\tau) = w(\tau)$, and at $\tau = t$ this is the MeanFlow identity $v - u_t = LU'(t)$. Substituting $\sigma = r + \theta(\tau - r)$ gives $U(\tau) = \int_0^1 w\big(r + \theta(\tau - r)\big)\,d\theta$, and differentiating under the integral,

$$U'(\tau) = \int_0^1 \theta\, w'\big(r + \theta(\tau - r)\big)\,d\theta, \qquad U''(\tau) = \int_0^1 \theta^2\, w''\big(r + \theta(\tau - r)\big)\,d\theta. \qquad (13)$$

Hence U is twice continuously differentiable on $[r,t]$ and $\sup_{\tau\in[r,t]}\|U''(\tau)\| \le \frac{1}{3}\sup_{\tau\in[r,t]}\|w''(\tau)\|$.

Taylor expansion and remainder bound. Taylor's theorem with integral remainder gives

$$U(s) = U(t) - \alpha L\,U'(t) + \rho_\alpha, \qquad \rho_\alpha = \int_s^t (\tau - s)\,U''(\tau)\,d\tau. \qquad (14)$$

Since $\int_s^t(\tau - s)\,d\tau = \alpha^2L^2/2$, this yields Eq. (6). Substituting $LU'(t) = v - u_t$ into Eq. (14) gives Eq. (5). Combining the remainder bound with Eq. (13),

$$\|\rho_\alpha\| \le \frac{\alpha^2L^2}{6}\sup_{\tau\in[r,t]}\|w''(\tau)\|, \qquad (15)$$

and since $\sup_{[r,t]}\|w''\| \le \sup_{[0,t]}\|w''\|$, this bound holds uniformly in both α and r.

Error decomposition. For $T_\beta = \beta v + (1-\beta)u_s$,

$$T_\beta - u_t = \beta(v-u_t) + (1-\beta)(u_s-u_t) = \beta(v-u_t) - (1-\beta)\alpha(v-u_t) + (1-\beta)\rho_\alpha,$$

which equals Eq. (7). For $\beta = \alpha/(1+\alpha)$, the coefficient of $v-u_t$ vanishes and $1-\beta = 1/(1+\alpha)$, giving Eq. (8). For $\beta = \alpha$, the coefficient equals α^2 and $1-\beta = 1-\alpha$, giving Eq. (9). If U is affine on $[s,t]$, then $U'' \equiv 0$ on $[s,t]$, so $\rho_\alpha = 0$ by Eq. (14), and $T_\beta - u_t = [\beta(1+\alpha)-\alpha](v-u_t)$. For $v \neq u_t$, this vanishes if and only if $\beta = \alpha/(1+\alpha)$.

Order in the interval length. The bound for DifferenceFlow in Eq. (10) follows from Eqs. (8) and (15). For AlphaFlow, by Eq. (13),

$$\begin{aligned}\Big\| LU'(t) - \tfrac{L}{2}w'(t) \Big\| &= L\Big\| \int_0^1 \theta\big(w'(r+\theta L) - w'(t)\big)\, d\theta \Big\| \\ &\le L^2 \sup_{\tau\in[r,t]} \|w''(\tau)\| \int_0^1 \theta(1-\theta)\, d\theta = \frac{L^2}{6} \sup_{\tau\in[r,t]} \|w''(\tau)\|,\end{aligned}$$

where we used $\|w'(r+\theta L) - w'(t)\| \le (1-\theta)L \sup_{[r,t]} \|w''\|$. Hence $v - u_t = LU'(t) = \frac{L}{2}w'(t) + O(L^2)$, and by Eqs. (9) and (15),

$$T_{\mathrm{AF}} - u_t = \frac{\alpha^2 L}{2}\, w'(t) + R_\alpha, \qquad \|R_\alpha\| \le \frac{(2-\alpha)\,\alpha^2 L^2}{6} \sup_{\tau\in[r,t]} \|w''(\tau)\|,$$

which is the second part of Eq. (10). For $\alpha = 1$, we have $s = r$ and $u(x_r, r, r) = w(r)$, so $T_{\mathrm{DF}} = \frac{1}{2}\big(w(t) + w(r)\big)$ and $T_{\mathrm{AF}} = w(t)$: the trapezoidal and right-endpoint rules for $u_t = \frac{1}{L}\int_r^t w(\tau)\, d\tau$.

