Beyond Euler: ODE Samplers for Rectified Flow Generative Models

Qiwen Xiao
Samueli School of Engineering
University of California, Irvine
Irvine, CA 92697
qiwenx6@uci.edu

Omar Attaalla **William Y. Chung**
Donald Bren School of Information & Computer Sciences
University of California, Irvine
Irvine, CA 92697
oattaall@uci.edu chungwy1@uci.edu

Abstract

Rectified flow (RF) models learn an ordinary differential equation (ODE) that transports Gaussian noise to data along nearly straight trajectories, and can therefore generate samples with far fewer function evaluations (NFEs) than traditional diffusion models. Once an RF vector field is trained, however, the choice of numerical ODE solver remains largely unexplored. In this work we fix a conditional rectified flow model trained on the EuroSAT remote-sensing dataset and study the sampling problem in isolation. We compare six explicit ODE solvers under a common experimental setting, measuring Fréchet Inception Distance (FID), effective NFEs, wall-clock time, and geometric properties of the trajectories. Results show that Euler is the cheapest but performs worst, AB2 and Midpoint clearly improve FID at comparable or moderately higher cost, and adaptive RK23 attains the best FID with an intermediate evaluation budget, outperforming even high-cost RK4. Our results suggest that simple Euler sampling is suboptimal for rectified flows and that multi-step or adaptive solvers provide a better trade-off.

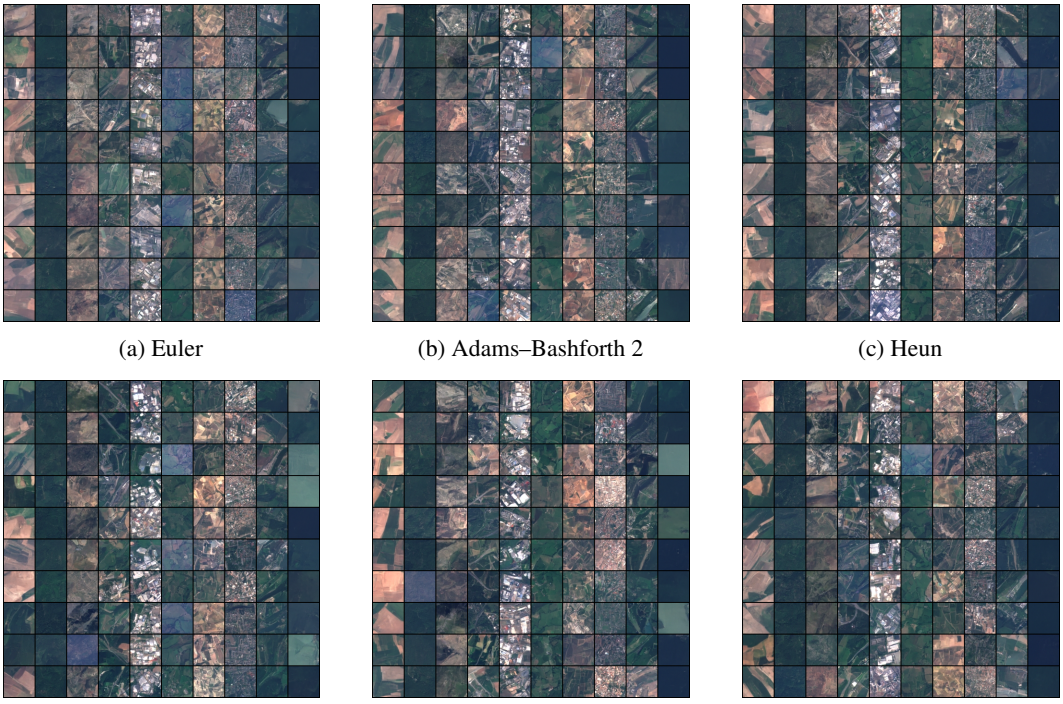
1 Introduction

Diffusion and score-based generative models have achieved state-of-the-art image generation quality by learning to reverse a gradual noising process, but they typically require tens to hundreds of function evaluations (NFEs) at inference time, leading to high sampling cost [1,2]. The associated probability-flow ODE perspective suggests that sampling can instead be cast as solving an ordinary differential equation driven by a neural vector field.

Flow Matching (FM) provides a simulation-free way to train such neural ODEs by directly regressing a time-dependent velocity field along prescribed probability paths between noise and data [3]. Rectified Flow (RF) further encourages these paths to be nearly straight line segments in data space, enabling surprisingly good samples with very few Euler steps [4]. In principle, once an RF model $v_\theta(x, t, y)$ is trained, we are free to plug in any off-the-shelf ODE solver at test time, but there is little systematic understanding of how different solvers trade off quality and cost in this setting.

