Bidirectional Trajectory Smoothing for Training-Free Image Generation with Rectified Flows

Yan Luo [1] **Henry Huang** [1] **Mengyu Wang** [1]

Abstract

Recent advances have reformulated diffusion models as deterministic ordinary differential equations (ODEs) through the framework of flow matching, providing a unified formulation for the noise-to-data generative process. Various training-free flow matching approaches have been developed to improve image generation through flow velocity field adjustment, eliminating the need for costly retraining. However, Modifying the velocity field v introduces errors that propagate through the full generation path, whereas adjustments to the latent trajectory z are naturally corrected by the pretrained velocity network, reducing error accumulation. In this paper, we propose two complementary training-free latent-trajectory adjustment approaches based on future and past velocity v and latent trajectory z information that refine the generative path directly in latent space. We propose two training-free trajectory smoothing schemes: *Look-Ahead*, which averages the current and next-step latents using a curvature-gated weight, and *Look-Back*, which smoothes latents using an exponential moving average with decay. We demonstrate through extensive experiments and comprehensive evaluation metrics that the proposed training-free trajectory smoothing models substantially outperform various state-of-the-art models across multiple datasets including COCO17, CUB-200, and Flickr30K.

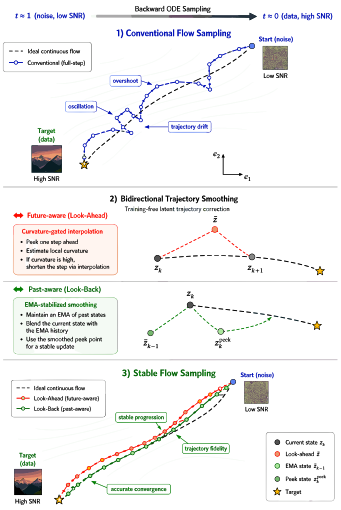


Figure 1. Conceptual illustration of training-free trajectory smoothing for flow sampling. Without trajectory smoothing (top), backward integration of the flow ODE suffers divergence and overshoot in low Signal-to-Noise Ratio (SNR) regions, causing the discrete trajectory to deviate from the ideal continuous flow and producing final samples that inaccurately reach the target distribution. With the trajectory smoothing mechanism (bottom), the trajectory maintains robust fidelity to the ideal continuous flow across both low and high SNR regions, ensuring stable progression and accurate convergence.

1. Introduction

Recently, diffusion models have been reformulated within the mathematical framework of ordinary differential equations (ODEs), known as *flow matching*, which defines a deterministic ODE-governed formulation of noise-to-data generative process (Lipman et al., 2023; Chen & Lipman, 2024; Albergo & Vanden-Eijnden, 2023; Luo et al., 2025). Among these ODE-based approaches, *Rectified Flow* has emerged as a particularly effective formulation that uses a linear and constant-velocity transport between noise and data as a training target (Liu et al., 2022; 2023; Esser et al., 2024). This simplification not only leads to more stable training and faster convergence (Lee et al., 2024; Wang et al., 2024a) but also enables efficient few-step generation and straightforward distillation of pretrained flow models (Kornilov et al., 2024; Liu et al., 2024). More recently, a series of *training-free* approaches have further advanced this line of work by adapting pretrained flow models through velocity re-parameterization or trajectory refinement, eliminating the need for costly retraining (Wang et al., 2025b; Bu et al., 2025; Li et al., 2025; Jin et al., 2025). These methods are especially appealing because they achieve substantial improvements in image generation fidelity and image editing capability with minimal computational overhead (Wang et al., 2025a; Kulikov et al., 2025; Avrahami et al., 2025), making high-quality generation and editing accessible even under limited resources.

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

Most training-free flow methods focus on image editing using pretrained rectified-flow or diffusion models without retraining. Stable Flow (Avrahami et al., 2025) identifies vital layers for improved editing; RF-Edit (Wang et al., 2024c) preserves structure by reusing stored self-attention features; RF-Inversion (Rout et al., 2024) uses an optimal-control adjustment of the image-to-noise trajectory for faithful zero-shot editing. FlowEdit (Kulikov et al., 2025) constructs a direct flow between source- and target-prompt images for inversion-free editing. FlowChef (Patel et al., 2025) nudges the latent trajectory using a task loss on the predicted clean image. SplitFlow (Yoon et al., 2025) decomposes a target prompt into semantic subflows and re-assembles them for higher-fidelity, inversion-free editing.

By contrast, only a few training-free flow methods focus on improving general image generation. Rectified Diffusion (Wang et al., 2024b) enhances fidelity by finetuning on synthetic samples. HiFlow (Bu et al., 2025) builds a virtual high-resolution reference flow to guide generation, while OC-Flow (Wang et al., 2025b) steers trajectories via reward-driven control velocities. A-Euler (Jin et al., 2025) enforces near-linear velocity by decomposing it into drift and residual components, and Self-Guidance (Li et al., 2025) stabilizes denoising by smoothing velocity with past-step corrections.

In this paper, we propose two complementary training-free latent-trajectory adjustment approaches based on future and past velocity v and latent trajectory z information that refine the generative path directly in latent space, in contrast to prior methods that modify the velocity field. Modifying the velocity field v introduces errors that propagate along the entire generation path, whereas adjustments to the latent trajectory z are subsequently regularized by the pretrained velocity network, limiting error accumulation.

The contributions of our paper are as follows:

- We propose a training-free *Look-Ahead* scheme that smoothes the latent trajectory with weighted average of current z and next-step z gated by spatial curvature.
- We introduce a training-free *Look-Back* scheme that smoothes the latent trajectory with exponential moving average of latent state z with a decay.
- We demonstrate through extensive experiments and comprehensive evaluation metrics that the proposed training-free trajectory smoothing models substantially outperforms various state-of-the-art models.

2. Related Work

Flow Matching and Its Variants Flow Matching (Lipman et al., 2023) formulates generative modeling as learning a continuous-time velocity field that transports samples from a simple prior distribution to the target data distribution via an ordinary differential equation (ODE). Subsequent works have extended this framework to various geometric and architectural settings, such as Flow Matching on general geometries (Chen & Lipman, 2024), discrete domains (Gat et al., 2024), and rigid-body manifolds through SE(3)-Stochastic Flow Matching (Bose et al., 2024). The Rectified Flow formulation (Liu et al., 2023) further simplifies the probability path by enforcing a linear and constant-velocity transport between data and noise, leading to faster convergence and trajectory-straightened flows. Follow-up studies explore optimal transport straightness (Wang et al., 2024a), one-step trajectory learning (Kornilov et al., 2024), and improved training objectives (Lee et al., 2024), while others incorporate adaptive control (Wang et al., 2025b), reinforcement learning (Liu et al., 2025), or velocity decomposition (Jin et al., 2025) to enhance stability and efficiency. Training-free rectified-flow variants, such as HiFlow (Bu et al., 2025) and Self-Guidance (Li et al., 2025), demonstrate that pretrained flow models can be refined through flow-aligned or self-consistent guidance without retraining. Collectively, these advances establish Rectified Flow as a unifying paradigm for deterministic generation, underscoring the widespread adoption of Stable Diffusion v3.5 (Esser et al., 2024) in modern generative modeling pipelines.