Discrete residual versus target error. Replacing the derivative in the MeanFlow identity by the backward difference $(z-u_s)/(\alpha L)$ for a candidate target z gives the discretized equation $\mathcal{R}(z) = 0$ with

$$\mathcal{R}(z) = v - z - \frac{z-u_s}{\alpha}. \tag{16}$$

Its unique solution is $z = (\alpha v + u_s)/(1+\alpha) = T_{\mathrm{DF}}$. Since $\mathcal{R}$ is affine in z with slope $-(1+\alpha)/\alpha$, every z satisfies

$$z - u_t = \frac{\alpha}{1+\alpha}\big(\mathcal{R}(u_t) - \mathcal{R}(z)\big). \tag{17}$$

By Eq. (5), the exact average velocity has residual $\mathcal{R}(u_t) = \rho_\alpha/\alpha$, which is $O(\alpha)$ and nonzero in general. A vanishing discrete residual therefore characterizes the solution of Eq. (16), not the exact average velocity. For AlphaFlow, $\mathcal{R}(T_{\mathrm{AF}}) = -\alpha(v-u_s)$, and Eq. (17) recovers Eq. (9).

Conditional velocity and path. In this paragraph, v denotes the sampled training velocity used in Algorithm 1, and x_τ denotes the shifted state $x_\tau = x_t - (t-\tau)v$. Without guidance, $v = e - x$ and $x_\tau = (1-\tau)x + \tau e$. Let $U^c(\tau) = u(x_\tau, r, \tau)$ along this path, and let $\bar v = \mathbb{E}[v \mid x_t]$ denote the corresponding marginal velocity. Conditioning on the fixed times r, t is implicit throughout this paragraph; any fixed class condition is also implicit. Assume that u is twice continuously differentiable in (x,τ) with second derivatives bounded by a constant C independent of r, and that $\mathbb{E}[\|v\|^2 \mid x_t] < \infty$. By the chain rule, $(U^c)'(t) = \partial_t u + \nabla_x u\, v$, where $\partial_t u$ and $\nabla_x u$ are evaluated at (x_t, r, t) and are therefore deterministic given x_t, and $\|(U^c)''(\tau)\| \le C(1+\|v\|)^2$. The MeanFlow identity for the marginal field reads $\bar v - u_t = L(\partial_t u + \nabla_x u\, \bar v)$. Consequently, the pointwise residual $\delta = v - u_t - L(U^c)'(t)$ is affine in v and satisfies

$$\mathbb{E}[\delta \mid x_t] = \bar v - u_t - L(\partial_t u + \nabla_x u\, \bar v) = 0.$$

Applying Eq. (14) to U^c with remainder ρ_α^c and substituting $L(U^c)'(t) = v - u_t - \delta$ gives $u_s = u_t - \alpha(v-u_t) + \alpha\delta + \rho_\alpha^c$, and hence

$$T_\beta - u_t = \big[\beta(1+\alpha) - \alpha\big](v-u_t) + (1-\beta)\big(\alpha\delta + \rho_\alpha^c\big).$$

Taking conditional expectations yields

$$\mathbb{E}[T_\beta \mid x_t] - u_t = \big[\beta(1+\alpha) - \alpha\big](\bar v - u_t) + (1-\beta)\,\mathbb{E}[\rho_\alpha^c \mid x_t], \tag{18}$$

