CORRECTING TRAJECTORY DRIFT IN FLOW MATCHING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-matching generative models are typically sampled by numerically integrating a learned velocity field using a small number of deterministic steps. In this regime, fixed-timestep solvers can drive inference trajectories away from the continuous learned trajectory, causing the learned velocity field to be evaluated at increasingly shifted states and potentially compounding error. We study this *inference trajectory drift* and propose a training-free sampler that combines cached finite-difference acceleration with discrepancy-guided online control. At each step, the method forms an Euler update and a finite-difference acceleration update from consecutive velocity evaluations, and conservatively shrinks the timestep when their magnitude and angular disagreement is large. The sampler uses one network evaluation per step, stores only the previous velocity, and requires no retraining. A scalar gain λ modulates the acceleration correction, while discrepancy-guided step control provides complementary online stabilization; their relative contributions depend on the model and sampling regime. The strongest gains occur in low-NFE RectFlow sampling, where the method outperforms Euler, standard multistep baselines, flow-prediction DPM-Solver++ 2M, and Flow-UniPC across the tested NFE range. Full-vector trajectory diagnostics further show reduced state deviation and learned-field discrepancy relative to AB2 at matched NFE. These findings highlight that sampler design for learned generative models should account not only for nominal integration order, but also for the behavior of the realized inference trajectory.

1 INTRODUCTION

Flow matching generative models generate samples by integrating learned continuous-time dynamics that transport simple base distributions to complex data distributions Lipman et al. (2023), with connections to probability-flow formulations of diffusion models Song et al. (2021) and continuous normalizing flows Chen et al. (2018). In practice, inference uses a finite number of numerical steps for computational efficiency, motivating specialized fast samplers designed to reduce discretization error or preserve sample quality at low numbers of function evaluations Lu et al. (2022); Zhang & Chen (2023).

However, when applied to learned velocity fields, discretization introduces a systematic mismatch between training and inference. During training, the model is optimized to produce accurate velocities along trajectories drawn from the underlying continuous flow. During inference, discrete updates produce states that deviate from these trajectories. As a result, the learned velocity field is repeatedly evaluated at shifted states along the realized numerical trajectory, which can contribute to error accumulation in the regimes studied. We refer to this phenomenon as *inference trajectory drift*.

Recent approaches improve sampling efficiency via higher-order solvers Lu et al. (2022); Zhang & Chen (2023), consistency models Song et al. (2023), or distillation-based acceleration Salimans & Ho (2022). However, these methods primarily target reducing the number of function evaluations or learning fast samplers, rather than addressing the mismatch between training trajectories and discretized inference dynamics. As we argue theoretically and show empirically, nominal truncation order alone does not fully predict empirical behavior in fixed-NFE learned samplers, because the learned velocity field is queried on the realized numerical trajectory rather than along the intended continuous flow.

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.

Training–inference mismatch. Let ϕ_t^θ denote the continuous flow map induced by the learned velocity field and $\tilde{\phi}_{t_i}$ the corresponding discrete inference map. While training targets accuracy along a prescribed path with marginal p_t^*, inference produces samples from $\tilde{p}_{t_i} = (\tilde{\phi}_{t_i})_{\#}p_0$, which can deviate from the continuous learned marginal $p_t^\theta = (\phi_t^\theta)_{\#}p_0$. This decomposition distinguishes deviation of the discrete inference distribution from the continuous learned flow from any pre-existing mismatch between the learned and target flows.

In learned generative dynamics, model approximation and numerical discretization introduce distinct sources of sampling error. Recent work has studied error propagation in diffusion models Li & van der Schaar (2024) and Euler–Maruyama discretization error Choi & Fan (2025). However, the interaction between numerical deviation and evaluations along the realized inference trajectory in deterministic flow-matching samplers remains insufficiently understood.

Our approach. We address trajectory drift through a simple, learning-aware modification of the inference procedure. Our method operates entirely at inference time, requiring no retraining or architectural changes, and can be applied as a drop-in replacement for standard samplers. At each step, it forms a first-order (Euler) candidate and a finite-difference acceleration-corrected candidate from consecutive velocity evaluations, measures their magnitude and angular disagreement, and conservatively shrinks the timestep in high-disagreement regions (Figure 1). The sampler combines two mechanisms: a cached finite-difference acceleration correction and discrepancy-guided step control. Their relative contributions are regime-dependent; on RectFlow, the acceleration correction accounts for most of the observed gain, while adaptive control provides an additional improvement when combined with it.

Contributions.

- We formalize *inference trajectory drift* as a training–inference mismatch in flow matching sampling, separating continuous-model error, numerical drift D_i^θ, and off-trajectory residual error.
- We propose a lightweight, training-free sampler that uses one network evaluation per step and combines a cached finite-difference acceleration correction with discrepancy-guided online control.
- We show strong matched-NFE results on RectFlow CIFAR-10 against classical integrators and modern fast solvers (flow-prediction DPM-Solver++ 2M and Flow-UniPC), with the lowest FID at every tested budget from 4 to 64 NFE.
- We provide quantitative full-vector evidence that the complete method tracks the continuous learned ODE more closely than Euler and standard AB2 at matched NFE, together with a component ablation clarifying their complementary roles.

Scope. We focus on deterministic probability-flow sampling, with emphasis on flow-matching and rectified-flow models Lipman et al. (2023). Stochastic SDE-based samplers Song et al. (2021) are outside the scope of this work. Our goal is to develop learning-aware multistep inference schemes that account for trajectory deviation induced by discretized sampling in learned generative dynamics.

2 Related Work

Diffusion and Flow Models. Diffusion models Ho et al. (2020); Nichol & Dhariwal (2021) and score-based methods Song et al. (2021) define generative processes via stochastic dynamics, which admit deterministic probability-flow ODE counterparts. Continuous normalizing flows Chen et al. (2018); Grathwohl et al. (2019) provide a related framework for modeling continuous transformations via neural ODEs. Flow matching Lipman et al. (2023) directly learns velocity fields defining deterministic trajectories, with rectified flow variants improving training efficiency and trajectory geometry Liu et al. (2023). Transformer backbones have become prominent across diffusion and continuous-time generative modeling, including Diffusion Transformers Peebles & Xie (2023), Scalable Interpolant Transformers Ma et al. (2024), and large-scale rectified-flow text-to-image models Esser et al. (2024).

Numerical Solvers for Sampling. Sampling requires numerical integration of learned dynamics. Methods such as PNDM Liu et al. (2022), DPM-Solver Lu et al. (2022), DEIS Zhang & Chen (2023),

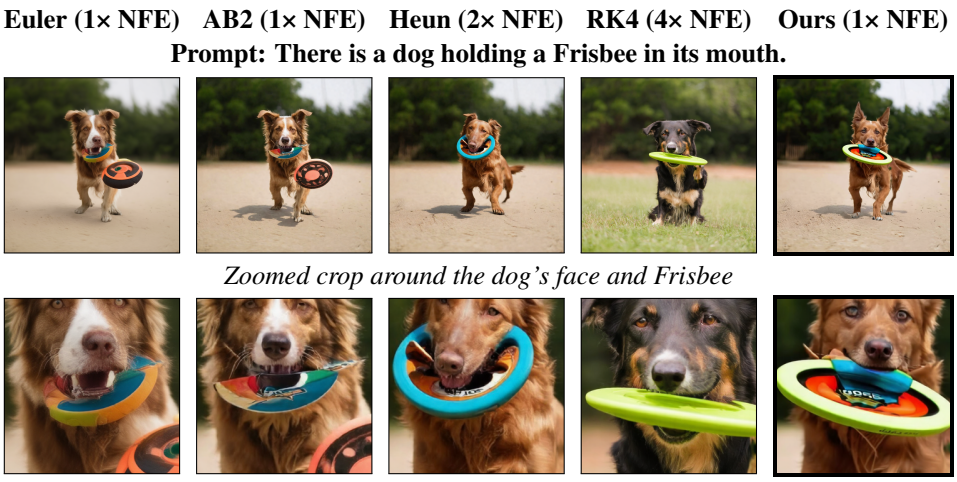

Figure 1: Visual comparison of samplers at matched initial noise and step count (20 steps). Euler, AB2, and our method use 1 network evaluation per step (1× NFE); Heun uses 2× NFE and RK4 uses 4× NFE. All images share the same initial noise seed; only the sampler varies. The second row zooms into the dog's face and Frisbee to make local structural differences easier to inspect. Additional comparisons appear in Appendix K.1 (Figure 4).

and EDM Karras et al. (2022) improve sampling efficiency through specialized numerical integration and discretization schemes tailored to diffusion dynamics. Classical numerical methods, including Runge–Kutta and linear multistep methods Hairer et al. (1993), are also commonly applied.

Orthogonal approaches reduce sampling cost via model-side modifications, including progressive distillation Salimans & Ho (2022) and consistency models Song et al. (2023), which learn fast approximate samplers.