Training-Free Flow-Based Image Generation Methods Most existing training-free flow models primarily target image editing (Avrahami et al., 2025; Wang et al., 2024c; Rout et al., 2024; Kulikov et al., 2025; Patel et al., 2025; Yoon et al., 2025), where pretrained flow matching models are adapted without additional training to enable high-quality and controllable visual modifications. By contrast, relatively few training-free flow-based methods target improvements in general image generation. For instance, Rectified Diffusion (Wang et al., 2024b) Rectified Diffusion demonstrates that leveraging synthetic samples generated by a pretrained diffusion or flow-matching model to perform a secondary finetuning stage leads to consistent improvements in downstream generation fidelity and sample quality. HiFlow (Bu et al., 2025) establishes a virtual reference flow within the high-resolution space that effectively captures the characteristics of low-resolution flow information, offering guidance for high-resolution generation. OC-Flow (Wang et al., 2025b) maximizes a chosen image-level reward such as CLIP similarity and aesthetic score to optimally steer the flow trajectory by adding a

term of control velocity to produce images that satisfy the desired condition. Training-free velocity-modification methods such as OC-Flow show strong potential for improving image generation quality by directly steering the flow trajectory without retraining. Along this line of research, A-Euler (Jin et al., 2025) accelerates few-step sampling by adaptively decomposing the velocity field into a linear drift and a temporally-suppressed residual, effectively enforcing a near-linear velocity trajectory. Self-Guidance (Li et al., 2025) smooths the sampling trajectory by correcting the current velocity using a weighted difference from the noisier immediate-past velocity, leading to more stable denoising and improved image generation quality. Both A-Euler and Self-Guidance temporally smooth the velocity field to stabilize the denoising trajectory and improve image generation quality.

3. Problem Statement: Training-Free Trajectory Smoothing

Let p_0 denote the target data distribution and $p_1 = \mathcal{N}(0, I)$ the noise prior. Flow-based generative models, including rectified flow and its variants, define a continuous probability path $(p_t)_{t\in[0,1]}$ governed by the ordinary differential equation (ODE)

$$\frac{dz_t}{dt} = v_\Theta(z_t, t, c), \qquad z_1 \sim p_1,\ z_0 \sim p_0, \tag{1}$$

where v_Θ denotes a learned velocity field parameterized by Θ and conditioned on optional context c.

During inference, the path is integrated backward in time using discrete timesteps $\{t_k\}_{k=0}^{K}$. However, the numerical integration of this flow ODE is often unstable due to high local curvature in the learned velocity field v_Θ, stiffness and oscillations in regions of low signal-to-noise ratio (SNR), and the limited step sizes prescribed by schedulers originally tuned for diffusion models. Such instabilities manifest as divergence, overshooting, or loss of trajectory fidelity, leading to degraded sample quality and requiring either retraining or costly higher-order solvers.

Objective The goal is to design a training-free, lightweight stabilization mechanism that improves numerical stability and trajectory fidelity during backward integration of the flow ODE, operates without additional model evaluations or retraining of v_Θ, and remains fully compatible with native scheduler configurations used in diffusion or rectified-flow sampling. This mechanism should enhance robustness while preserving the efficiency and generality of standard inference pipelines.

Formal Optimization View Given a discrete trajectory $\{z_k\}$ governed by the update rule

$$z_{k+1} = \Phi_\Theta(z_k, t_k, \eta_k, c), \tag{2}$$

we seek a correction operator $\mathcal{R}$ such that

$$\begin{aligned} &\min_{\mathcal{R}}\ \mathbb{E}_{t,z}\left[\|z_{k+1}^{\mathcal{R}} - \psi_t(z_1)\|_2^2\right] \\ &\text{s.t.}\quad z_{k+1}^{\mathcal{R}} = \mathcal{R}\big(\Phi_\Theta(z_k, t_k, \eta_k, c)\big), \end{aligned} \tag{3}$$

where $\psi_t(\cdot)$ denotes the ideal continuous flow trajectory. To achieve stable and faithful integration without retraining, we introduce two training-free correction schemes for $\mathcal{R}$. The first, the *Look-Ahead* scheme, stabilizes the trajectory by referencing future curvature trends to anticipate deviations and adjust the step accordingly. The second, the *Look-Back* scheme, achieves complementary stability by referencing past averaged states, effectively damping high-frequency oscillations while maintaining trajectory fidelity. Both mechanisms adaptively regulate the integration process to preserve the expected flow dynamics of the learned velocity field v_Θ.

4. Look-Ahead Scheme for Flow Sampling

Let p_0 be the data distribution and $p_1 = \mathcal{N}(0, I)$ the noise prior. A rectified-flow model defines a probability path $(p_t)_{t\in[0,1]}$ governed by

$$\frac{dz_t}{dt} = v_\Theta(z_t, t, c), \qquad z_1 \sim p_1,\ z_0 \sim p_0, \tag{4}$$

where v_Θ denotes a learned velocity field conditioned on context c. Sampling integrates (4) backward on a grid $1 = t_0 > \cdots > t_K = 0$ with steps $\Delta_k = t_k - t_{k+1} > 0$. The forward Euler update is

$$z_{k+1}^{\text{Eul}} = z_k - \eta_k\, v_\Theta(z_k, t_k, c), \tag{5}$$

where η_k follows the scheduler's native step configuration.

To improve stability without retraining, we introduce a *training-free look-ahead* scheme that inspects the local velocity trend: it accepts long steps in nearly straight directions but interpolates conservatively under high curvature, keeping constant cost per step.

Predictor and Efficient Velocity Estimation At (z_k, t_k), we evaluate the instantaneous velocity $v_k = v_\Theta(z_k, t_k, c)$ and perform the scheduler's native predictor step, $\tilde{z} = \text{SchedulerStep}(z_k, v_k, t_k)$ with $\tilde{t} = t_k - \Delta_k$. Instead of re-evaluating $v_\Theta(\tilde{z}, \tilde{t}, c)$, we estimate the peek velocity by finite difference:

$$\tilde{v} \approx \frac{z_k - \tilde{z}}{\Delta_k}. \tag{6}$$

This captures the effective directional change implied by the scheduler without doubling model evaluations.