In this work, we fix a conditional RF model trained on the EuroSAT remote-sensing dataset [5] and focus entirely on the sampling problem. We compare several explicit ODE solvers (Euler, Midpoint, Heun, RK4, Adams–Bashforth 2, and adaptive RK23) under matched compute budgets, and evaluate them not only by Fréchet Inception Distance (FID), but also by geometric diagnostics including path length and an approximation of integrated curvature of their generated trajectories.

Final report. COMPSCI 274E, Fall 2025. University of California, Irvine.



(a) Euler (b) Adams–Bashforth 2 (c) Heun

(d) Midpoint (e) RK4 (f) RK23 (adaptive)

Figure 1: Class-conditioned generation of EuroSAT samples for different ODE samplers. From left to right, the classes are: Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River, Sea/Lake.

2 Methodology

2.1 Problem Setting and Model

We study fast sampling for RF models on conditional image generation. Let $x \in \mathbb{R}^{C\times H\times W}$ denote an image and $y \in \{0, \dots, K-1\}$ a class label. Sampling proceeds by solving an ODE on $t \in [0, 1]$,

$$\frac{dx_t}{dt} = v_\theta(x_t, t, y), \qquad x_{t=0} = z \sim \mathcal{N}(0, I), \tag{1}$$

to obtain $x_{t=1}$ as a sample. The velocity field v_θ is parameterized by a conditional U-Net.

RF training objective. Given a data sample x and a noise sample z, define the linear interpolation $x_t \coloneqq (1-t)\, z + t\, x$. The RF objective regresses the target velocity $x - z$ at random $t \sim \mathrm{Beta}(0.5, 0.5)$:

$$\mathcal{L}_{\mathrm{RF}}(\theta) \;=\; \mathbb{E}_{x,z,t}\Big[\, \| \, v_\theta(x_t, t, y) - (x - z) \, \|_2^2 \,\Big]. \tag{2}$$

We train a single RF model and do not modify the training loss when comparing samplers, so that all differences arise solely from the numerical integrator used at inference.

Conditioning and classifier-free guidance. We implement class conditioning via an embedding of y that is added to the time embedding at each block of the U-Net. For optional classifier-free guidance (CFG) with scale $s \geq 0$, we evaluate two velocities and combine them

$$v_\theta^{\mathrm{CFG}}(x, t, y; s) \;=\; (1 + s)\, v_\theta(x, t, y) \;-\; s\, v_\theta(x, t, \texttt{uncond}), \tag{3}$$

where `uncond` is a learned null class (used when $y = -1$).

2.2 Samplers Under Comparison

We compare six ODE solvers (all explicit, single- or multi-step). For a uniform grid $t_k = kh$, $h = \frac{1}{N}$, we write $f_k \coloneqq v_\theta(x_k, t_k, y)$ and denote one step by $x_{k+1} = \Phi_h(x_k, t_k)$.

(1) Euler (explicit, first order).

$$x_{k+1} = x_k + h\, f_k. \quad \text{NFE per step: } 1. \tag{4}$$

(2) Midpoint (explicit RK2).

$$k_1 = f_k, \qquad k_2 = v_\theta(x_k + \tfrac{h}{2}k_1, t_k + \tfrac{h}{2}, y), \tag{5}$$

$$x_{k+1} = x_k + h\, k_2. \quad \text{NFE per step: } 2. \tag{6}$$

(3) Heun / Trapezoid (explicit RK2).

$$k_1 = f_k, \qquad k_2 = v_\theta(x_k + h\, k_1, t_k + h, y), \tag{7}$$

$$x_{k+1} = x_k + \tfrac{h}{2}\,(k_1 + k_2). \quad \text{NFE per step: } 2. \tag{8}$$

(4) Classical RK4 (explicit, fourth order).

$$k_1 = f_k, \quad k_2 = v_\theta(x_k + \tfrac{h}{2}k_1, t_k + \tfrac{h}{2}, y), \tag{9}$$

$$k_3 = v_\theta(x_k + \tfrac{h}{2}k_2, t_k + \tfrac{h}{2}, y), \quad k_4 = v_\theta(x_k + h\, k_3, t_k + h, y), \tag{10}$$

$$x_{k+1} = x_k + \tfrac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4). \quad \text{NFE per step: } 4. \tag{11}$$

(5) Adams–Bashforth 2 (AB2, explicit two-step). Bootstrap the first step with Heun to obtain x_1 and f_1. For $k \geq 1$:

$$x_{k+1} = x_k + h\Big(\tfrac{3}{2}\, f_k - \tfrac{1}{2}\, f_{k-1}\Big). \quad \text{NFE: 2 (bootstrap) } + 1 \text{ per subsequent step.} \tag{12}$$

AB2 can exploit smoothness of v_θ across t to reduce evaluations.