Related approaches optimize sampling schedules or numerical integration strategies, including Align Your Steps (AYS) Sabour et al. (2024), BOSS Nguyen et al. (2024), and SDM Jo & Choi (2026). AYS optimizes solver-specific sampling schedules, BOSS uses Bellman-optimal stepsizes together with model straightening, and SDM combines adaptive solver selection with Wasserstein-bounded timesteps. In contrast, our method performs sample-dependent online control along the realized trajectory while retaining one network evaluation per step and requiring no retraining.

Training–Inference Mismatch and Error Propagation. Recent work has studied several related sources of inference-time error in diffusion models. Li and van der Schaar Li & van der Schaar (2024) analyze how prediction errors propagate through sequential diffusion sampling. Choi and Fan Choi & Fan (2025) analyze Euler–Maruyama discretization error, while Ning et al. Ning et al. (2024) study exposure bias as a mismatch between training and sampling inputs. These works establish complementary perspectives on sampling error. Our work focuses specifically on sampler-controllable deviation from the continuous learned flow in deterministic flow-matching inference.

3 Method

3.1 Problem Setup

We consider flow-matching and related deterministic transport models with a learned velocity field Lipman et al. (2023); Liu et al. (2023):

$$\frac{dx}{dt} = v_\theta(x,t), \quad t \in [0,1], \quad x(0) \sim p_0, \tag{1}$$

where v_θ is trained to approximate a target velocity field v^* along a prescribed training path with marginal p_t^*.

Let ϕ_t^θ denote the continuous flow map induced by the learned field v_θ, with corresponding learned marginal

$$p_t^\theta = (\phi_t^\theta)_{\#} p_0. \tag{2}$$

At inference time, a discrete solver produces states x_i and an induced numerical marginal $\tilde{p}_i$.

Training–inference mismatch. The learned field is trained using states from the prescribed training distribution, whereas numerical inference evaluates v_θ along the solver-induced distribution $\tilde{p}_i$. This distinction is closely related to exposure bias in diffusion models, where a mismatch between training and sampling inputs can induce prediction error Ning et al. (2024); such local errors may then propagate through sequential sampling Li & van der Schaar (2024). We refer to the corresponding sampler-induced phenomenon considered here as *inference trajectory drift*.

3.2 Inference Trajectory Drift

Three-trajectory decomposition. To precisely locate the source of inference error, we distinguish three trajectories. Let $x^*(t)$ denote the *target trajectory* under the true velocity field v^*, with marginal p_t^*. Let $x^\theta(t)$ denote the *continuous learned trajectory* under v_θ, with marginal p_t^θ. Let x_i denote the *discrete numerical trajectory* produced by a solver applied to v_θ, with induced marginal $\tilde{p}_i$.

Define:

$$D_i^\theta = W_2(\tilde{p}_i,\, p_{t_i}^\theta), \quad D_i^* = W_2(\tilde{p}_i,\, p_{t_i}^*). \tag{3}$$

By the triangle inequality:

$$W_2(\tilde{p}_i, p_{t_i}^*) \leq \underbrace{W_2(\tilde{p}_i, p_{t_i}^\theta)}_{\text{numerical/inference drift}} + \underbrace{W_2(p_{t_i}^\theta, p_{t_i}^*)}_{\text{continuous model error}}. \tag{4}$$

The first term, D_i^θ, measures how far the discrete inference trajectory has drifted from the continuous learned ODE—this is what numerical solvers can control. The second term is fixed by the quality of v_θ and cannot be reduced at inference time. We focus exclusively on controlling D_i^θ.

Off-trajectory residual. Training controls velocity error primarily under the intended marginal p_t^θ (or p_t^*), not under the realized inference distribution $\tilde{p}_i$. We define the *off-trajectory residual*:

$$\rho_i = \left(\mathbb{E}_{x\sim\tilde{p}_i} \left\|v_\theta(x,t_i) - v^*(x,t_i)\right\|^2\right)^{1/2}. \tag{5}$$

When the solver-induced distribution $\tilde{p}_i$ differs from the training marginal, v_θ is evaluated under a distribution shift relative to training. Consequently, its approximation error under $\tilde{p}_i$ need not be controlled by its training-time error.

Scope of the result. The following result characterizes how discretization error grows under the learned ODE v_θ and how it relates to the overall sampling error via a triangle inequality decomposition. The analysis bounds the growth of numerical drift to theoretically motivate controlling sampler-induced deviation. The theoretical bound does not derive the exact discrepancy controller or establish its optimality; it motivates the design while the empirical sections establish the effectiveness of the complete practical sampler.

Assumptions. Assume v_θ is L-Lipschitz in x, uniformly in t, and that the local truncation error of the Euler step applied to $\dot{x} = v_\theta(x,t)$ is bounded by Ch_i^2.

Theorem 1 (Numerical drift to the learned flow)**.** *Let $x^\theta(t)$ denote the continuous solution of $dx/dt = v_\theta(x,t)$, and let x_i be produced by Euler with step sizes h_i, starting from the same initial condition. Assume v_θ is L-Lipschitz in x and the one-step local defect is bounded by Ch_i^2. Define $e_i := \|x^\theta(t_i) - x_i\|$. Then:*

$$e_{i+1} \leq (1 + h_i L)\, e_i + Ch_i^2. \tag{6}$$

Unrolling via the discrete Grönwall inequality, for uniform step size h over a bounded interval $[0,T]$:

$$W_2(\tilde{p}_i,\, p_{t_i}^\theta) \leq C\, h\, e^{Lt_i}, \tag{7}$$

which is $\mathcal{O}(h)$ for fixed T. By the triangle inequality,

$$W_2(\tilde{p}_i,\, p^*_{t_i}) \;\le\; \underbrace{W_2(\tilde{p}_i,\, p^{\theta}_{t_i})}_{\textit{numerical drift (sampler-controllable)}} \;+\; \underbrace{W_2(p^{\theta}_{t_i},\, p^*_{t_i})}_{\textit{model error (inference-fixed)}} . \tag{8}$$

The first term is numerical/inference drift and can be influenced by the sampler design. The second term reflects the quality of v_θ as an approximation to v^* and cannot be reduced at inference time. This decomposition motivates limiting local update size to control sampler-induced deviation (proof in Appendix B).

Design motivation. This analysis suggests controlling trajectory deviation by limiting local step size. Motivated by this principle, we use disagreement between candidate updates as a low-cost empirical indicator of local proposal inconsistency along the realized numerical trajectory, and reduce the step size when disagreement is large.

3.3 Candidate Updates

Let $\Delta t_i = t_{i+1} - t_i$ and $v_i := v_\theta(x_i, t_i)$.

First-order candidate. The Euler candidate is:

$$s_i^{(1)} = \Delta t_i\, v_i. \tag{9}$$

Finite-difference acceleration estimate. Using consecutive velocity evaluations, we estimate the local temporal acceleration along the inference trajectory:

$$a_i \approx \frac{v_i - v_{i-1}}{\Delta t_{i-1}}. \tag{10}$$

This is a finite-difference estimate along the *realized inference trajectory*, not an exact acceleration; it captures both local curvature and drift-induced inconsistency between successive evaluations.

Acceleration-corrected candidate.

$$s_i^{(2)} = \Delta t_i\, v_i + \lambda \cdot \frac{1}{2}\frac{(\Delta t_i)^2}{\Delta t_{i-1}}(v_i - v_{i-1}), \tag{11}$$

where $\lambda > 0$ is a scalar gain (default $\lambda = 2.0$, selected by ablation; details in Appendix H). Under smooth dynamics and $\lambda = 1$, this recovers a familiar two-step finite-difference correction.[1]

Connection to multistep methods. The acceleration term is related in form to explicit multistep finite-difference corrections. For $\lambda = 1$, no adaptation, and a uniform schedule, the finite-difference correction has the familiar AB2-like form. The complete proposed sampler uses gain modulation and discrepancy-guided control and is not claimed to have a fixed classical convergence order. Evaluations along the inference trajectory $\{x_i\}$ differ from evaluations along the continuous learned trajectory $x^\theta(t)$, so the finite-difference estimate a_i incorporates both temporal variation and drift-induced perturbations.

3.4 Discrepancy-Guided Adaptive Timestep (AdaptDT)

A second mechanism is a conservative adaptive timestep rule that shrinks the step when the first-order and acceleration-corrected candidates disagree.

Magnitude disagreement.

$$e_{\mathrm{mag},i} = \frac{\|s_i^{(2)} - s_i^{(1)}\|}{\|s_i^{(1)}\| + \eta}, \tag{12}$$

where $\eta > 0$ is an empirical stabilization constant. This detects when the acceleration correction is large relative to the Euler step.

[1] When $\lambda = 1$ and $\alpha_i = 1$ under a fixed uniform schedule, the acceleration term reduces to a familiar two-step explicit multistep form. Our contribution is the complete sampler combining this correction with discrepancy-guided control, not the limiting AB2-like case alone.