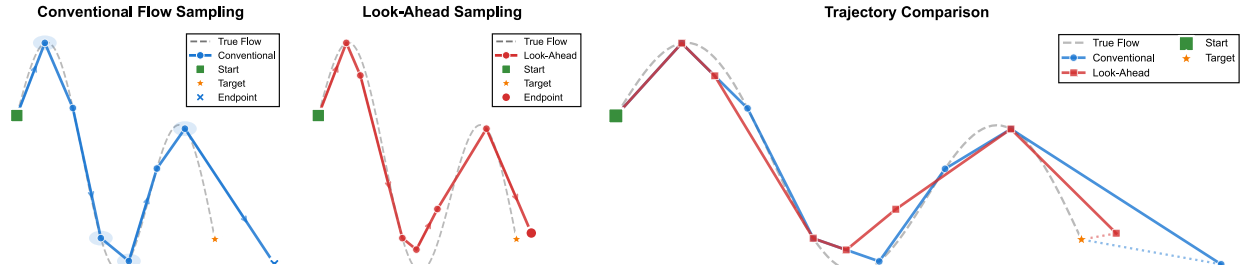


Figure 2. Schematic view of the proposed look-ahead sampling. Conventional flow sampling always takes full steps, which can overshoot in regions of high curvature and lead to a large deviation from the target. In contrast, the Look-Ahead scheme adaptively interpolates based on local curvature, modulating step sizes to better follow the underlying flow and achieve a significantly smaller endpoint error.

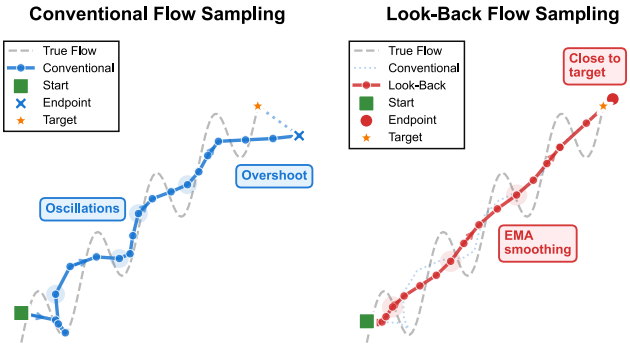


Figure 3. Schematic view of the proposed look-back sampling. Conventional sampling exhibits oscillations and overshoots the target, while Look-Back produces a smooth trajectory through exponential state averaging.

Curvature-Gated Interpolation We quantify local curvature as the normalized directional deviation:

$$\kappa_k = \frac{\|\tilde{v} - v_k\|_2}{\|\tilde{z} - z_k\|_2 + \varepsilon}, \tag{7}$$

where $\varepsilon = 10^{-8}$ ensures numerical stability by preventing division-by-zero. If $\kappa_k \leq \tau_{\text{curv}}$, the full step is accepted, $z_{k+1} = \tilde{z}$, $t_{k+1} = \tilde{t}$. Otherwise, a partial interpolation mitigates instability:

$$z_{k+1} = z_k + \gamma(\tilde{z} - z_k), \qquad t_{k+1} = \tilde{t}, \tag{8}$$

where $\gamma \in (0, 1)$ controls conservativeness. The timestep always advances to preserve scheduler synchronization. This curvature-gated interpolation adaptively balances efficiency and stability: when the local trajectory is smooth (κ_k small), it takes confident, full steps to accelerate sampling; when curvature increases, the interpolation automatically dampens the update, preventing divergence or oscillation. In essence, the gate acts as a lightweight, geometry-aware step controller that preserves trajectory fidelity without extra model evaluations.

Algorithm Algorithm 1 summarizes the full inference routine. It iteratively integrates the flow ODE using a scheduler-consistent predictor, estimates future velocity via a single finite-difference peek, and adaptively accepts or interpolates each step based on local curvature, achieving stability and efficiency without additional model evaluations, as illustrated in Figure 2.

Algorithm 1 Look-Ahead Sampling with Curvature Gate

Require: trained field $v_\Theta(\cdot, \cdot, c)$, scheduler timesteps $\{t_k\}_{k=0}^{K}$ with $\Delta_k = t_k - t_{k+1}$, initial latent $z_0 \sim p_1$ at $t_0 = 1$, curvature threshold $\tau_{\text{curv}} > 0$, interpolation factor $\gamma \in (0, 1)$, numerical epsilon ε

$z \leftarrow z_0$, $t \leftarrow t_0$
for $k = 0, 1, \ldots, K-1$ **do**
  $v \leftarrow v_\Theta(z, t, c)$
  $\tilde{z} \leftarrow \text{SchedulerStep}(z, v, t)$, $\tilde{t} \leftarrow t - \Delta_k$
  $\tilde{v} \leftarrow (z - \tilde{z})/\Delta_k$
  $\kappa \leftarrow \|\tilde{v} - v\|_2 / (\|\tilde{z} - z\|_2 + \varepsilon)$
  if $\kappa \leq \tau_{\text{curv}}$ **then**
    $z \leftarrow \tilde{z}$ {accept full step}
  else
    $z \leftarrow z + \gamma(\tilde{z} - z)$ {interpolate}
  end if
  $t \leftarrow \tilde{t}$
end for
return z

Reduction to Conventional Flow Sampling When $\tau_{\text{curv}} = \infty$ or $\gamma = 1$, the curvature gate never triggers interpolation, it becomes identical to conventional flow sampling.

5. Look-Back Scheme for Flow Sampling

While the Look-Ahead scheme stabilizes integration by probing future trends, the *Look-Back* scheme complements it by peeking into the past. It leverages exponentially averaged latents to damp high-frequency oscillations, providing a memory-efficient and training-free stabilization mechanism.

Exponential State Averaging We maintain an exponentially weighted history of latent states:

$$\bar{z}_k = \gamma(t_k)\,\bar{z}_{k-1} + \big(1-\gamma(t_k)\big)\,z_k, \quad \bar{z}_{-1} = z_0, \quad (9)$$

where $\gamma(t) \in [0,1)$ controls the decay rate. The averaged state $\bar{z}_k$ acts as a low-variance memory of the trajectory, filtering transient noise while preserving the coarse structure of the flow dynamics. As $\gamma(t)$ decreases, $\bar{z}_k$ becomes more responsive to the current latent, ensuring the averaging influence vanishes near convergence. Exponential state averaging functions as a temporal low-pass filter, suppressing jitter during mid-range timesteps with high stochastic variance. This yields smoother latent paths and improved numerical robustness for rectified-flow dynamics.