(6) Adaptive RK23 (Bogacki–Shampine 3(2)). We use an embedded Runge–Kutta pair that provides both a third-order solution $\tilde{x}_{k+1}$ and a second-order solution $\hat{x}_{k+1}$; their difference estimates local error e_k and controls the next step size h:

$$\tilde{x}_{k+1} = x_k + h\big(\tfrac{2}{9}k_1 + \tfrac{1}{3}k_2 + \tfrac{4}{9}k_3\big)\,, \tag{13}$$

$$\hat{x}_{k+1} = x_k + h\big(\tfrac{7}{24}k_1 + \tfrac{1}{4}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{8}k_4\big)\,, \tag{14}$$

$$e_k = \|(\tilde{x}_{k+1} - \hat{x}_{k+1})/\mathrm{sc}\|_2/\sqrt{d}, \tag{15}$$

where $k_1 = f_k$, $k_2 = v_\theta(x_k + \frac{h}{2}k_1, t_k + \frac{h}{2}, y)$, $k_3 = v_\theta(x_k + \frac{3h}{4}k_2, t_k + \frac{3h}{4}, y)$, $k_4 = v_\theta(\tilde{x}_{k+1}, t_k + h, y)$, $d = CHW$, and $\mathrm{sc} = \mathrm{atol} + \mathrm{rtol} \cdot \max(|x_k|, |\tilde{x}_{k+1}|)$ is a per-sample scaling. A standard controller updates $h \leftarrow \mathrm{clip}\big(\alpha\,(1/e_k)^{1/4}h,\, h_{\min},\, h_{\max}\big)$ with safety factor $\alpha \in (0, 1)$ and accepts the step if $e_k \leq 1$. We record the *effective* NFE, which varies across samples.

2.3 Geometry of Trajectories

Beyond image quality, we quantify how straight the numerical trajectories are.

Discrete path length. Given a discrete path $\{x_k\}_{k=0}^{N}$, the (arc) length is

$$L \;\approx\; \sum_{k=0}^{N-1} \big\|x_{k+1} - x_k\big\|_2. \tag{16}$$

Integrated curvature. Let $u_k = \frac{x_{k+1} - x_k}{\|x_{k+1} - x_k\|_2}$ be the unit step direction. The total turning along the path is

$$\mathcal{C} \;\approx\; \sum_{k=0}^{N-2} \arccos(\langle u_k, u_{k+1}\rangle)\,, \tag{17}$$

accumulated only where both steps are non-degenerate. It is natural to expect that lower values of L and $\mathcal{C}$, reflecting straighter flows, would correlate with fewer steps required to achieve a given image quality. However, our results do not fully support this intuition, and we examine the relationship among L, $\mathcal{C}$, and sample quality in Sec. 3. We also include a sketchy discussion regarding this discrepancy in Sec. D of the appendix.

Table 1: Comparison of ODE samplers on EuroSAT at 64 steps and $s = 0$ (no guidance), using a fixed rectified-flow model. L and $\mathcal{C}$ denote mean $\pm$ standard deviation of path length and integrated curvature over $2{,}048$ trajectories. Lower is better for FID. Eff. NFE here denotes the *total* effective NFE across 32 batches, with each batch containing 64 samples.

Method	Eff. NFE	Time (s)	FID ↓	L (path)	$\mathcal{C}$ (curv.)
Euler	2048	146.7	85.76	213.8 ± 68.6	1.78 ± 0.56
Heun	4096	291.2	85.63	234.0 ± 78.5	1.62 ± 0.52
Midpoint	4096	291.2	83.34	242.4 ± 84.9	1.61 ± 0.50
RK4	8192	579.6	84.16	247.9 ± 91.6	1.51 ± 0.48
AB2	2112	150.7	84.68	232.1 ± 77.5	2.03 ± 0.60
RK23	4524	320.2	**82.75**	251.4 ± 98.5	1.71 ± 0.60

2.4 Experimental Setup

Data and preprocessing. We use the EuroSAT land use and land cover classification dataset based on Sentinel-2 satellite imagery [5]. EuroSAT consists of 27,000 geo-referenced image patches of size 64×64 pixels collected over more than 30 European countries. Each patch belongs to one of 10 land cover classes: Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial Buildings, Pasture, Permanent Crop, Residential Buildings, River, or Sea/Lake. We use the RGB variant of the dataset, which retains only the red, green, and blue bands from the original 13-band Sentinel-2 imagery. We use EuroSAT (RGB) with $K = 10$ classes at 64×64 resolution. We perform a random split with 80% for training and 20% for validation and normalize with per-channel statistics:

$$\mu = (0.3444, 0.3803, 0.4078), \qquad \sigma = (0.0914, 0.0651, 0.0552).$$

Training and validation tensors remain in normalized space. For qualitative grids and FID computation, we unnormalize with $x \mapsto x \odot \sigma + \mu$ and clamp to $[0, 1]$ before writing PNGs.