Angular disagreement.

$$e_{\mathrm{ang},i} = \frac{1}{\pi} \arccos\left(\frac{\langle s_i^{(1)},\, s_i^{(2)} \rangle}{(\|s_i^{(1)}\| + \eta)(\|s_i^{(2)}\| + \eta)} \right), \tag{13}$$

where the cosine argument is clamped to $[-1, 1]$. This detects directional instability between the two candidates, even when their norms are similar.

Total disagreement and shrinkage.

$$e_i = e_{\mathrm{mag},i} + e_{\mathrm{ang},i}, \qquad \alpha_i = \max\left(0.5,\ \frac{1}{1 + e_i}\right). \tag{14}$$

The adaptive timestep is $\Delta t_i^{\mathrm{adap}} = \alpha_i \Delta t_i$. Since $\alpha_i \in [0.5, 1]$, the method is conservative: it never expands beyond the base schedule and only shrinks the step when the two candidates disagree. Candidate disagreement is a low-cost empirical indicator of local proposal inconsistency along the realized numerical trajectory, not a direct estimator of off-trajectory model error or a formal local-error estimator in the sense of Runge–Kutta methods.

Final update.

$$x_{i+1} = x_i + \alpha_i \Delta t_i\, v_i + \lambda \cdot \frac{1}{2} \frac{(\alpha_i \Delta t_i)^2}{\Delta t_{i-1}} (v_i - v_{i-1}), \tag{15}$$

with time update $t_{i+1} = t_i + \alpha_i (t_{i+1}^{\mathrm{base}} - t_i)$.

The complete procedure is given in Algorithm 1 (Appendix A).

Interpretation. Magnitude disagreement detects when the acceleration correction is large; angular disagreement detects directional instability even when norms are similar. Together they provide a two-dimensional indicator of local proposal inconsistency. In regions where successive candidates agree, e_i is small and $\alpha_i \approx 1$ (near-full step). In regions of high curvature or inconsistent proposals, e_i is large and α_i shrinks toward 0.5, limiting trajectory deviation. This is designed to reduce one pathway of the drift amplification described in Theorem 1.

3.5 Discussion

Unlike classical higher-order solvers, which primarily improve nominal truncation order, our method combines a cached finite-difference acceleration correction with discrepancy-guided step control. Their relative contributions are regime-dependent; on RectFlow, the acceleration correction accounts for most of the observed gain, while adaptive control provides an additional improvement when combined with it (Appendix H.1). The gain $\lambda = 2.0$ further adjusts the precision–recall trade-off in SiT settings (Appendix H). The method requires one network evaluation per step, stores only v_{i-1}, and is complementary to diffusion-specific samplers Lu et al. (2022); Karras et al. (2022).

4 Experiments

We evaluate the proposed inference-time sampler across multiple flow matching generative models, with primary emphasis on low-NFE RectFlow settings where the method's gains are strongest. Secondary evaluations on Stable Diffusion 3.5 and SiT provide cross-model context. All experiments are strictly *inference-only*: the underlying models are never retrained or modified.

4.1 Experimental Protocol

Models and datasets. We evaluate RectFlow models on CelebA-HQ and CIFAR-10, Stable Diffusion 3.5 (SD 3.5) from the rectified-flow transformer family described by Esser et al. Esser et al. (2024) on the COCO validation set Lin et al. (2014), and SiT models Ma et al. (2024) trained on ImageNet Deng et al. (2009).

Metrics. We report Fréchet Inception Distance (FID) Heusel et al. (2017), Kernel Inception Distance (KID) Bińkowski et al. (2018), and PRDC metrics (precision, recall, density, coverage) Naeem et al. (2020). FID and KID measure global distributional similarity, while PRDC metrics characterize fidelity and diversity. In Section 4.4, we also discuss *reference FID*—FID computed against a dense 500-step Euler approximation of the continuous learned ODE rather than real data; this measures how closely a low-NFE sampler reproduces the learned continuous-time dynamics, not real-data FID. Full SD 3.5 reference-FID numbers appear in Appendix D.

NFE accounting. We distinguish step-matched from compute-matched comparisons. At N steps: Euler uses N NFE, Ours uses N NFE, Heun uses $2N$ NFE, and RK4 uses $4N$ NFE. Matched-NFE comparisons use different step counts across solvers to equalize total compute. Direct model-call auditing confirmed that actual network evaluations equal the requested NFE for the matched-compute RectFlow comparisons.

Solvers. We compare classical ODE integrators—Euler (1 NFE/step), AB2/AB4 (1 NFE/step) Hairer et al. (1993), Heun (2 NFE/step), RK4 (4 NFE/step), and DOPRI5 where included—and, on RectFlow CIFAR-10, two modern fast solvers adapted to flow prediction: Flow-DPM-Solver++ 2M Lu et al. (2025) and Flow-UniPC Zhao et al. (2023). These are flow-prediction versions of DPM-Solver++ 2M and UniPC, appropriate because RectFlow exposes a flow velocity rather than the original diffusion noise/score parameterization; we do not apply the original diffusion parameterizations unchanged. AYS, BOSS, and SDM are discussed in Section 2 as complementary schedule- or solver-design approaches rather than direct inference-only baselines: they involve, respectively, offline schedule optimization, model straightening, or additional adaptive solver-selection machinery.

Implementation details. Our method uses $\lambda = 2.0$ by default with AdaptDT enabled. All solvers share identical noise schedules, initialization, and conditioning. Deterministic per-image seeds enable paired comparisons across solvers (full details in Appendix C).

4.2 RectFlow Results

The strongest empirical evidence for the proposed method is on RectFlow, especially CIFAR-10. Table 1 reports FID across a range of NFE budgets on CelebA-HQ and CIFAR-10.

On both datasets, our method substantially improves over Euler at low NFE and outperforms the other 1-evaluation-per-step baselines (AB2, AB4) at all but one budget. At 64 NFE on CelebA-HQ, AB2 achieves a slightly lower FID (10.36 vs. 10.67); the differences at higher budgets are small. On CIFAR-10, we additionally compare against flow-prediction DPM-Solver++ 2M and Flow-UniPC at matched NFE. The proposed method achieves the lowest FID at every tested NFE from 4 through 64, outperforming Euler, AB2, AB4, Heun, RK4, and both modern fast solvers. Direct model-call auditing confirms that the actual number of network evaluations equals the requested NFE for the compared RectFlow samplers.

A component ablation on RectFlow CIFAR-10 (Appendix H.1) shows that the finite-difference acceleration correction accounts for most of the improvement over Euler, while discrepancy-guided adaptive control provides an additional benefit when combined with it; the full method achieves the best FID at every tested budget.

Importantly, the RectFlow gains are not explained by a shortened effective integration horizon. The adaptive controller terminates essentially at the target endpoint in these experiments, while the fixed-acceleration variant, which covers the complete integration interval, retains most of the large low-NFE improvement over Euler (Appendix H.1). Thus, although endpoint handling can affect adaptive sampling in some regimes, it does not account for the principal RectFlow gains reported here.

4.3 Quantitative Trajectory Diagnostics

To strengthen the trajectory-drift evidence beyond norm-based plots, we measure full-vector deviation from a dense numerical approximation of the continuous learned ODE on RectFlow. Using 256 paired samples, we compare Euler, standard AB2, and the full proposed method against a dense learned-ODE reference. At each physical time, we measure (i) full-state deviation from the reference

Table 1: **RectFlow results on CelebA-HQ and CIFAR-10.** FID at different NFE budgets. Lower is better. Best values within each NFE column are **bold**. On CIFAR-10, Flow-DPM-Solver++ 2M and Flow-UniPC are flow-prediction adaptations of the corresponding modern solvers.

Experiment 1: RectFlow CelebA-HQ					
Method	4 NFE	8 NFE	16 NFE	32 NFE	64 NFE
Euler	134.50	82.43	42.55	20.92	12.78
Heun	151.20	91.44	49.01	23.55	12.26
RK4	262.19	96.37	51.21	25.73	13.46
AB2	103.92	56.91	27.96	13.86	**10.36**
AB4	113.51	54.08	27.53	14.52	11.00
Ours	**74.44**	**44.40**	**22.85**	**13.12**	10.67
Experiment 2: RectFlow CIFAR-10					
Method	4 NFE	8 NFE	16 NFE	32 NFE	64 NFE
Euler	55.97	21.17	12.55	9.54	8.16
Heun	104.99	23.02	12.14	9.70	8.19
RK4	378.49	42.05	18.08	12.92	10.09
AB2	32.84	12.63	10.23	8.16	7.07
AB4	29.18	13.26	9.95	8.10	7.30
Flow-DPM-Solver++ 2M	46.69	15.28	10.03	8.23	7.34
Flow-UniPC	40.16	13.44	10.06	8.34	7.35
Ours	**24.43**	**12.04**	**8.91**	**7.46**	**7.00**

Table 2: **Quantitative trajectory diagnostics on RectFlow (256 paired samples).** State-deviation AUC and learned-field discrepancy AUC relative to a dense numerical approximation of the continuous learned ODE. Lower is better. All paired-bootstrap 95% intervals for Full-minus-AB2 comparisons exclude zero.