Backward-Referenced yet Forward-Stabilizing Unlike the look-ahead method that extrapolates into the future, the Look-Back scheme stabilizes integration by referencing past states. Velocity is evaluated at a blended latent that combines the current state with its exponentially smoothed history:

$$z_k^{\text{peek}} = (1-\lambda)\,z_k + \lambda\,\bar{z}_{k-1}, \qquad \lambda \in [0,1]. \quad (10)$$

Here, $\bar{z}_{k-1}$ captures the slow manifold of recent dynamics, serving as a denoised, low-frequency reference. Although the mechanism peeks backward, its effect is *forward-stabilizing*: the blended latent aligns velocity evaluation with the stable trajectory toward which the flow is converging. The corresponding update reads

$$z_{k+1} = z_k - \eta_k\, v_\Theta(z_k^{\text{peek}}, t_k, c), \quad (11)$$

where v_Θ is the learned velocity field. This implicit damping mitigates overshooting and oscillatory motion without biasing the underlying flow manifold. By peeking into the past, the sampler anticipates the stable manifold of the trajectory, reducing variance and promoting smooth convergence in stiff or noisy regions of the flow.

SNR-Aware Decay Scheduling To adapt the smoothing strength across noise levels, we define $\gamma(t)$ as a logistic function of the log-SNR:

$$\gamma(t) = \gamma_{\max} \cdot \sigma\Big(\beta[\xi(t) - \xi_*]\Big), \quad (12)$$

where σ denotes the sigmoid, β controls transition steepness, and $\xi(t) = \log(a_t^2/b_t^2)$ is the log-SNR derived from the generative process $z_t = a_t x_0 + b_t \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. For rectified flows, $a_t = 1-t$ and $b_t = t$ yield $\xi(t) = 2\log\frac{1-t}{t}$; for diffusion models, $a_t = \sqrt{\bar{\alpha}_t}$ and $b_t = \sqrt{1-\bar{\alpha}_t}$, so $\xi(t) = \log(\bar{\alpha}_t/(1-\bar{\alpha}_t))$. The midpoint ξ_* marks maximal uncertainty: when $\xi(t) < \xi_*$, $\gamma(t)$ stays high for strong smoothing; as $\xi(t) \to \infty$, $\gamma(t) \to 0$, recovering the native solver near convergence.

Algorithm Algorithm 2 outlines the complete inference procedure for the Look-Back sampling, as illustrated in Figure 3. At each step, it maintains an exponentially averaged latent state, peeks into this smoothed history to evaluate a stabilized velocity, and updates the current latent accordingly, achieving noise-averaged, training-free flow sampling with improved stability and smoothness.

Algorithm 2 Look-Back Sampling for Flow Models

Require: v_Θ, schedule $\{(t_k, \eta_k)\}_{k=0}^{K-1}$, decay $\gamma(t)$, blend λ, condition c

$z_0 \sim \mathcal{N}(0, I)$; $\bar{z}_{-1} \leftarrow z_0$
for $k = 0$ to $K-1$ **do**
  $\bar{z}_k \leftarrow \gamma(t_k)\bar{z}_{k-1} + \big(1-\gamma(t_k)\big)z_k$
  $z_k^{\text{peek}} \leftarrow (1-\lambda)z_k + \lambda\bar{z}_{k-1}$
  $v_k \leftarrow v_\Theta(z_k^{\text{peek}}, t_k, c)$
  $z_{k+1} \leftarrow z_k - \eta_k v_k$
end for
return z_K

Reduction to Conventional Flow Sampling When $\lambda = 0$, the blended latent in (10) reduces to the current state z_k, and the update rule in (11) becomes identical to the standard Euler step.

6. Complexity Analysis

Let C_v denote the cost of one model (velocity field) evaluation $v_\Theta(z, t, c)$, K the total number of sampling steps, and d the latent dimensionality. Both proposed schemes preserve the same asymptotic complexity as standard flow sampling, requiring only one model evaluation per step. As summarized in Table 3, Look-Ahead and Look-Back introduce minor $\mathcal{O}(K, d)$ vector operations for curvature gating and exponential averaging, resulting in negligible runtime overhead and slightly higher memory usage for the running average in Look-Back. Overall, both methods achieve improved stability and smoothness at nearly identical computational cost to the baseline.

Table 1. Complexity comparison of different flow sampling schemes. C_v denotes a single model evaluation cost; d is latent dimension.

Method	Model Calls / Step	Time Complexity	Extra Memory
Conventional Flow Sampling	1	$\mathcal{O}(K, C_v)$	$\mathcal{O}(d)$
Look-Ahead	1	$\mathcal{O}(K, C_v + K, d)$	$\mathcal{O}(d)$
Look-Back	1	$\mathcal{O}(K, C_v + K, d)$	$\mathcal{O}(2d)$

7. Experiment & Analysis

Experimental Set-Up We evaluate all methods under identical SDv3.5 inference settings, comparing the baseline sampler with the proposed Look-Ahead and Look-

Table 2. Performance comparison of different methods on COCO17, CUB-200, and Flickr30K.