Model and training. To isolate sampler effects, we train a single RF model with a small conditional U-Net with around 7.8 million parameters using AdamW, learning rate 2×10^{-4}, and batch size 128 for 600 epochs. We fix seed 42, persist the split, and log configuration and parameter counts for reproducibility. All experiments are done on an NVIDIA GeForce RTX 5060 Laptop GPU.

Sampling. We evaluate methods Euler, Midpoint, Heun, RK4, AB2, and RK23-adaptive with NFE=64 and no sampling guidance scale. For each condition, we generate $M = 2048$ samples with class labels uniformly cycled for class-conditional grids. For RK23, we set $(\mathrm{rtol}, \mathrm{atol}) = (10^{-3}, 10^{-4})$, safety factor 0.9, and $h_{\min} = 2^{-9}$, $h_{\max} = 2^{-2}$.

3 Results and Analysis

We report results for a single sampling budget with 64 time steps between $t = 0$ and $t = 1$ and no classifier-free guidance ($s = 0$). For each method, we generate $2{,}048$ samples on the EuroSAT validation split and measure FID, effective number of function evaluations (NFE), wall-clock time, and the trajectory metrics described in Sec. 2. The results are summarized in Tab. 1 and Fig. 2.

Quality and efficiency. The top-left panel of Fig. 2 shows that all methods have almost identical cost per evaluation around 0.07s, so total runtime is essentially proportional to effective NFE. The top-right panel therefore directly characterizes the compute–quality trade-off. Euler is cheapest (effective NFE 2048) but has the worst FID (85.8). AB2 operates at a very similar budget (2112 NFE) yet achieves a noticeably better FID (84.7), indicating a more efficient use of evaluations. Midpoint and Heun roughly double the compute (4096 NFE); Midpoint improves FID further to 83.3, while Heun gives only a minor gain over Euler. RK4 is the most expensive (8192 NFE) but does not yield the best FID (84.2), suggesting diminishing returns at this budget. The adaptive RK23 sampler attains the lowest FID (82.8) at an intermediate cost (4524 NFE), indicating that adaptivity can modestly outperform both low-order fixed-step schemes and high-order RK4 for rectified flows.

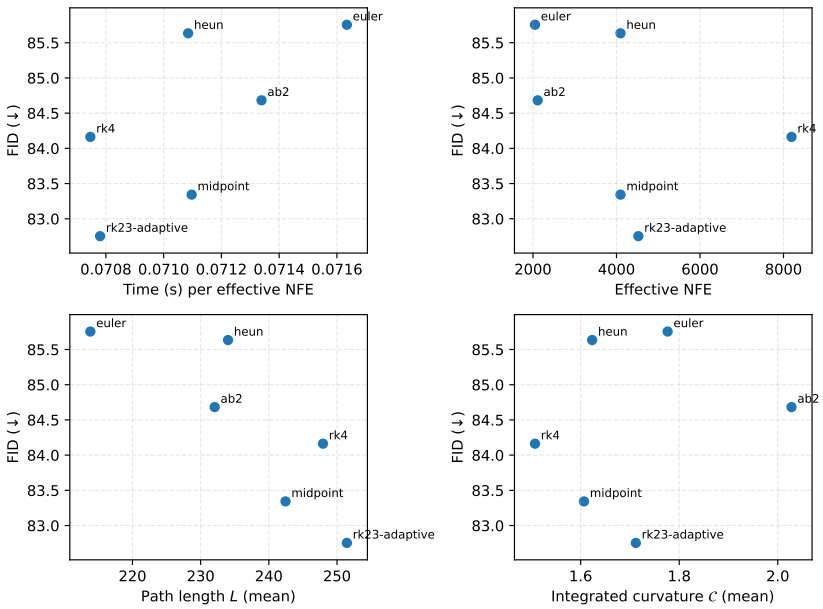


Figure 2: Trade-offs between FID, computation, and geometry for different ODE samplers.

Trajectory geometry. The bottom panels relate FID to geometric properties of the discrete trajectories. FID tends to improve as the mean path length L increases (bottom-left): methods that track longer trajectories (Midpoint, RK4, RK23) generally achieve lower FID than Euler, whose paths are shortest. This is consistent with higher-order or adaptive solvers following the underlying flow more faithfully instead of cutting across it. The relation between FID and integrated curvature $\mathcal{C}$ (bottom-right) is weaker. AB2 has the largest curvature and only moderate FID, as we discuss this as an example in Sec. E of the appendix. RK4 has the lowest curvature but does not outperform RK23 or Midpoint. The best-performing sampler, RK23, exhibits moderate curvature (around 1.7), suggesting that extremely low curvature is not necessary for good samples; instead, a balance between path length and curvature appears sufficient at this discretization.

Overall, at a fixed base of 64 time steps, our experiments indicate that simple Euler steps are suboptimal for rectified flows. Multi-step and higher-order methods such as AB2 and Midpoint provide better FID at comparable or moderately higher cost, and an adaptive solver, RK23, offers the best quality among the tested methods without incurring the very high evaluation count of RK4.