Metric	Method	4 NFE	8 NFE
State-deviation AUC	Euler	0.1498	0.1033
	AB2	0.1335	0.0732
	Full method	**0.1174**	**0.0509**
Learned-field discrepancy AUC	Euler	0.3236	0.2268
	AB2	0.2973	0.1731
	Full method	**0.2731**	**0.1300**

and (ii) learned-field discrepancy

$$\frac{\|v_\theta(x_i,t_i) - v_\theta(x^\theta_{\mathrm{ref}}(t_i),t_i)\|_2}{\|v_\theta(x^\theta_{\mathrm{ref}}(t_i),t_i)\|_2 + \epsilon}. \tag{16}$$

We report the area under the curve (AUC) of each metric over the inference horizon (Table 2).

Standard AB2 provides the expected improvement over Euler, but the proposed method consistently reduces both full-state deviation and learned-field discrepancy beyond AB2 at matched NFE (Full-minus-AB2: -0.0161 and -0.0223 for state deviation at 4/8 NFE; -0.0242 and -0.0432 for learned-field discrepancy). All corresponding paired-bootstrap 95% intervals exclude zero. This shows that the reduction in trajectory deviation is not reproduced by standard AB2 alone. The diagnostic measures sampler-controllable deviation from the continuous learned ODE (D_i^θ); it does not measure the unknown target-field residual ρ, because the true target velocity v^* is unavailable. Thus the proposed method reduces sampler-controllable deviation from the learned continuous-time trajectory and reduces the perturbation of learned-field evaluations caused by coarse numerical displacement.

4.4 Reference-Trajectory Diagnostics

On SD 3.5, the method attains the best reference FID, KID, precision, density, and coverage among 1-NFE-per-step samplers and improves over Euler on all six reference metrics; full results, in-

cluding the rationale for dense learned-ODE references and matched-NFE comparisons against Heun/RK4/DOPRI5, are provided in Appendix D. Perceptual and semantic metrics (LPIPS, DISTS, CLIP, IS) appear in Appendix K.2.

4.5 SiT Results

As secondary cross-model evidence, we evaluate on SiT/ImageNet at fixed step budgets of 20 and 30 steps. Table 4 (Appendix E) reports FID, classification accuracy (Top-1/Top-5), and PRDC metrics. At 30 steps the proposed method attains the lowest FID and the highest Top-1 accuracy and recall of all compared solvers, including Heun and RK4 at 2–4$\times$ the compute. At 20 steps it shows the same fidelity-oriented profile as Heun and RK4—higher Top-1/Top-5 accuracy, precision, density, and coverage than Euler at lower recall—while Euler retains the lowest FID (25.09 vs. 26.16). Qualitative comparisons at low step counts appear in Appendix E (Figure 2); additional comparisons appear in Appendix J.1.

4.6 Performance at Matched Compute

Matched-NFE comparisons on SD 3.5 are reported in Appendix D (Table 3). At 20 NFE, our method improves all six reference metrics over Euler and outperforms Heun, RK4, and DOPRI5 while using only one evaluation per step; higher-order solvers that reduce step count to match compute fall into a more aggressive low-step regime.

4.7 Additional Analyses

The appendix provides additional analyses, including the RectFlow component ablation (Appendix H.1), SD 3.5 reference FID (Appendix D), SiT results and hyperparameter studies (Appendices E and H), step-count sweeps (Appendix F), trajectory analyses (Appendix I), and hyperparameter sensitivity (Appendix G).

5 Limitations

Our method operates purely at inference time and cannot correct errors from model misspecification or systematic biases in the learned velocity field; it targets flow matching inference only. The finite-difference correction relies on local temporal consistency and may become unstable in highly irregular regimes, though AdaptDT mitigates this. The AdaptDT rule is a heuristic controller, not a Runge–Kutta-style embedded error estimator. Empirical evaluation focuses on image generation (SD 3.5, RectFlow, SiT); other modalities may behave differently.

6 Conclusion

We studied inference trajectory drift in flow matching samplers, where fixed discrete updates can move samples away from the continuous learned trajectory and thereby perturb subsequent learned-field evaluations. We proposed a training-free sampler that combines cached finite-difference acceleration with discrepancy-guided online control. The method uses one network evaluation per step and stores only the previous velocity.

The strongest improvements occur in low-NFE RectFlow settings, where the finite-difference correction substantially reduces numerical trajectory deviation and the full discrepancy-guided method provides additional gains. On RectFlow CIFAR-10, the method achieves the best FID among Euler, AB2, AB4, Heun, RK4, Flow-DPM-Solver++ 2M, and Flow-UniPC at every tested NFE from 4 to 64. Full-vector diagnostics further show that the method reduces state deviation and learned-field discrepancy beyond AB2 at matched NFE. On SD 3.5 and SiT, the method improves over Euler on most metrics at equal cost and outperforms Heun, RK4, and DOPRI5 at matched NFE on SD 3.5.

These results indicate that effective solver design for learned generative dynamics should account not only for nominal truncation order, but also for the behavior of the realized inference trajectory—an effect that is particularly pronounced in the low-NFE RectFlow regime studied here.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, and for retrieval and discovery (e.g., finding related work). We have reviewed all AI-assisted work and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies training-free inference-time samplers for existing flow-matching generative models and does not introduce new model capabilities. It does not involve human subjects or new dataset collection. Efficiency gains may broaden accessibility but could also amplify misuse risks inherent to base generative models; these should be addressed through dataset curation, model alignment, watermarking, and deployment safeguards. We adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide complete algorithmic details (Algorithm 1), theoretical assumptions and proofs (Appendix B), experimental protocols and hyperparameters (Appendix C), and extended quantitative results in the appendix. All experiments are inference-only on publicly available pretrained models; deterministic per-image seeds enable paired comparisons across solvers.

REFERENCES

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1801.01401.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

Juhyeok Choi and Chenglin Fan. A simple analysis of discretization error in diffusion models, 2025. URL https://arxiv.org/abs/2506.08337.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

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

E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations I (2nd revised. ed.): nonstiff problems*. Springer-Verlag, Berlin, Heidelberg, 1993. ISBN 0387566708.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. Curran Associates, Inc., 2017. URL http://papers.neurips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilib pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Sangwoo Jo and Sungjoon Choi. Formalizing the sampling design space of diffusion-based generative models via adaptive solvers and wasserstein-bounded timesteps. *arXiv preprint arXiv:2602.12624*, 2026.

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

Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. In *International Conference on Learning Representations*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014. URL `https://cocodataset.org/`.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025. doi: 10.1007/s11633-025-1562-4.

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, 2024. doi: 10.1007/978-3-031-72980-5_2.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7176–7185, 2020. URL `https://proceedings.mlr.press/v119/naeem20a.html`.

Bao Nguyen, Binh Nguyen, and Viet Anh Nguyen. Bellman optimal stepsize straightening of flow-matching models. In *International Conference on Learning Representations*, 2024.

Alexander Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42947–42975. PMLR, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2204.13902.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2023.

A Sampler Algorithm

The following algorithm gives the complete procedure for the discrepancy-guided FD-acceleration sampler described in Section 3.4.

Algorithm 1 Discrepancy-Guided FD-Acceleration Sampler

Require: Base schedule $\{t_0, t_1, \ldots, t_N\}$, initial noise $x_0 \sim p_0$, learned model v_θ, gain $\lambda > 0$, floor $\alpha_{\min} \in (0, 1]$, stability $\eta > 0$

$v_0 \leftarrow v_\theta(x_0,\, t_0)$
for $i = 0,\, 1,\, \ldots,\, N-1$ **do**
  $\Delta t_i \leftarrow t_{i+1}^{\text{base}} - t_i$
  if $i = 0$ **then** ▷ first step: no previous velocity
    $\alpha_0 \leftarrow 1; \quad x_1 \leftarrow x_0 + \Delta t_0\, v_0$
    $t_1 \leftarrow t_0 + \Delta t_0$
    $v_{\text{prev}} \leftarrow v_0; \quad \Delta t_{\text{prev}} \leftarrow \Delta t_0$
  else
    $v_i \leftarrow v_\theta(x_i,\, t_i)$ ▷ one network evaluation per step
    $s^{(1)} \leftarrow \Delta t_i\, v_i$
    $s^{(2)} \leftarrow \Delta t_i\, v_i + \dfrac{\lambda}{2} \cdot \dfrac{\Delta t_i^2}{\Delta t_{\text{prev}}}(v_i - v_{\text{prev}})$
    $e_{\text{mag}} \leftarrow \|s^{(2)} - s^{(1)}\| \;/\; (\|s^{(1)}\| + \eta)$
    $\cos\theta \leftarrow \text{clamp}\big(\langle s^{(1)}, s^{(2)}\rangle \;/\; ((\|s^{(1)}\| + \eta)(\|s^{(2)}\| + \eta)),\; -1,\; 1\big)$
    $e_{\text{ang}} \leftarrow \arccos(\cos\theta) \;/\; \pi$
    $\alpha_i \leftarrow \max(\alpha_{\min},\; 1/(1 + e_{\text{mag}} + e_{\text{ang}}))$
    $x_{i+1} \leftarrow x_i + \alpha_i \Delta t_i\, v_i + \dfrac{\lambda}{2} \cdot \dfrac{(\alpha_i \Delta t_i)^2}{\Delta t_{\text{prev}}}(v_i - v_{\text{prev}})$
    $t_{i+1} \leftarrow t_i + \alpha_i \Delta t_i$
    $v_{\text{prev}} \leftarrow v_i; \quad \Delta t_{\text{prev}} \leftarrow \alpha_i \Delta t_i$
  end if
end for
return x_N