Dataset	Method	FID ↓	IS ↑	CLIPScore ↑	Caption-based Metrics			
					BLEU-4 ↑	METEOR ↑	ROUGE-L ↑	CLAIR ↑
COCO17 (Lin et al., 2014)	SDv3.5	28.46	33.64	0.3284	7.99	29.17	35.11	71.45
	SDv3.5 w/ A-Euler (Jin et al., 2025)	27.64	34.63	0.3284	7.96	29.09	35.17	71.23
	SDv3.5 w/ Self-Guidance (Li et al., 2025)	40.98	28.54	0.3121	6.52	26.18	31.59	62.26
	SDv3.5 w/ Momentum	29.36	33.27	**0.3329**	8.26	29.43	35.27	71.41
	SDv3.5 w/ Look-Ahead	**26.17**	34.60	0.3296	**8.82**	30.45	36.15	**72.99**
	SDv3.5 w/ Look-Back	26.27	**34.81**	0.3294	8.76	**30.53**	**36.17**	72.77
CUB-200 (Welinder et al., 2010)	SDv3.5	24.92	4.81	0.3240	0.13	16.57	17.64	66.69
	SDv3.5 w/ A-Euler (Jin et al., 2025)	22.98	5.05	0.3250	0.12	16.39	17.51	66.22
	SDv3.5 w/ Self-Guidance (Li et al., 2025)	61.16	4.98	0.3125	**0.18**	**17.85**	17.78	62.28
	SDv3.5 w/ Momentum	26.25	4.67	0.3264	**0.18**	17.02	**17.94**	66.57
	SDv3.5 w/ Look-Ahead	21.99	5.12	0.3264	0.14	16.64	17.65	**66.78**
	SDv3.5 w/ Look-Back	**19.73**	**5.32**	**0.3296**	0.15	16.81	17.86	66.65
Flickr30K (Plummer et al., 2015)	SDv3.5	79.58	17.95	0.3379	4.12	23.88	29.30	68.03
	SDv3.5 w/ A-Euler (Jin et al., 2025)	78.57	17.83	0.3371	3.91	22.99	29.00	67.56
	SDv3.5 w/ Self-Guidance (Li et al., 2025)	93.07	14.44	0.3160	3.35	20.80	26.27	57.35
	SDv3.5 w/ Momentum	79.05	17.56	**0.3420**	4.18	24.27	29.32	67.20
	SDv3.5 w/ Look-Ahead	75.88	18.12	0.3388	**4.97**	**25.32**	**30.58**	**70.16**
	SDv3.5 w/ Look-Back	**75.62**	**18.30**	0.3386	4.73	25.26	30.07	69.43

Back schemes, as well as prior training-free baselines A-Euler (Jin et al., 2025) and Self-Guidance (Li et al., 2025). All models adopt a 25-step sampling schedule with classifier-free guidance (CFG) = 7, ensuring that performance variations reflect only the stabilization behavior. The chosen configuration follows recommendations from prior studies (Liu et al., 2023; Lu et al., 2022; Wang et al., 2024b) and the HuggingFace guidelines. A single random seed is sampled once and fixed across all experiments. All hyperparameters related to the proposed methods are provided in the appendix.

We conduct evaluations on three benchmark datasets: COCO17 (Lin et al., 2014) (validation set), CUB-200 (Welinder et al., 2010) (test set), and Flickr30K (Plummer et al., 2015) (test set), covering diverse image–text domains for assessing both visual fidelity and semantic alignment. All images are generated at a resolution of 512×512, following the protocol in (Ma et al., 2024).

We evaluate image generation quality using FID (Heusel et al., 2017), IS (Salimans et al., 2016), and CLIP-Score (Hessel et al., 2021) to assess visual fidelity and semantic alignment. Additionally, captions for generated images are produced using BLIPv2 (Li et al., 2023) and compared with ground-truth captions via language-based metrics including BLEU-4 (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), and CLAIR (Chan et al., 2023), providing comprehensive evaluation of text–image consistency.

Performance As shown in Table 2, the proposed Look-Ahead and Look-Back schemes consistently outperform existing training-free samplers in both fidelity and semantic alignment. On COCO17, Look-Ahead achieves the lowest FID of 26.17 and the highest BLEU-4 of 8.82, surpassing A-Euler (27.64 FID) and Momentum (29.36 FID). Look-Back attains a similar FID (26.27) while further improving IS to 34.81 and METEOR to 30.53. These results indicate that curvature-aware interpolation and exponential latent averaging jointly enhance the numerical stability of ODE integration, reducing oscillations without extra model evaluations. In particular, both methods substantially outperform Self-Guidance (40.98 FID, 28.54 IS), which modifies the velocity field but often amplifies instability in rectified-flow inference. The observed gains in CLAIR, 72.99 for Look-Ahead and 72.77 for Look-Back, demonstrate their effectiveness in maintaining high-level semantic coherence while stabilizing low-level integration dynamics.

The advantages of the proposed methods generalize across diverse datasets with distinct statistical characteristics. On CUB-200, Look-Ahead and Look-Back reduce the FID from 24.92 (baseline) to 21.99 and 19.73, respectively. The improvement of about 5 points indicates enhanced image fidelity and better integration stability. On Flickr30K, Look-Ahead improves BLEU-4 from 4.12 to 4.97 and ROUGE-L from 29.30 to 30.58, showing that curvature-gated correction strengthens text-image consistency. Meanwhile, Look-Back achieves the lowest FID (75.62 vs. 79.58 baseline) and the highest IS (18.30 vs. 17.95 baseline), confirming smoother and more reliable convergence in the presence of high-curvature dynamics. Together, these results demonstrate that the proposed Look-Ahead and Look-Back schemes not only stabilize the numerical integration but also consistently improve both visual fidelity and semantic alignment across datasets of varying complexity.