4 Conclusion

We studied how standard ODE solvers behave when applied to a fixed rectified-flow generative model. On conditional RF trained on EuroSAT, we compared Euler, Midpoint, Heun, RK4, Adams–Bashforth 2, and adaptive RK23 in terms of FID, computational cost, and trajectory geometry. At 64 steps and zero guidance, Euler is the cheapest but clearly the worst in FID; AB2 improves FID over Euler at almost identical cost, Midpoint gives the best fixed-step FID at a moderate NFE increase, and RK4 shows diminishing returns. Adaptive RK23 achieves the lowest FID overall with an intermediate evaluation budget, suggesting that adaptivity can exploit the largely straight RF trajectories.

Our geometric diagnostics indicate that methods tracing longer paths with moderate curvature tend to achieve better FID, but curvature alone is not a reliable predictor of quality. The study is limited to one dataset, one RF architecture, a single discretization, and no guidance, so absolute FID values should be read comparatively rather than competitively. Still, our results give simple practical advice: for rectified flows, Euler should be replaced by at least a second-order or multi-step solver, and adaptive schemes such as RK23 offer an appealing quality–efficiency trade-off when extra compute is available.

References

[1] J. Ho, A. Jain, and P. Abbeel (2020). Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33*.

[2] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

[3] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le (2022). Flow Matching for Generative Modeling. *arXiv preprint* arXiv:2210.02747.

[4] X. Liu, C. Gong, and Q. Liu (2023). Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*.

[5] P. Helber, B. Bischke, A. Dengel, and D. Borth (2019). EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

A Teamwork

The project was carried out by three team members. Below we briefly summarize individual responsibilities and how the work came together as a whole. All three authors jointly discussed the research questions, designed the experiments, and contributed to the writing and revision of the report.

Qiwen Xiao. Qiwen initiated the project idea of studying ODE samplers for rectified flows and led the overall technical direction. He implemented the conditional rectified-flow model, the training pipeline, and the sampler API used by all methods. Qiwen also integrated the trajectory geometry metrics (path length and curvature) into the sampling code and coordinated the experiments and weekly meetings. Apart from the main text, he is also in charge of the theoretical insights in the appendix.

Omar Attaalla. Omar focused on the evaluation and analysis aspects. He implemented the FID computation pipeline based on exported EuroSAT validation images, set up the scripts to sweep over samplers and collect metrics (effective NFE, wall time, and geometry statistics), and generated the trade-off plots used in the Results and Analysis section. Omar also helped with debugging and profiling the sampling code.

William Y. Chung. William concentrated on data handling, training stability, and experimental tuning. He prepared the EuroSAT dataset split and preprocessing, helped select and tune model and optimizer hyperparameters, and ran training runs to obtain the baseline rectified-flow checkpoint. William also carried out sanity checks on generated samples, assisted in interpreting the geometry metrics, and contributed text to the Methodology and Conclusion sections.

Joint work. All three team members participated in designing the choice of samplers, metrics, and sampling budget, interpreting the resulting plots, and deciding on the final presentation of the results. The abstract and introduction were written collaboratively, with iterative feedback from all authors during weekly meetings.

B Challenges and Iterative Design

During the course of the project, we went through several iterations on both the modeling and experimental design. We summarize the most relevant issues and how we addressed them, as they motivated the final focus on sampler comparisons and trajectory geometry.

B.1 RF-Div experiments and computational cost

We initially planned to augment rectified flow with an additional divergence regularizer ("RF-Div") based on Hutchinson-type stochastic trace estimates of the Jacobian of the velocity field.

In practice, this approach was prohibitively slow on EuroSAT. Each divergence estimate required multiple forward/backward passes through the U-Net per sample, and the additional computational overhead dominated training time. Even after reducing the number of Hutchinson probes and switching to smaller batch sizes, epochs became too long for a realistic project timeline. As a result, we decided to drop RF-Div from the main experiments and focus on comparing ODE samplers for a fixed rectified-flow model instead. The RF-Div idea is therefore left as future work.

B.2 Architecture and normalization issues

We experimented with shrinking the U-Net to reduce training cost by lowering `base_channels` and adjusting `channel_mult`. This exposed a subtle issue with group normalization: for some configurations, `GroupNorm` raised an error because the number of channels was not divisible by the number of groups. We resolved this by enforcing channel counts that remain divisible by a fixed number of groups and by making the function that chooses `num_groups` explicitly depend on the current channel dimension. After this fix, the architecture remained stable across different width configurations.

We also considered using an exponential moving average (EMA) of model weights for sampling, but in early experiments the benefits were marginal relative to the additional implementation and bookkeeping overhead. For simplicity and transparency, the main results reported in the paper use the raw model checkpoint without EMA.