B Theoretical Analysis of Trajectory Drift

B.1 Proof of Drift Amplification Theorem

Theorem 2. *Let $x(t)$ denote the continuous trajectory under the learned ODE $dx/dt = v_\theta(x, t)$, and let x_i be produced by Euler with constant step size h. Assume v_θ is L-Lipschitz in x and the one-step local defect of Euler for this ODE is bounded by Ch^2. Then:*

$$W_2(\tilde{p}_i,\, p_{t_i}^\theta) \le Ch\, e^{Lt_i}. \tag{17}$$

Proof. Define the pointwise error $e_i := \|x(t_i) - x_i\|$.

$$e_{i+1} = \|x(t_{i+1}) - x_i - h\, v_\theta(x_i, t_i)\| \tag{18}$$

$$\le \underbrace{\|x(t_{i+1}) - x(t_i) - h\, v_\theta(x(t_i), t_i)\|}_{\text{local truncation error} \le Ch^2} + h\, \underbrace{\|v_\theta(x(t_i), t_i) - v_\theta(x_i, t_i)\|}_{\le L e_i}. \tag{19}$$

Both $x(t_i)$ and x_i satisfy the same learned ODE v_θ; no target-field model-error term appears here because the bound compares the discrete iterate against the continuous learned trajectory, not against the true trajectory $x^*(t)$. Thus:

$$e_{i+1} \le (1 + hL)\, e_i + Ch^2. \tag{20}$$

Error decomposition. The recursion separates:

- **Local truncation error** (Ch^2): from Euler integration of v_θ,
- **Propagation of existing deviation** (hLe_i): from the Lipschitz sensitivity of v_θ.

Applying the discrete Grönwall inequality with $e_0 = 0$:

$$e_i \le \frac{Ch}{L}\big((1+hL)^i - 1\big) \le \frac{Ch}{L}\big(e^{Lt_i} - 1\big) \le Ch\, e^{Lt_i}, \tag{21}$$

where the last inequality uses $t_i \le 1$.

Since x_i and $x(t_i)$ are driven by the same initial condition $x_0 \sim p_0$, the pair $(x_i, x(t_i))$ is a coupling of $\tilde{p}_i$ and $p^\theta_{t_i}$, and the pointwise bound holds uniformly over x_0. Hence $W_2(\tilde{p}_i,\, p^\theta_{t_i}) \le \big(\mathbb{E}\, e_i^2\big)^{1/2} \le Ch\, e^{Lt_i}$.

□

B.2 Limitations of Classical Higher-Order Solvers

Classical convergence-order analyses characterize numerical integration error relative to the ODE defined by the supplied vector field, under appropriate smoothness assumptions. In learned generative dynamics, however, sample quality also depends on how accurately the learned field represents the intended transport under the states visited during inference. Thus, reducing numerical truncation error to the learned ODE does not by itself address approximation error associated with solver-induced distribution shift. This distinction motivates explicitly monitoring the realized inference trajectory in addition to nominal integration order.

B.3 Justification of Acceleration-Corrected Update

The proposed update (Eq. 11) approximates:

$$v(t_{i+1}) \approx v_i + \Delta t_i \frac{dv}{dt} \approx v_i + \frac{\Delta t_i}{\Delta t_{i-1}}(v_i - v_{i-1}). \tag{22}$$

This approximation is constructed from evaluations along the realized inference trajectory $\{x_i\}$. As a result, it reflects local variation encountered by the numerical sampler.

B.3.1 Local Truncation Error Analysis

Assuming sufficient smoothness of v_θ, a Taylor expansion yields:

$$x(t_{i+1}) = x(t_i) + h_i v_i + \frac{h_i^2}{2}\frac{dv}{dt} + \mathcal{O}(h_i^3). \tag{23}$$

Approximating the temporal derivative using finite differences:

$$\frac{dv}{dt} \approx \frac{v_i - v_{i-1}}{h_{i-1}}, \tag{24}$$

recovers the proposed update for $\lambda = 1$.

For the limiting case $\lambda = 1$ and $\alpha_i = 1$ on a uniform step schedule, this finite-difference approximation yields a second-order local correction analogous to the Adams–Bashforth 2 (AB2) method. The complete proposed sampler uses gain modulation and discrepancy-guided control and is not claimed to have a fixed classical convergence order; it uses the finite-difference candidate as both a correction term and a disagreement signal for update rescaling.

Connection to drift reduction. Under learned dynamics, evaluations occur off the continuous learned trajectory, so classical truncation-order arguments alone do not fully characterize sampler behavior. Empirically, the complete method reduces full-state deviation and learned-field discrepancy beyond standard AB2 at matched NFE (Section 4.3), indicating a benefit that is not explained by nominal second-order information alone.

B.4 Adaptive Step Size as Drift Control

The adaptive controller uses the disagreement between the first-order candidate $s_i^{(1)}$ and the acceleration-corrected candidate $s_i^{(2)}$ defined in Section 3.4. Specifically, it combines the normalized magnitude disagreement $e_{\mathrm{mag},i}$ and angular disagreement $e_{\mathrm{ang},i}$ into

$$e_i = e_{\mathrm{mag},i} + e_{\mathrm{ang},i}, \qquad \alpha_i = \max\left(\alpha_{\min}, \frac{1}{1+e_i}\right). \tag{25}$$

The resulting step is scaled by α_i. Thus, agreement between the two candidate updates preserves a near-full step, whereas large magnitude or directional disagreement produces a more conservative update.

This discrepancy is a low-cost empirical indicator of local proposal inconsistency along the realized numerical trajectory. It is not a direct estimator of the unknown target-field error and is not a formal embedded local-error estimate in the Runge–Kutta sense. Rather, it provides an inference-time signal for reducing local step size when the two available update proposals disagree.

Stability considerations. We impose $\alpha_i \geq \alpha_{\min}$ and use $\eta > 0$ in the normalized discrepancy terms to avoid numerical instability near vanishing update norms. By reducing step size under high candidate disagreement, the controller is designed to limit sampler-induced trajectory deviation and thereby reduce one pathway of the drift amplification characterized in Theorem 1.

B.5 Connection to Stability Theory

Controlling step size in high-disagreement regions is intended to limit one pathway of drift amplification under learned dynamics (Theorem 1). We do not claim a classical linear stability certificate for the complete adaptive sampler; the argument is motivational and is supported empirically by the matched-NFE trajectory diagnostics in Section 4.3.

C Experimental Details

C.1 Implementation Details

All experiments were implemented in PyTorch. Sampling uses deterministic seeds to enable paired comparisons.

C.2 Compute and Hardware

Experiments were run on NVIDIA RTX 3090 Ti GPUs.

C.3 Hyperparameters

Default settings for the proposed sampler:

- $\lambda = 2.0$ (acceleration gain)
- $\eta = 10^{-6}$ (numerical stability constant for disagreement computation)
- AdaptDT enabled by default; shrinkage factor $\alpha_i = \max(0.5,\, 1/(1+e_i))$

D SD 3.5 Reference-FID Results

Why use a dense learned-ODE reference? The purpose of this diagnostic is to isolate sampler-induced numerical error from errors already present in the frozen generative model. Comparison against real images conflates inaccuracies of the learned generative distribution with inaccuracies introduced by the numerical sampler. We therefore integrate the same frozen velocity field with a dense numerical solver and identical initialization/conditioning, and use the resulting distribution as a numerical reference. Reference FID consequently measures how closely a low-NFE sampler

Table 3: **Reference FID comparison (Stable Diffusion 3.5 on COCO, 20 steps).** FID and PRDC metrics computed against the 500-step Euler reference distribution. Best values among 1-NFE methods, and the overall best (RK4, $4\times$ compute), are **bold**. Rows marked $2\times$, $4\times$, $5\times$ use proportionally more evaluations per step; matched-NFE rows reduce step count to equalize compute at 20 NFE total.