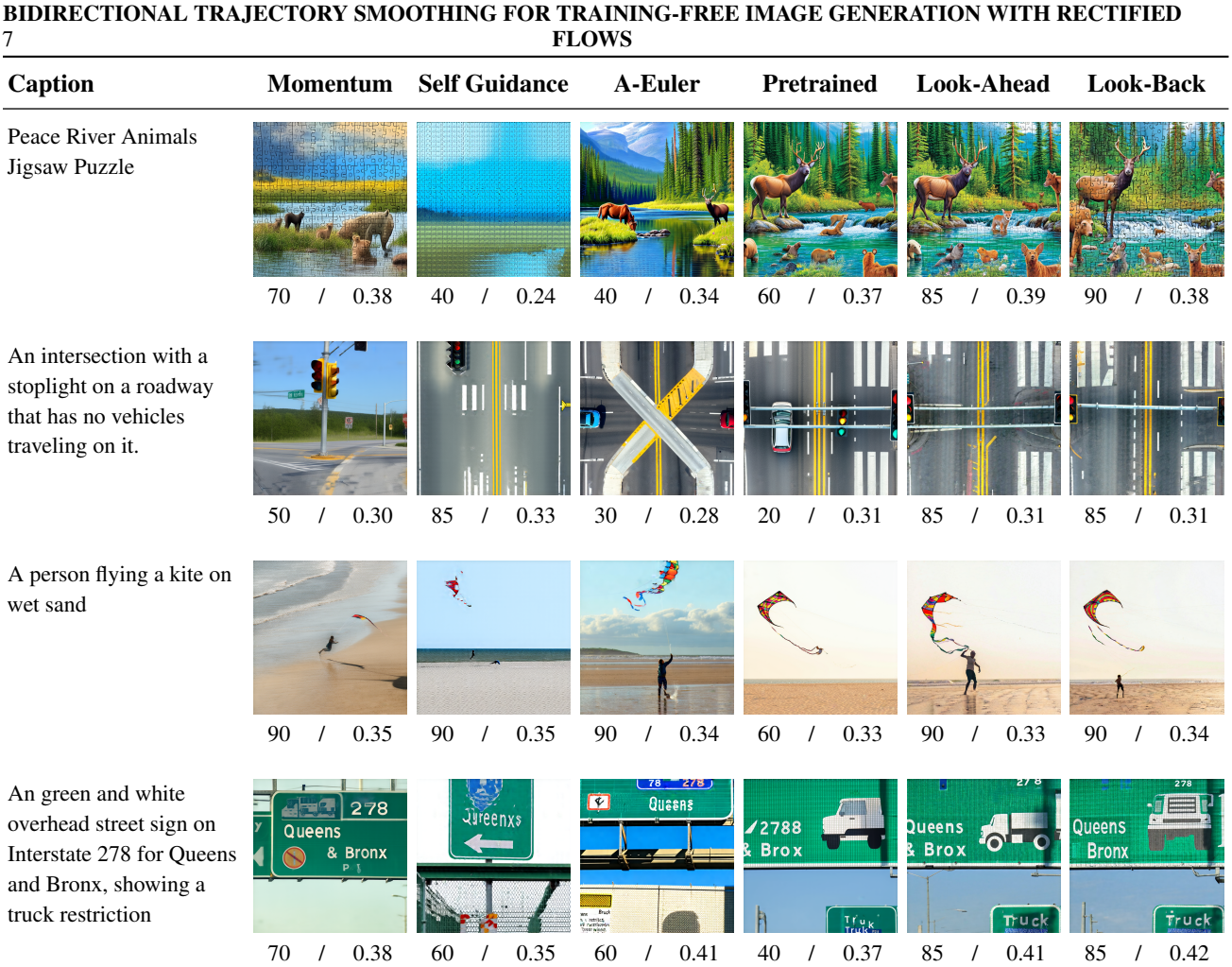


Figure 4. Qualitative comparison showing LookAhead and LookBack produce higher quality images with better coherence and detail than baseline methods. Scores shown are CLAIR / CLIPScore.

Qualitative Analysis Figure 4 qualitatively highlights that the proposed Look-Ahead and Look-Back schemes produce sharper, more coherent, and semantically faithful images than all baselines. For instance, in the Peace River Animals Jigsaw Puzzle example, both methods recover fine-grained details in the animal textures and forest background that are lost in Momentum or A-Euler, while maintaining natural composition and color consistency. Similarly, in the Street sign example, Look-Ahead and Look-Back clearly render the "Queens / Bronx" text and truck icon with legible edges, whereas baselines produce fragmented signage.

Figure 5 reveals clear qualitative advantages of the proposed schemes over standard sampling. For the astronaut composition, Look-Ahead and Look-Back generate markedly richer cosmic scenes with more intricate starfield details and refined suit textures, whereas standard sampling produces a flatter, less detailed result. In the portrait, the key difference lies in the realistic raindrop details rendered on the woman's coat by both proposed methods, an atmospheric nuance completely absent in the standard sampling.

Ablation Study Figure 6 analyzes the sensitivity of the proposed Look-Ahead and Look-Back schemes to their key hyperparameters, revealing how curvature gating and exponential averaging jointly govern stability and fidelity during ODE integration. In the Look-Ahead scheme, both the curvature threshold τ_{curv} and interpolation factor γ control the balance between numerical conservativeness and trajectory progress. As shown in Figures 6(a)–(b), excessively large τ_{curv} values permit unstable full-step updates resembling Euler behavior, while overly restrictive thresholds hinder convergence; the optimal regime emerges around $\tau_{\text{curv}} = 1$ and $\gamma = 0.9$, which achieve the lowest FID (75.88) and highest CLIPScore (0.3388), confirming that moderate curvature moderation yields the

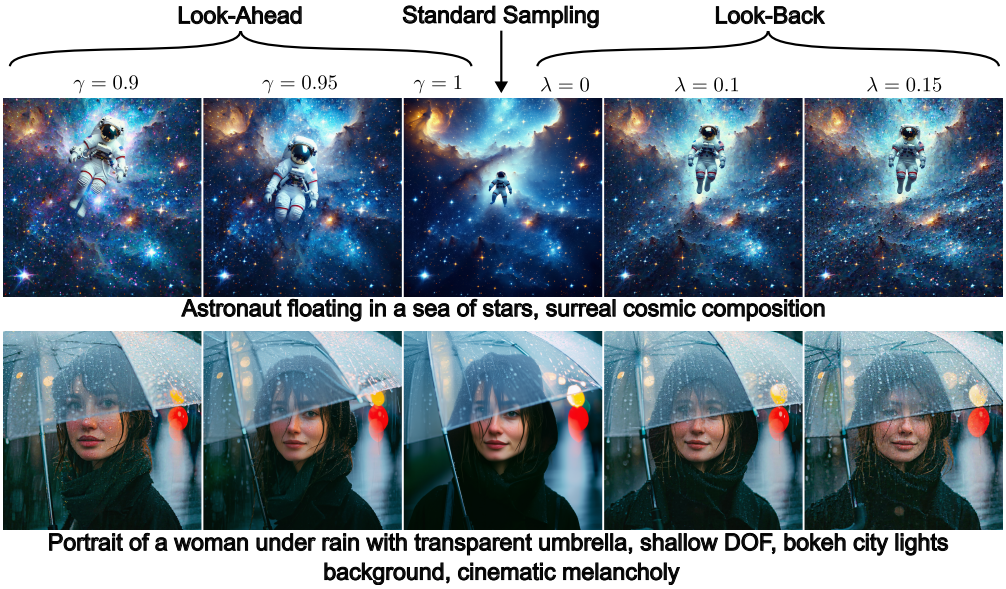
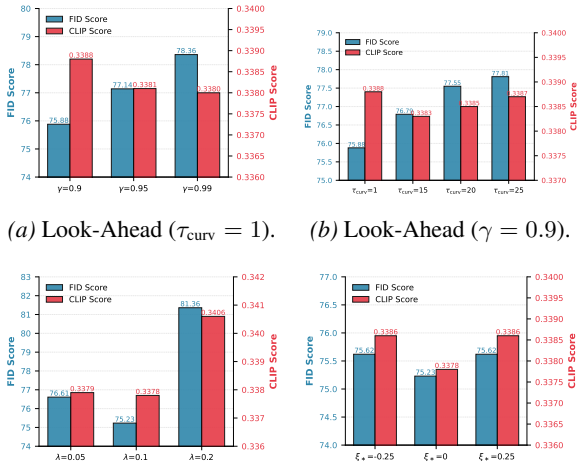


Figure 5. Visual effects of different γ (Look-Ahead) and λ (Look-Back). The Look-Ahead and Look-Back generations exhibit richer and more intricate visual details in the astronaut compared to the standard sampling. In the rainy portrait, the Look-Ahead and Look-Back generations produce more realistic raindrop details on the girl's coat, making the scene more consistent with a rainy atmosphere, whereas the standard sampling fails to capture such effects.