B.3 Geometry metrics and numerical testing

To study trajectory geometry, we implemented path length L and integrated curvature $\mathcal{C}$ inside the ODE integration routine. To gain confidence in the implementation, we designed synthetic test fields with piecewise constant velocity field and checked that the metrics behaved as expected.

One test case considered a unit-speed flow along a straight line, where the path length over $t \in [0, 1]$ should be exactly $L = 1$ and curvature should vanish. With a Heun solver, the numerical path length initially converged only to approximately 0.97 instead of 1. This discrepancy turned out to be a consequence of using a small number of time steps in the test: the polygonal path is shorter than the true arc length, and the error decreases as the step size is refined. We relaxed the test tolerance and interpreted the remaining discrepancy as standard discretization error, while keeping the test as a sanity check on the scaling and units of L and $\mathcal{C}$.

We also had to be careful with numerical stability when computing curvature: the dot products between successive unit directions are clamped to $[-1 + \varepsilon, 1 - \varepsilon]$ before applying `arccos` to avoid NaNs due to floating-point roundoff.

B.4 Curvature profiles and design of the geometric study

Our theoretical analysis of Adams–Bashforth 2 (AB2) and Euler requires a notion of how curvature varies along the trajectory. Rather than postulating a particular pattern, we implemented an experiment that records the full RF trajectory for a high-resolution solver and estimates a local curvature profile $\kappa(t)$ along $t \in [0, 1]$.

The resulting profile turned out to be U-shaped: curvature is largest near the noise and data endpoints and very small in the middle of the interval. This observation led us to abandon an earlier, overly simplistic assumption of globally non-decreasing curvature and to instead derive a local analysis in terms of the curvature derivative $\kappa'(s)$. The appendix section Sec. E is a direct consequence of this empirical curvature study and represents the theoretical component of our project.

C Generated Samples

In Fig. 1 we show generated samples from the rectified-flow model using different ODE samplers.

D Interpretation of Discrete Path Length

In this section we discuss the relationship between the discrete path length used in our experiments and the continuous arc length of a rectified-flow trajectory. The key message is that, for a fixed smooth trajectory $x(t)$, the sum of chord lengths over a time partition always *underestimates* the true arc length and converges to it as the partition is refined. This helps interpret our empirical finding that samplers with better FID tend to have longer discrete paths.

D.1 Setup and definitions

Let $x : [0, 1] \to \mathbb{R}^d$ be a continuously differentiable curve. In our generative setting, $x(t)$ can be thought of as the exact solution of the ODE $\dot{x}(t) = v_\theta(x(t), t, y)$ with initial condition $x(0) = z$. The continuous arc length of x over $[0, 1]$ is defined as

$$L^\star \;=\; \int_0^1 \big\|\dot{x}(t)\big\|_2 \, dt. \tag{18}$$

Let $0 = t_0 < t_1 < \cdots < t_N = 1$ be any partition of $[0, 1]$. The corresponding discrete path length of x along this partition is

$$L_{\text{disc}}(t_0, \ldots, t_N) \;=\; \sum_{k=0}^{N-1} \big\|x(t_{k+1}) - x(t_k)\big\|_2. \tag{19}$$

In our experiments we use a uniform grid $t_k = kh$ with $h = 1/N$, and we compute the discrete length of the numerical trajectory. For the moment we work with the exact curve $x(t)$.

D.2 Discrete length underestimates arc length

We first show that chord-length sums are always bounded above by the true arc length.

Lemma D.1 (Chord length and arc length)**.** *Let $x : [0, 1] \to \mathbb{R}^d$ be continuously differentiable, and let $0 = t_0 < \cdots < t_N = 1$ be any partition. Then*

$$L_{\text{disc}}(t_0, \ldots, t_N) \;\le\; L^\star \;=\; \int_0^1 \big\|\dot{x}(t)\big\|_2 \, dt. \tag{20}$$

Proof. Consider a single segment $[t_k, t_{k+1}]$. By the fundamental theorem of calculus,

$$x(t_{k+1}) - x(t_k) \;=\; \int_{t_k}^{t_{k+1}} \dot{x}(t) \, dt. \tag{21}$$

Taking norms and applying the triangle inequality for integrals,

$$\big\|x(t_{k+1}) - x(t_k)\big\|_2 \;=\; \left\|\int_{t_k}^{t_{k+1}} \dot{x}(t) \, dt\right\|_2 \;\le\; \int_{t_k}^{t_{k+1}} \big\|\dot{x}(t)\big\|_2 \, dt. \tag{22}$$

Summing over $k = 0, \ldots, N - 1$, we obtain

$$\sum_{k=0}^{N-1} \big\|x(t_{k+1}) - x(t_k)\big\|_2 \;\le\; \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \big\|\dot{x}(t)\big\|_2 \, dt \;=\; \int_0^1 \big\|\dot{x}(t)\big\|_2 \, dt \;=\; L^\star. \tag{23}$$

This is precisely $L_{\text{disc}}(t_0, \ldots, t_N) \le L^\star$. □

Thus, for any choice of time grid, the polygonal path that connects the points $\{x(t_k)\}$ is always shorter than or equal to the true curve $x(t)$.