with $\|\mathbb{E}[\rho_\alpha^c \mid x_t]\| \le \frac{\alpha^2 L^2}{2} C\,\mathbb{E}\big[(1+\|v\|)^2 \mid x_t\big] = O(\alpha^2 L^2)$. Thus Eqs. (7) to (9) hold for $\mathbb{E}[T_\beta \mid x_t]$ with v replaced by $\bar{v}$ and ρ_α by $\mathbb{E}[\rho_\alpha^c \mid x_t]$. Since $\bar{v} - u_t$ is the same as on the marginal trajectory, the orders in Eq. (10) also carry over. The zero-mean term $(1-\beta)\alpha\delta$ affects only the variance of the target. Finally, for fixed α and a stop-gradient target, the minimizer of the unweighted squared-error objective $\mathbb{E}\|f(x_t,r,t) - T_\beta\|_2^2$ over measurable f is $\mathbb{E}[T_\beta \mid x_t, r, t]$. Thus, Eq. (18) characterizes this conditional-mean regression target. The same argument applies with $v = v_{\text{cfg}}$ from Section 4.4, provided that the guided velocity is used in both the target and the state shift, $\bar{v} = \mathbb{E}[v_{\text{cfg}} \mid x_t]$ denotes the corresponding guided marginal field, and u denotes its average-velocity field under the same regularity and moment assumptions.

Network predictions. If u_s is replaced by a prediction $\hat{u}_s = u_s + \varepsilon_s$, the target error acquires the additional term $(1-\beta)\varepsilon_s$. The coefficient is $1/(1+\alpha)$ for DifferenceFlow and $1-\alpha$ for AlphaFlow; these differ by $\alpha^2/(1+\alpha)$, the same quantity that separates the consistency coefficients in Section 5.5.

Example. Let $r = 0$ and $w(\tau) = 3\tau^2$, so that $U(\tau) = \tau^2$, $L = t$, $v = 3t^2$, and $v - u_t = 2t^2 = tU'(t)$. Then $\rho_\alpha = (1-\alpha)^2 t^2 - t^2 + 2\alpha t^2 = \alpha^2 t^2$, and Eqs. (8) and (9) give

$$T_{\text{DF}} - u_t = \frac{\alpha^2 t^2}{1+\alpha}, \qquad T_{\text{AF}} - u_t = (3\alpha^2 - \alpha^3)\, t^2.$$

D Evaluation Metrics

Let $\mathcal{G} = \{x_i^g\}_{i=1}^{N_g}$ and $\mathcal{R} = \{x_i^r\}_{i=1}^{N_r}$ denote the generated and reference image sets, respectively. For a representation model m and a feature pooling strategy p, let $\phi_{m,p}(x) \in \mathbb{R}^{d_m}$ denote the corresponding image representation.

D.1 Representation Fréchet Distance

For an image set $\mathcal{X} = \{x_i\}_{i=1}^N$, its empirical feature mean and covariance are defined as

$$\mu_{m,p}(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^N \phi_{m,p}(x_i).$$

$$\Sigma_{m,p}(\mathcal{X}) = \frac{1}{N-1}\sum_{i=1}^N \big(\phi_{m,p}(x_i) - \mu_{m,p}(\mathcal{X})\big)\big(\phi_{m,p}(x_i) - \mu_{m,p}(\mathcal{X})\big)^\top.$$

For compactness, define the Fréchet distance between two pairs of feature moments as

$$\mathcal{D}_{\text{F}}\big((\mu_1,\Sigma_1),(\mu_2,\Sigma_2)\big) = \|\mu_1 - \mu_2\|_2^2 + \text{Tr}\big(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}\big).$$

The representation Fréchet distance between the generated and reference sets is

$$\text{FD}_{m,p} = \mathcal{D}_{\text{F}}\big((\mu_{m,p}(\mathcal{G}),\Sigma_{m,p}(\mathcal{G})),(\mu_{m,p}(\mathcal{R}),\Sigma_{m,p}(\mathcal{R}))\big).$$

All reported FD metrics follow Section D.1 and differ only in the representation model, pooling strategy, and reference feature moments.

D.2 CLS and Average-Pooled Features

For a Vision Transformer representation model with a CLS token m, let

$$H_m(x) = [h_{\text{cls}}, h_1, \ldots, h_L], \qquad h_\ell \in \mathbb{R}^{d_m},$$

where h_{cls} denotes the CLS token and $h_1, \ldots, h_L$ denote the patch tokens.