(a) Look-Ahead ($\tau_{\text{curv}} = 1$). *(b)* Look-Ahead ($\gamma = 0.9$).

(c) Look-Back ($\xi_* = 0$). *(d)* Look-Back ($\lambda = 0.1$).

Figure 6. Ablation study of key hyperparameters related to the proposed methods on Flickr30K.

best trade-off between efficiency and smoothness. In the Look-Back scheme, Figures 6(c)–(d) illustrate that small blending weights and a centered SNR midpoint ($\xi^* = 0$) produce the most stable results. Larger λ over-smooths latent trajectories and biases the generative path, whereas proper SNR-dependent decay $\gamma(t)$ ensures that smoothing fades near convergence, allowing high-frequency details to re-emerge. Collectively, these ablations demonstrate that both schemes are robust within a broad parameter range and validate the importance of curvature-aware moderation and adaptive temporal averaging for stable training-free flow sampling.

Runtime Under identical settings, SDv3.5 requires 11.0 seconds per image, while Look-Ahead and Look-Back take 11.4 and 11.2 seconds, respectively, introducing only a marginal overhead.

8. Conclusion

We presented two complementary training-free latent-trajectory smoothing methods, Look-Ahead and Look-Back, which refine the flow-matching generative process directly in latent space. Unlike prior approaches that modify the velocity field and risk accumulating errors along the sampling path, our latent-trajectory adjustments benefit from the intrinsic corrective behavior of the pre-trained velocity network, leading to more stable and robust generation. Look-Ahead leverages curvature-gated averaging with future latents, while Look-Back applies an exponentially decayed moving average informed by past states. Extensive experiments on COCO17, CUB-200, and Flickr30K show that our methods consistently outperform strong state-of-the-art baselines across a wide range of metrics. These results highlight latent-trajectory smoothing as an effective and general strategy for improving training-free flow-based image generation.

Impact Statement

This work introduces Look-Ahead and Look-Back, training-free mechanisms that enhance the numerical stability and trajectory fidelity of flow-based generative models. By correcting discretization errors such as divergence and overshooting, these methods ensure that the generated output adheres more strictly to the semantic intent and distribution defined by the pre-trained velocity field.

There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. Because our contribution is restricted to a numerical stabilization technique that does not modify model parameters or training data, it introduces no new ethical dimensions or risks independent of the pre-trained model weights.

References

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Avrahami, O., Patashnik, O., Fried, O., Nemchinov, E., Aberman, K., Lischinski, D., and Cohen-Or, D. Stable flow: Vital layers for training-free image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Banerjee, S. and Lavie, A. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, USA, June 2005. Association for Computational Linguistics.

Bose, A. J., Akhound-Sadegh, T., Huguet, G., Fatras, K., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M. M., and Tong, A. Se(3)-stochastic flow matching for protein backbone generation. In *International Conference on Learning Representations*, 2024.

Bu, J., Ling, P., Zhou, Y., Zhang, P., Dong, X., Zang, Y., Cao, Y., Wu, T., Lin, D., and Wang, J. Hiflow: Training-free high-resolution image generation with flow-aligned guidance. *Advances in Neural Information Processing Systems*, 2025.

Chan, D., Petryk, S., Gonzalez, J., Darrell, T., and Canny, J. Clair: Evaluating image captions with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13638–13646, 2023.

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

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., and Rombach, R. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T. Q., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow matching. 2024.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. CLIPscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Jin, C. et al. A-flops: Accelerating diffusion sampling with adaptive flow path sampler. *arXiv preprint arXiv:2509.00036*, 2025.

Kornilov, N., Mokrov, P., Gasnikov, A., and Korotin, A. Optimal flow matching: Learning straight trajectories in just one step. 2024.

Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., and Michaeli, T. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19721–19730, 2025.

Lee, S., Lin, Z., and Fanti, G. Improving the training of rectified flows. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63082–63109, 2024.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Li, T. et al. Self-guidance: Boosting flow and diffusion generation on their own. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., and Ouyang, W. Flow-grpo: Training flow matching models via online rl. *Advances in Neural Information Processing Systems*, 2025.

Liu, X., Gong, C., and Liu, Q. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.03003*, 2022.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Liu, X., Zhang, X., Ma, J., Peng, J., and Liu, Q. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Luo, Y., Du, D., Huang, H., Fang, Y., and Wang, M. Curveflow: Curvature-guided flow matching for image generation. *arXiv preprint arXiv:2508.15093*, 2025.

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

Papineni, K., Roukos, S., Ward, T., and Zhu, W. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

Patel, M., Wen, S., Metaxas, D. N., and Yang, Y. Flowchef: Steering of rectified flow models for controlled generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15308–15318, 2025.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Rout, L., Chen, Y., Ruiz, N., Caramanis, C., Shakkottai, S., and Chu, W.-S. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

Wang, F.-Y., Huang, Z., Bergman, A. W., Shen, D., Gao, P., Lingelbach, M., Sun, K., Bian, W., Song, G., Liu, Y., and Li, H. Straightness is not your need in rectified flow. In *International Conference on Learning Representations*, 2024a.

Wang, F.-Y., Yang, L., Huang, Z., Wang, M., and Li, H. Rectified diffusion: Straightness is not your need in rectified flow. *arXiv preprint arXiv:2410.07303*, 2024b.

Wang, J., Pu, J., Qi, Z., Guo, J., Ma, Y., Huang, N., Chen, Y., Li, X., and Shan, Y. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024c.

Wang, J., Pu, J., Qi, Z., Guo, J., Ma, Y., Huang, N., Chen, Y., Li, X., and Shan, Y. Taming rectified flow for inversion and editing. In *International Conference on Machine Learning*, 2025a.

Wang, L. et al. Training free optimal control flow (ocflow). In *International Conference on Learning Representations*, 2025b.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. 2010.