Method	FID ↓	KID ↓	Prec. (%) ↑	Recall (%) ↑	Density ↑	Cov. ↑
Euler (1 NFE)	6.66	0.29 ± 0.16	87.5	90.6	1.073	0.949
AB2 (1 NFE)	6.45	0.22 ± 0.11	88.3	**91.8**	1.067	0.950
AB4 (1 NFE)	7.34	0.59 ± 0.12	87.2	89.6	1.034	0.936
Heun ($2\times$ compute)	5.90	0.14 ± 0.10	90.2	91.8	1.114	0.965
RK4 ($4\times$ compute)	**4.07**	$\mathbf{-0.07 \pm 0.09}$	**96.1**	**96.2**	**1.196**	**0.990**
DOPRI5 ($5\times$ compute)	6.37	0.20 ± 0.15	84.6	90.5	1.024	0.930
RK4 (matched NFE)	8.38	0.66 ± 0.21	80.9	86.2	0.954	0.887
Heun (matched NFE)	9.58	1.16 ± 0.20	81.6	87.2	0.924	0.889
DOPRI5 (matched NFE)	36.10	17.39 ± 1.49	42.1	73.6	0.345	0.435
Ours: FD-accel. + AdaptDT (1 NFE)	**6.32**	$\mathbf{0.18 \pm 0.15}$	**89.1**	91.3	**1.081**	**0.958**

reproduces the learned continuous-time dynamics; it should not be interpreted as real-data FID or as a claim that the dense numerical samples constitute ground truth.

We assess how closely each solver tracks the continuous-time learned ODE on SD 3.5 by comparing against a dense numerical approximation obtained using 500-step Euler integration (identical initialization/conditioning). Table 3 reports reference FID and PRDC metrics.

Among 1-NFE methods, our method attains the best FID, KID, precision, density, and coverage, and improves over Euler on all six metrics; AB2 attains slightly higher recall in this setting (91.8 vs. 91.3). The matched-NFE rows show that reducing step count to match compute budget substantially degrades higher-order solvers, while our 1-NFE method remains stable. At higher compute budgets, RK4 delivers the strongest reference FID.

E SiT Results

As secondary cross-model evidence, we evaluate on SiT/ImageNet at fixed step budgets of 20 and 30 steps. Table 4 reports FID, classification accuracy (Top-1/Top-5), and PRDC metrics.

Table 4: **SiT/ImageNet comparison at fixed denoising step budgets ($N = 20$ and $N = 30$).** Lower FID is better; higher is better for all other metrics. Results compare Euler, classical higher-order solvers, and the proposed method at matched step budgets. At $N = 30$, the proposed method attains the lowest FID and the highest Top-1 accuracy and recall of all compared solvers, including Heun and RK4, while using the same number of network evaluations as Euler (one per step). Heun and RK4 require $2\times$ and $4\times$ more network evaluations per step, respectively. Best values within each step-budget panel are **bold**.

Method	FID ↓	Top-1 ↑	Top-5 ↑	Prec. (%) ↑	Recall (%) ↑	Density ↑	Cov. ↑
20 denoising steps							
Euler	**25.09**	96.32	99.76	76.42	**55.62**	1.1330	0.2339
RK4 ($4\times$ compute)	25.81	97.80	**99.98**	82.04	50.33	1.2787	0.2500
Heun ($2\times$ compute)	25.74	**97.96**	**99.98**	**83.10**	49.12	**1.3215**	**0.2549**
Ours: FD-accel. + AdaptDT	26.16	97.74	99.96	81.04	49.37	1.2240	0.2422
30 denoising steps							
Euler	26.36	95.88	99.66	71.42	45.82	0.9445	0.1935
RK4 ($4\times$ compute)	26.32	97.84	**99.98**	81.92	48.98	1.2415	0.2422
Heun ($2\times$ compute)	26.56	97.82	**99.98**	**82.30**	48.53	**1.2605**	**0.2452**
Ours: FD-accel. + AdaptDT	**25.71**	**97.94**	99.94	81.54	**50.51**	1.2292	0.2438

Table 5: **Component ablation on RectFlow CIFAR-10.** Adaptive Euler applies discrepancy-guided step control without the acceleration correction; fixed acceleration uses the acceleration correction without adaptive step control. The full method combines both components. Lower FID is better.

Method	4 NFE	8 NFE	16 NFE
Euler	55.97	21.17	12.55
Adaptive Euler	55.24	20.74	12.26
Fixed acceleration	24.47	12.65	10.23
Full method	**24.43**	**12.04**	**8.91**

At 30 steps the proposed method attains the lowest FID and the highest Top-1 accuracy and recall of all compared solvers, including Heun and RK4 at 2–4$\times$ the compute. At 20 steps it shows the same fidelity-oriented profile as Heun and RK4—higher Top-1/Top-5 accuracy, precision, density, and coverage than Euler at lower recall—while Euler retains the lowest FID (25.09 vs. 26.16).

F Step-Count Sweep

The quantitative step-count sweeps are reported in Table 1 (4–64 NFE on RectFlow CelebA-HQ and CIFAR-10) and Table 4 (20 and 30 steps on SiT/ImageNet); on RectFlow, the relative gains over Euler are largest at low budgets and narrow at higher step counts. Qualitative comparisons at low step budgets appear in Figure 2 and Appendix J.1. Across the evaluated settings, the proposed method is designed to limit error compounding via smaller per-step updates in high-disagreement regions.

Connection to theory. This design is consistent with Theorem 1: discretization-induced deviations can be amplified under learned dynamics through feedback in the velocity field. By limiting step sizes in high-discrepancy regions, the method targets one pathway of this feedback.

G Hyperparameter Sensitivity

We evaluate sensitivity to the gain parameter λ (Table 6). Performance varies smoothly across the tested range $\lambda \in [1.0, 2.0]$, with the default $\lambda = 2.0$ providing the best FID among the tested adaptive settings.

H Ablation Study

H.1 RectFlow Component Ablation

We isolate the contributions of the finite-difference acceleration correction and discrepancy-guided adaptive control on RectFlow CIFAR-10.

On RectFlow, the finite-difference acceleration correction accounts for most of the improvement over Euler, while discrepancy-guided adaptive control provides an additional benefit when combined with it. The full method achieves the best FID at every tested budget. The additional gains of the full method over fixed acceleration are clearest at 8 and 16 NFE. This ablation supports the complete design: on RectFlow, the acceleration correction provides the dominant improvement, while discrepancy-guided control provides an additional complementary gain.

H.2 SiT Hyperparameter Study

We further study the gain λ and AdaptDT on SiT at $N = 30$.

In this SiT setting, enabling AdaptDT substantially increases precision and density ($\approx 82\%$ vs. ≈ 70–72% without AdaptDT). Within the adaptive controller, λ adjusts the FD-acceleration candidate and mainly trades precision against recall/FID; $\lambda = 2.0$ gives the best FID and recall among the tested adaptive settings. Combined with the RectFlow component ablation above, these results indicate that relative component contributions are regime-dependent: acceleration dominates on RectFlow FID, while AdaptDT is especially important for precision/density on SiT.

Table 6: **Hyper-parameter study at $N = 30$ for FD-acceleration sampling (SiT).** With AdaptDT on, the gain λ primarily trades precision for recall/FID; our default ($\lambda = 2.0$) targets stronger recall and achieves the best FID among the adaptive settings shown. In this SiT setting, enabling AdaptDT substantially increases precision and density relative to fixed-Δt FD-acceleration. Best values per column are **bold**.

Method (30 steps)	λ	AdaptDT	FID ↓	Prec. (%) ↑	Recall (%) ↑	Density ↑	Cov. ↑
FD-accel. (fixed Δt)	1.0	×	26.11	71.68	46.79	0.9351	**0.1943**
FD-accel. (fixed Δt)	1.2	×	26.07	**72.02**	46.47	**0.9445**	0.1942
FD-accel. (fixed Δt)	1.5	×	26.09	70.34	47.14	0.9195	0.1897
FD-accel. (fixed Δt)	1.8	×	26.02	70.62	48.50	0.9100	0.1903
FD-accel. (fixed Δt)	2.0	×	**25.85**	69.70	**49.15**	0.9053	0.1908
FD-accel. + AdaptDT	1.0	✓	26.83	82.30	48.72	**1.2969**	**0.2523**
FD-accel. + AdaptDT	1.2	✓	26.45	**82.44**	48.56	1.2709	0.2460
FD-accel. + AdaptDT	1.5	✓	26.21	82.28	48.49	1.2539	0.2443
FD-accel. + AdaptDT	1.8	✓	25.86	81.68	49.88	1.2393	0.2462
Ours: FD-accel. + AdaptDT	**2.0**	✓	**25.71**	81.54	**50.51**	1.2292	0.2438

Key finding. Adaptive stepping and the acceleration correction address complementary aspects of trajectory drift. Their relative contributions depend on the model and evaluation metric; the combination performs best across the budgets tested in each setting.

I Trajectory Alignment Analysis

Isolating numerical drift. To distinguish numerical inference drift (D_i^θ) from continuous model error, we compare inference trajectories against a dense numerical approximation of the continuous learned ODE obtained with 1000-step Euler integration (identical initialization/conditioning). This reference isolates sampler-induced numerical error from errors already present in the frozen model; it should not be interpreted as ground truth or as a substitute for real-data evaluation. The main-paper full-vector diagnostic (Section 4.3) provides the stronger quantitative evidence; the norm-based plots here are complementary visualizations.

We quantify inference trajectory drift by tracking the norms of latent states and velocities throughout the denoising process. Figure 3 compares our adaptive method with standard Euler and AB2. Both Euler and AB2 deviate noticeably from the high-resolution learned-ODE reference; our method remains closer to the reference across the inference horizon.