D.3 Convergence as the grid is refined

We now show that as the partition is refined, the discrete path length converges to the arc length.

Theorem D.2 (Convergence of discrete length)**.** *Let $x : [0,1] \to \mathbb{R}^d$ be continuously differentiable. For each $N \in \mathbb{N}$, consider the uniform partition $t_k^{(N)} = k/N$ and define*

$$L_N \;=\; \sum_{k=0}^{N-1} \big\|x(t_{k+1}^{(N)}) - x(t_k^{(N)})\big\|_2. \tag{24}$$

Then

$$\lim_{N\to\infty} L_N \;=\; L^\star. \tag{25}$$

Proof. By Lemma D.1 we already know that $L_N \le L^\star$ for all N, so it remains to show that $\liminf_{N\to\infty} L_N \ge L^\star$.

Fix $\varepsilon > 0$. Because $\dot{x}$ is continuous on the compact interval $[0,1]$, it is uniformly continuous. Hence there exists $\delta > 0$ such that for any $s, t \in [0,1]$ with $|s-t| \le \delta$,

$$\big\|\dot{x}(s) - \dot{x}(t)\big\|_2 \;\le\; \varepsilon. \tag{26}$$

Choose N large enough that the uniform step size $h = 1/N$ satisfies $h \le \delta$. For each k, pick a point $\tau_k \in [t_k^{(N)}, t_{k+1}^{(N)}]$ (the choice does not matter, e.g. the midpoint), and write

$$x(t_{k+1}^{(N)}) - x(t_k^{(N)}) \;=\; \int_{t_k^{(N)}}^{t_{k+1}^{(N)}} \dot{x}(t)\,dt \;=\; \int_{t_k^{(N)}}^{t_{k+1}^{(N)}} \big(\dot{x}(\tau_k) + (\dot{x}(t) - \dot{x}(\tau_k))\big)dt. \tag{27}$$

Taking norms and using the triangle inequality,

$$\big\|x(t_{k+1}^{(N)}) - x(t_k^{(N)})\big\|_2 \ge \left\|\int_{t_k^{(N)}}^{t_{k+1}^{(N)}} \dot{x}(\tau_k)\,dt\right\|_2 \;-\; \left\|\int_{t_k^{(N)}}^{t_{k+1}^{(N)}} (\dot{x}(t) - \dot{x}(\tau_k))\,dt\right\|_2 \tag{28}$$

$$= h\big\|\dot{x}(\tau_k)\big\|_2 \;-\; \int_{t_k^{(N)}}^{t_{k+1}^{(N)}} \big\|\dot{x}(t) - \dot{x}(\tau_k)\big\|_2\,dt. \tag{29}$$

By the choice of δ and $h \le \delta$, we have $\|\dot{x}(t) - \dot{x}(\tau_k)\|_2 \le \varepsilon$ whenever $t \in [t_k^{(N)}, t_{k+1}^{(N)}]$, so

$$\big\|x(t_{k+1}^{(N)}) - x(t_k^{(N)})\big\|_2 \;\ge\; h\big\|\dot{x}(\tau_k)\big\|_2 - h\varepsilon. \tag{30}$$

Summing over k,

$$L_N = \sum_{k=0}^{N-1} \big\|x(t_{k+1}^{(N)}) - x(t_k^{(N)})\big\|_2 \tag{31}$$

$$\ge \sum_{k=0}^{N-1} h\big\|\dot{x}(\tau_k)\big\|_2 \;-\; \sum_{k=0}^{N-1} h\varepsilon \tag{32}$$

$$= \sum_{k=0}^{N-1} h\big\|\dot{x}(\tau_k)\big\|_2 \;-\; \varepsilon. \tag{33}$$

The sum $\sum_k h\|\dot{x}(\tau_k)\|_2$ is a Riemann sum for the integral $\int_0^1 \|\dot{x}(t)\|_2 dt$. As $N \to \infty$ (and $h \to 0$), the Riemann sums converge to the integral, hence there exists N_0 such that for all $N \ge N_0$,

$$\left|\sum_{k=0}^{N-1} h\big\|\dot{x}(\tau_k)\big\|_2 \;-\; \int_0^1 \big\|\dot{x}(t)\big\|_2\,dt\right| \;\le\; \varepsilon. \tag{34}$$

Combining with the previous inequality, we obtain for all $N \ge N_0$,

$$L_N \;\ge\; \int_0^1 \big\|\dot{x}(t)\big\|_2\,dt - \varepsilon - \varepsilon \;=\; L^\star - 2\varepsilon. \tag{35}$$

Since $\varepsilon > 0$ was arbitrary, this shows $\liminf_{N\to\infty} L_N \ge L^\star$. Together with $L_N \le L^\star$ from Lemma D.1, we conclude $\lim_{N\to\infty} L_N = L^\star$. □

D.4 Implications for our experiments

The above results apply to a fixed smooth trajectory $x(t)$ whose positions $x(t_k)$ are exactly known. In practice, our discrete trajectories are produced by numerical samplers: starting from $x_0 = z$, each method generates a sequence $\{x_k^{(\text{method})}\}$ that approximates the unknown exact solution at times $t_k = k/N$.