The CLS representation is

$$\phi_{m,\text{cls}}(x) = h_{\text{cls}}.$$

Metric	Representation model	Feature pooling
FID(JiT) / FID(ADM)	InceptionV3	Inception pooled feature
MAE cls FD	MAE	CLS token
MAE avg FD	MAE	Patch-token average
DINOv2 cls FD	DINOv2	CLS token
DINOv2 avg FD	DINOv2	Patch-token average
CLIP cls FD	CLIP	CLS token
CLIP avg FD	CLIP	Patch-token average
SigLIP cls FD	SigLIP	Attention pooling
SigLIP avg FD	SigLIP	Patch-token average
ConvNeXt FD	ConvNeXt	Spatial global average

Table 8: Feature definitions used for the representation Fréchet distances.

For SigLIP, which does not use a CLS token, we use its attention-pooled representation, denoted by $\phi_{\mathrm{SigLIP,cls}}$ for consistency with the metric labels in our tables. Its average-pooled representation is computed by averaging all patch tokens.

The average-pooled representation is

$$\phi_{m,\mathrm{avg}}(x) = \frac{1}{L}\sum_{\ell=1}^{L} h_\ell.$$

For $m \in \{\mathrm{MAE}, \mathrm{DINOv2}, \mathrm{CLIP}, \mathrm{SigLIP}\}$, the CLS and average-pooled Fréchet distances are independently defined as

$$\mathrm{FD}_{m,\mathrm{cls}} = \mathcal{D}_{\mathrm{F}}\big((\mu_{m,\mathrm{cls}}(\mathcal{G}), \Sigma_{m,\mathrm{cls}}(\mathcal{G})), (\mu_{m,\mathrm{cls}}(\mathcal{R}), \Sigma_{m,\mathrm{cls}}(\mathcal{R}))\big).$$

$$\mathrm{FD}_{m,\mathrm{avg}} = \mathcal{D}_{\mathrm{F}}\big((\mu_{m,\mathrm{avg}}(\mathcal{G}), \Sigma_{m,\mathrm{avg}}(\mathcal{G})), (\mu_{m,\mathrm{avg}}(\mathcal{R}), \Sigma_{m,\mathrm{avg}}(\mathcal{R}))\big).$$

For ConvNeXt, let $F_m(x) \in \mathbb{R}^{d_m \times H \times W}$ denote its final spatial feature map. The image-level representation is obtained by global average pooling:

$$\phi_{\mathrm{ConvNeXt}}(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} F_m(x)_{:,h,w}.$$

The feature definitions used by the reported metrics are summarized in Table 8.

D.3 Inception Fréchet Distances

FID(JiT) and FID(ADM) use the same Inception representation of the generated images but different reference feature moments. Let $(\mu^g_{\mathrm{Inc}}, \Sigma^g_{\mathrm{Inc}})$ denote the generated Inception moments, and let $(\mu^r_{\mathrm{JiT}}, \Sigma^r_{\mathrm{JiT}})$ and $(\mu^r_{\mathrm{ADM}}, \Sigma^r_{\mathrm{ADM}})$ denote the two reference moment pairs. The corresponding metrics are

$$\mathrm{FID(JiT)} = \mathcal{D}_{\mathrm{F}}\big((\mu^g_{\mathrm{Inc}}, \Sigma^g_{\mathrm{Inc}}), (\mu^r_{\mathrm{JiT}}, \Sigma^r_{\mathrm{JiT}})\big).$$

$$\mathrm{FID(ADM)} = \mathcal{D}_{\mathrm{F}}\big((\mu^g_{\mathrm{Inc}}, \Sigma^g_{\mathrm{Inc}}), (\mu^r_{\mathrm{ADM}}, \Sigma^r_{\mathrm{ADM}})\big).$$

D.4 Relative Fréchet Distance

Let $\mathcal{V}$ denote the ImageNet validation set. The validation-set normalization factor for representation space m is

$$\mathrm{valFD}_m = \mathrm{FD}_m(\mathcal{V}, \mathcal{R}).$$

The relative Fréchet distance is defined as

$$\mathrm{FDr}_m = \frac{\mathrm{FD}_m(\mathcal{G}, \mathcal{R})}{\mathrm{FD}_m(\mathcal{V}, \mathcal{R})} = \frac{\mathrm{FD}_m}{\mathrm{valFD}_m}.$$

D.5 FDr-6

Let

$$\mathcal{M}_6 = \{\text{ADM}, \text{ConvNeXt}, \text{DINOv2}_{\text{cls}}, \text{MAE}_{\text{cls}}, \text{SigLIP}_{\text{cls}}, \text{CLIP}_{\text{cls}}\}.$$