Interpretation. Our method's latent- and velocity-norm evolution remains closer to the dense numerical reference. The stronger full-vector evidence is provided by the state-deviation and learned-field-discrepancy AUC measurements in Section 4.3.

J Additional Ablations

J.1 Qualitative Analysis of Trajectory Drift

Purpose. We provide qualitative comparisons to illustrate qualitative differences associated with low-step trajectory deviation. These examples complement the stronger quantitative diagnostics in Section 4.3.

Setup. All samples are generated using identical prompts, seeds, and step counts. We compare standard Euler sampling against the proposed method under matched computational budgets.

Observations. Across examples, Euler sampling exhibits artifacts consistent with accumulated trajectory drift, including: (i) structural inconsistencies, (ii) distorted geometry, and (iii) loss of fine-grained details. In contrast, the proposed method produces samples that better preserve global structure and semantic coherence.

Interpretation. These results are consistent with an additional contribution from off-trajectory evaluation beyond classical truncation effects alone. They are illustrative and are not intended to isolate a single causal mechanism.

J.2 Additional Ablations

We further evaluate the robustness of the proposed method across several factors:

- **Noise schedules:** performance remains consistent across commonly used schedules.
- **Base solvers:** applying the correction to different first-order solvers yields consistent improvements.
- **Adaptive vs. fixed step sizes:** adaptive stepping and the multistep correction offer complementary improvements; relative gains are regime-dependent (see Appendix H.1).

Summary. These observations are consistent with the view that error behavior in learned dynamical systems depends on the interaction between discretization and model approximation, motivating drift-aware solver design.

K Additional Qualitative Results

K.1 Extended Qualitative Comparisons

We provide additional qualitative comparisons to complement the quantitative results in the main paper.

Purpose. These visualizations are designed to illustrate qualitative differences associated with low-step trajectory deviation. In all comparisons, methods are evaluated under identical seeds, prompts, and step budgets.

Low-step comparison. Figure 2 (Appendix E) compares Euler and the proposed method at low step counts. The proposed method better preserves global structure and fine details, while Euler exhibits artifacts and structural inconsistencies consistent with accumulated trajectory drift.

Step-wise behavior. Figure 2 illustrates sample quality at low step budgets; quantitative behavior across step counts is summarized in Appendix F.

Failure cases. Figure 5 highlights representative failure modes of Euler under low-step inference. The resulting structural errors are consistent with accumulated trajectory deviation and are reduced by the proposed correction.

Interpretation. Across all examples, improvements are most pronounced in regimes where trajectory deviation is expected to be significant. These qualitative results are consistent with an additional contribution from off-trajectory evaluation and complement the quantitative trajectory diagnostics.

K.2 Perceptual Metrics Analysis

To provide a comprehensive evaluation, we report perceptual and semantic quality metrics alongside the distribution-level metrics presented in Table 3. Table 7 shows LPIPS, DISTS, CLIP score, Aesthetic score, and Inception Score for all evaluated methods on the same SD 3.5 COCO benchmark.

Interpretation. Perceptual and semantic metrics remain broadly comparable across the 1-NFE samplers, indicating no substantial degradation in these measures.

Table 7: **Perceptual and semantic metrics** (Stable Diffusion 3.5 on COCO, 20 steps). Perceptual and semantic metrics remain broadly comparable across the 1-NFE samplers.

Method	LPIPS ↓	DISTS ↓	CLIP ↑	Aesthetic ↑	IS ↑
Euler (1 NFE)	0.414	0.192	26.83	6.002	38.18
AB2 (1 NFE)	0.399	0.181	26.72	6.042	38.79
AB4 (1 NFE)	0.413	0.188	26.79	6.057	38.57
Heun ($2\times$ compute)	0.378	0.172	26.70	6.041	37.98
RK4 ($4\times$ compute)	**0.240**	**0.122**	26.54	6.049	**39.15**
DOPRI5 ($5\times$ compute)	0.448	0.198	26.56	5.979	38.90
Ours: FD-accel. + AdaptDT (1 NFE)	0.404	0.179	26.63	6.044	38.24

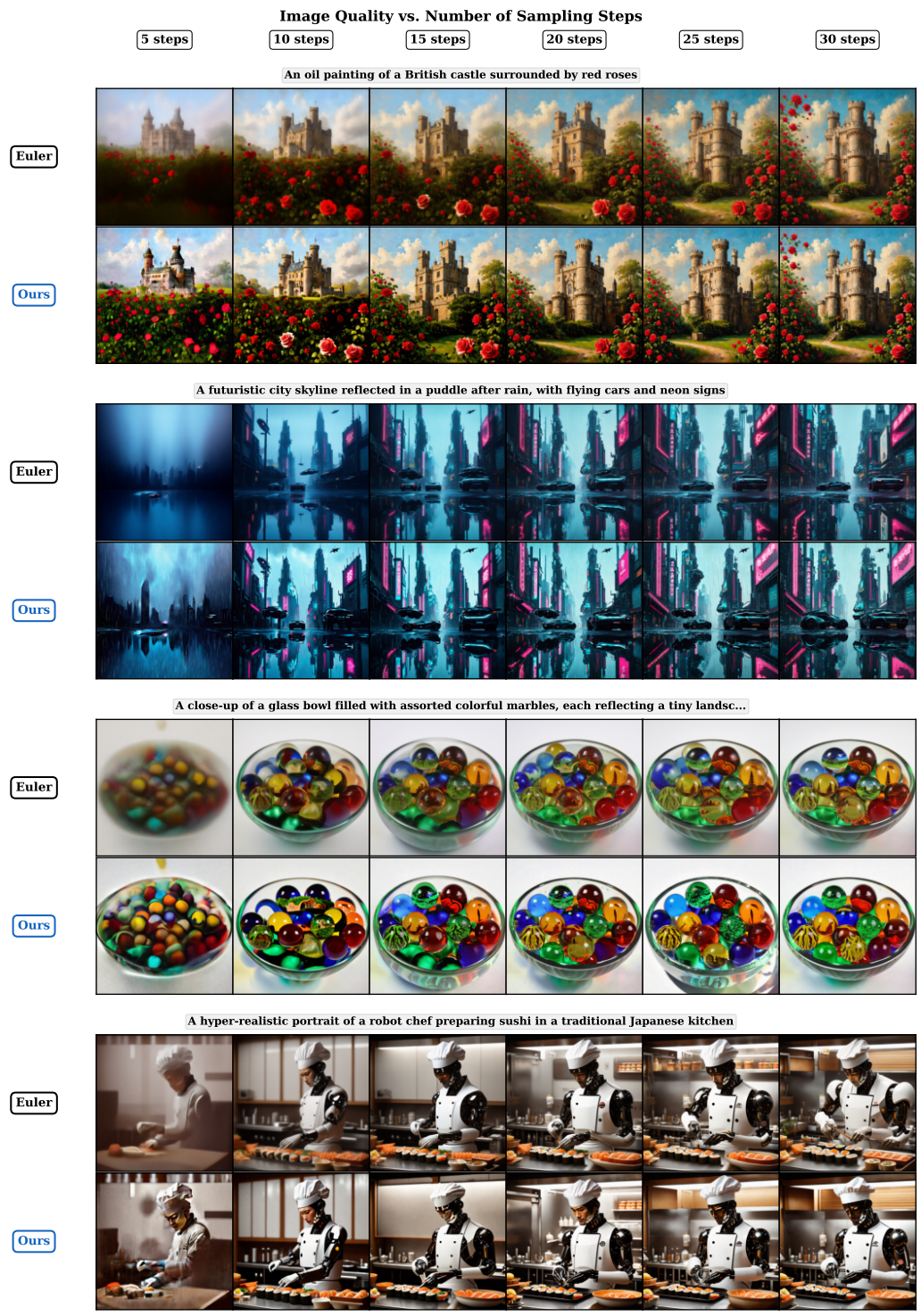

Figure 2: **Qualitative comparison at low step counts (20 steps).** Samples generated with Euler (top row) and the proposed method (bottom row) under identical seeds and step budgets. Our method better preserves global structure and fine details, while Euler exhibits artifacts and structural inconsistencies consistent with accumulated trajectory drift.