Yoon, S.-H., Li, M., Beaudouin, G., Wen, C., Azhar, M. R., and Wang, M. Splitflow: Flow decomposition for inversion-free text-to-image editing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A. Sampling with Momentum

For each sample trajectory, we maintain a first-moment vector m_k initialized to zero. At each step, we update this momentum using an exponential moving average:

$$g_k = -\, v_\Theta(z_k, t_k, c), \tag{13}$$

$$m_{k+1} = \beta_1\, m_k + (1-\beta_1)\, g_k, \tag{14}$$

$$z_{k+1} = z_k + \eta_k\, m_{k+1}, \tag{15}$$

where $\beta_1 \in [0, 1)$ controls the momentum strength (e.g., 0.8 by default). This formulation introduces temporal smoothness in velocity directions and stabilizes sampling for stiff or high-curvature trajectories.

This training-free method maintains momentum independently for each trajectory without batch sharing, introducing negligible computational overhead by requiring only one additional vector state per sample. Notably, in the specific case where $\beta_1 = 0$, the algorithm reduces exactly to the vanilla Euler solver. Empirically, we observe that $\beta_1 = 0.8$ achieves the optimal performance on Flickr30K.

B. Pseudo Algorithm

Algorithm 3 Sampling with Momentum

Require: Trained velocity field v_Θ, condition c, time grid $1 = t_0 > \cdots > t_K = 0$, step sizes η_k, momentum coefficient β_1

Sample $z_0 \sim p_1$ {initialize from noise prior}
$m_0 \leftarrow 0$
for $k = 0, 1, \ldots, K-1$ **do**
  $g_k \leftarrow -v_\Theta(z_k, t_k, c)$
  $m_{k+1} \leftarrow \beta_1 m_k + (1-\beta_1) g_k$
  $z_{k+1} \leftarrow z_k + \eta_k m_{k+1}$
end for
return z_K {final generated sample}

Fundamentally, this approach operates as a form of trajectory smoothing. The method acts as a temporal low-pass filter on the flow velocity, suppressing abrupt direction changes (high-frequency oscillations) while preserving the global flow geometry. By aggregating historical directional information, the momentum term effectively straightens the integration path, making it robust to local irregularities or noise in the learned velocity field. This avoids the need to assume diminishing gradients and ensures the solver smoothly reduces to Euler when $\beta_1 \to 0$, providing convergence stability with minimal overhead.

C. Complexity Analysis

We now analyze and compare the computational complexity of three flow sampling schemes, i.e., conventional, Look-Ahead, and Look-Back, in terms of their dominant operations per timestep. Let C_v denote the cost of one model (velocity field) evaluation $v_\Theta(z, t, c)$, and K the total number of sampling steps.

Conventional Flow Sampling Each step directly evaluates the velocity field once and performs a single latent update Eq. (5). Hence, the total cost scales linearly with K:

$$\mathcal{C}_{\text{flow}} = K\, C_v.$$

Look-Ahead Sampling The proposed Look-Ahead scheme estimates the peek velocity $\tilde{v}$ via a finite-difference extrapolation Eq. (6) without invoking an additional model call. Thus, its computational complexity remains identical to the conventional solver, but with extra $\mathcal{O}(d)$ vector arithmetic per step (where d is latent dimensionality), which is negligible compared to C_v.

$$\mathcal{C}_{\text{LA}} = K\, C_v + \mathcal{O}(K\, d).$$

Table 4. Key hyperparameters related to the proposed look-ahead scheme.

Dataset	γ	τ_{curv}
COCO-2017	0.9	10
CUB-200	0.95	1
Flickr30K	0.9	1

Table 5. key hyperparameters related to the proposed look-back scheme.

Dataset	λ	ξ_*
COCO-2017	0.1	0
CUB-200	0.1	0
Flickr30K	0.1	0.25

In practice, the runtime overhead is marginal ($< 3\%$), as the dominant cost remains model evaluation.

Look-Back Sampling The Look-Back scheme introduces exponentially weighted averaging and blending operations Eq. (9)–Eq. (10), both linear in d. Like Look-Ahead, it performs one velocity-field query per step; thus its asymptotic complexity also matches the baseline:

$$\mathcal{C}_{\text{LB}} = K\,C_v + \mathcal{O}(K\,d).$$

However, its memory footprint is slightly higher due to storage of the running average $\bar{z}_k$.

Table 3 summarizes the comparative complexity, model calls, and memory cost. Both Look-Ahead and Look-Back retain the same asymptotic complexity as conventional flow sampling, offering stability or smoothness improvements at nearly constant computational cost.

Table 3. Complexity comparison of different flow sampling schemes. C_v denotes a single model evaluation cost; d is latent dimension.

Method	Model Calls / Step	Time Complexity	Extra Memory
Conventional Flow Sampling	1	$\mathcal{O}(K\,C_v)$	$\mathcal{O}(d)$
Look-Ahead (Curvature-Gated)	1	$\mathcal{O}(K\,C_v + K\,d)$	$\mathcal{O}(d)$
Look-Back (EMA-Stabilized)	1	$\mathcal{O}(K\,C_v + K\,d)$	$\mathcal{O}(2d)$

The additional $\mathcal{O}(K\,d)$ arithmetic in Look-Ahead and Look-Back schemes is computationally negligible in practice, since d corresponds to the latent dimensionality of the VAE or diffusion backbone rather than the pixel space. Typical latent sizes (e.g., $d \approx 4 \times 64 \times 64$ for Stable Diffusion) are orders of magnitude smaller than the full image dimensionality. Consequently, the total overhead $K\,d$ remains well within modern GPU memory and compute budgets even for large K (e.g., $K \leq 100$). This ensures that both Look-Ahead and Look-Back sampling remain fully feasible and efficient in high-resolution generative pipelines, maintaining near-identical runtime to the baseline solver while providing enhanced stability and smoothness.

D. Implementation Details

CLAIR (Chan et al., 2023) leverages a Large Language Model to evaluate semantic equivalence. For the CLAIR implementation, we utilize the OpenAI gpt-4.1-mini model as the evaluator backend.

To ensure reproducibility, we detail the specific hyperparameter configurations used for the Look-Ahead and Look-Back schemes across the evaluated datasets (COCO17, CUB-200, and Flickr30K). For the Look-Ahead scheme, the curvature threshold τ_{curv} and the interpolation factor γ are tuned to balance trajectory fidelity with sampling efficiency; these settings are provided in Table 4. For the Look-Back scheme, we specify the blending weight λ and the SNR midpoint ξ_*, which jointly control the intensity of the history-based stabilization and the decay scheduling. These values are listed in Table 5. Regarding the dataset protocols, since both COCO17 and Flickr30K provide five captions per image, we use the specific prompt that achieves the best CLIPScore for evaluation purposes.