FDr-6 is the unweighted arithmetic mean of the relative Fréchet distances over these six representation spaces:

$$\text{FDr-6} = \frac{1}{6} \sum_{m \in \mathcal{M}_6} \text{FDr}_m = \frac{1}{6} \sum_{m \in \mathcal{M}_6} \frac{\text{FD}_m}{\text{valFD}_m}.$$

D.6 Inception Score

For a generated image x, let $p(y \mid x)$ denote the conditional class distribution predicted by InceptionV3. The marginal class distribution over the generated set is

$$p(y) = \frac{1}{N_g} \sum_{i=1}^{N_g} p(y \mid x_i^g).$$

The Inception Score is

$$\text{IS} = \exp\left(\frac{1}{N_g} \sum_{i=1}^{N_g} D_{\text{KL}}\big(p(y \mid x_i^g) \,\|\, p(y)\big) \right).$$

When the generated samples are partitioned into $S = 10$ disjoint subsets, the reported score is the mean of the subset-level scores:

$$\text{IS}_{\text{reported}} = \frac{1}{S} \sum_{s=1}^{S} \text{IS}_s, \qquad S = 10.$$

E Algorithms Details

Algorithm 1 DifferenceFlow Training.

```
# fn(z, r, t): function to predict u
# x: training batch, k: training iterations
t, r = sample_t_r()
alpha = sample_alpha(k)
s = alpha * r + (1 - alpha) * t
e = randn_like(x)
zt = (1 - t) * x + t * e
v = e - x

if alpha == 0:
    u, dudt = jvp(fn, (zt, r, t), (v, 0, 1))
    u_tgt = v - (t - r) * dudt
else:
    u = fn(zt, r, t)
    zs = zt - (t - s) * v
    u_tgt = (α * v + fn(zs, r, s)) / (1 + α)

error = u - stopgrad(u_tgt)

loss = metric(error)
```

Algorithm 2 DifferenceFlow Scheduling.

```
# ks, ke: start and end schedule iterations
# gamma: sigmoid temperature parameter
# beta: logarithmic curvature parameter
# k: current iteration, eta: clamping value

# Constant
alpha = alphaConst
q = (k - ks) / (ke - ks)

# Sigmoid
alpha = 1 - sigmoid((q - 1 / 2) * gamma)

# Linear
alpha = 1 - q

# Logarithmic
alpha = 1 - log(1 + beta * q) / log(1 + beta)
alpha = 1 if k < ks else 0 if k > ke
alpha = 1 if alpha > (1-eta) else 0 if alpha
     < eta
```

F Additional Visualizations

F.1 CIFAR-10 Visualizations

F.2 ImageNet-1000 Visualizations



Figure 4: **Unconditional generation for Cifar.**



Figure 5: Conditional generation for Class 42 for NFE=1.



Figure 6: Conditional generation for Class 108 for NFE=1.



Figure 7: Conditional generation for Class 140 for NFE=1.

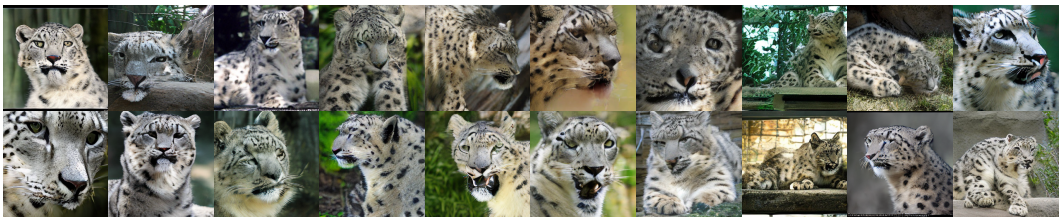


Figure 8: Conditional generation for Class 289 for NFE=1.



Figure 9: Conditional generation for Class 698 for NFE=1.



Figure 10: Conditional generation for Class 963 for NFE=1.



Figure 11: Conditional generation for Class 975 for NFE=1.



Figure 12: Conditional generation for Class 985 for NFE=1.