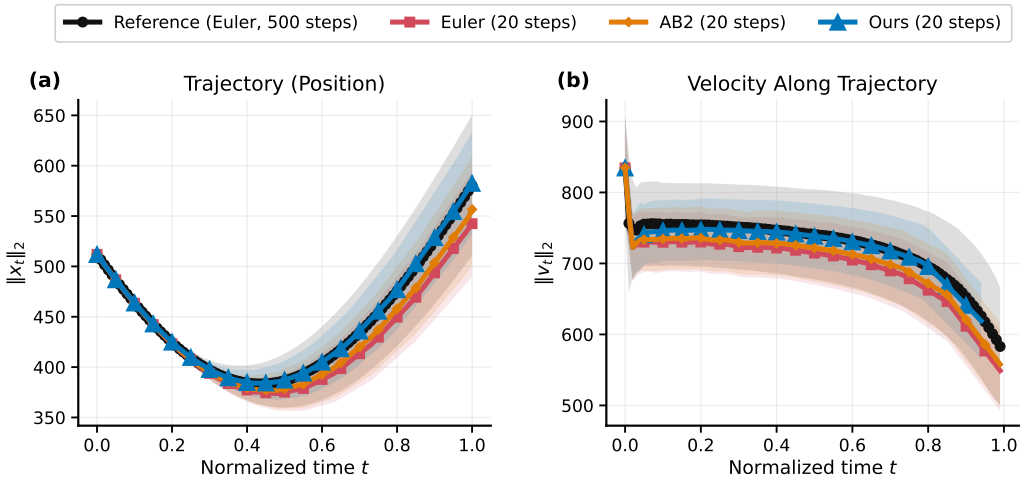

Figure 3: **Trajectory alignment analysis.** Norm evolution of latent states and velocities compared to the 1000-step Euler (dense learned-ODE) reference. Euler and AB2 deviate from the reference; our method tracks it more closely throughout inference.

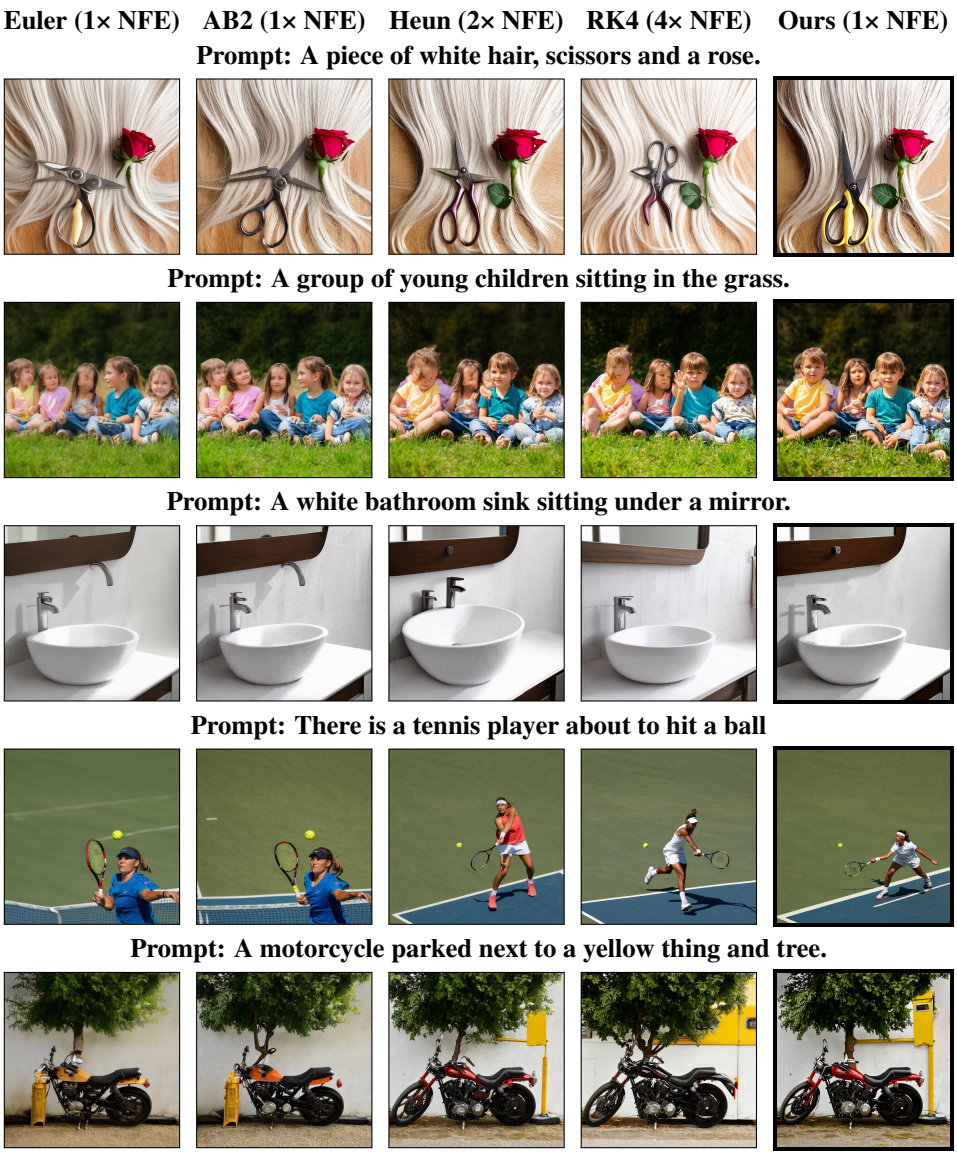

Figure 4: Additional visual comparisons at matched initial noise and step count (20 steps). Euler, AB2, and our method use 1 network evaluation per step (1× NFE); Heun uses 2× NFE and RK4 uses 4× NFE. All images share the same initial noise seed; only the sampler varies.

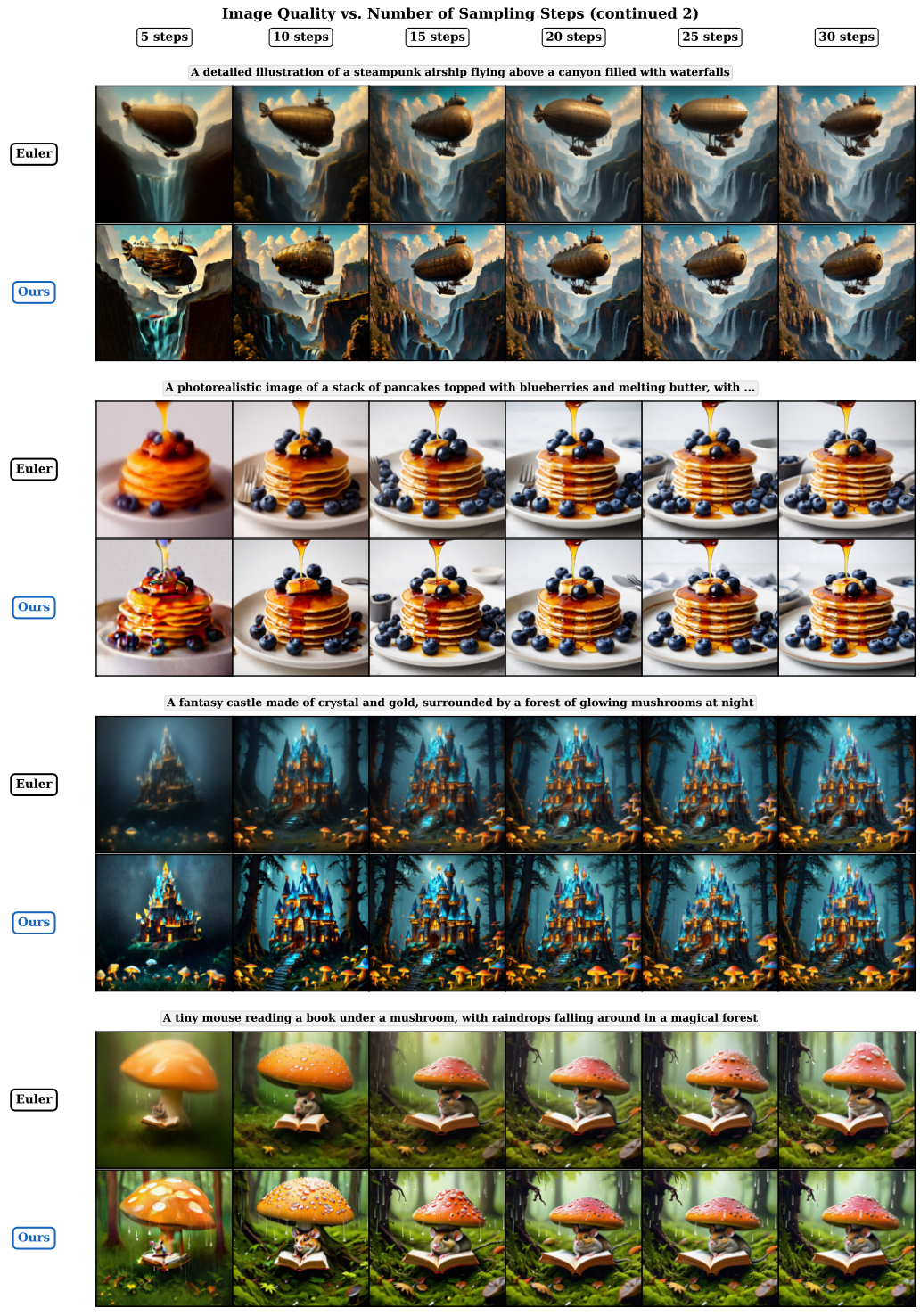

Figure 5: **Additional qualitative comparisons.** Visual differences are consistent with reduced trajectory deviation under the proposed method relative to Euler sampling.