If a sampler is low order such as Euler, it tends to cut corners, as its discrete trajectory deviates from the true curve and can be significantly shorter than the arc length of the underlying rectified flow. Higher-order or adaptive methods such as Midpoint, AB2, or RK23 track the continuous dynamics more faithfully, so their discrete path lengths move closer to the true arc length $L^\star$, which is typically larger than the length obtained by coarse Euler. In other words, for a fixed ODE, better samplers should have discrete lengths L_{disc} that are closer to the continuous length $L^\star$.

This perspective explains our empirical observation that, at 64 steps, methods with lower FID (Midpoint, AB2, RK23) tend to exhibit longer discrete paths than Euler: a larger L_{disc} in our setting is better interpreted as a proxy for how well the sampler follows the rectified-flow trajectory, rather than as a sign of a less straight underlying flow.

E Why AB2 Exhibits Higher Curvature than Euler

In this section we compare the discrete curvature induced by the Euler and Adams–Bashforth 2 (AB2) methods when they discretize a smooth trajectory. The key point is that AB2 effectively evaluates the tangent slightly further along the curve than Euler; as a result, it amplifies turning in regions where curvature increases along the path. This local effect, combined with the curvature profile of our rectified-flow trajectories, explains why AB2 exhibits larger integrated discrete curvature than Euler in our experiments.

E.1 Geometric setup

Let $x : [0, S] \to \mathbb{R}^d$ be a C^3 curve parametrized by arc length, so that

$$\big\|x'(s)\big\|_2 = 1 \quad \text{for all } s \in [0, S].$$

The unit tangent is

$$T(s) = x'(s),$$

and the curvature is

$$\kappa(s) = \big\|T'(s)\big\|_2.$$

Informally, $\kappa(s)$ measures how quickly the direction of the curve changes per unit arc length.

We consider a uniform grid $s_k = kh$ with $h = S/N$, and interpret the ODE

$$\frac{dx}{ds} = T(s)$$

as the idealized rectified-flow dynamics. Applying a numerical method to this ODE generates a polygonal path whose per-step turning angles we can compare to the underlying curvature.

E.2 Step directions for Euler and AB2

For each k, let x_k denote the numerical approximation of $x(s_k)$, and define the step direction

$$u_k = \frac{x_{k+1} - x_k}{\|x_{k+1} - x_k\|_2}.$$

The turning angle between two successive steps is

$$\theta_k = \arccos\big(\langle u_k, u_{k+1}\rangle\big),$$

and the integrated discrete curvature of the polygonal path is $\mathcal{C}_{\text{disc}} = \sum_k \theta_k$.

Euler method. The explicit Euler method with step size h is

$$x_{k+1}^{\mathrm{E}} = x_k^{\mathrm{E}} + h\,T(s_k).$$

Thus the step direction is exactly the tangent at the left endpoint:

$$u_k^{\mathrm{E}} = \frac{h\,T(s_k)}{\|h\,T(s_k)\|_2} = T(s_k).$$

AB2 method. After one initial step, the two-step Adams–Bashforth method updates via

$$x_{k+1}^{\mathrm{AB2}} = x_k^{\mathrm{AB2}} + h\left(\tfrac{3}{2}T(s_k) - \tfrac{1}{2}T(s_{k-1})\right).$$

Define the unnormalized step

$$g_k = \tfrac{3}{2}T(s_k) - \tfrac{1}{2}T(s_{k-1}),$$

so that $x_{k+1}^{\mathrm{AB2}} - x_k^{\mathrm{AB2}} = hg_k$, and

$$u_k^{\mathrm{AB2}} = \frac{g_k}{\|g_k\|_2}.$$

The following lemma shows that, for small h, u_k^{AB2} is aligned with the tangent at a point shifted forward by approximately $h/2$.

Lemma E.1 (AB2 uses a forward-shifted tangent)**.** *Assume x is C^3 and parametrized by arc length. Then, as $h \to 0$,*

$$u_k^{\mathrm{AB2}} = T\big(s_k + \tfrac{h}{2}\big) + \mathcal{O}(h^2),$$

where the $\mathcal{O}(h^2)$ term is uniform in k on compact intervals.

Proof. Using a Taylor expansion of T around s_k,

$$T(s_{k-1}) = T(s_k - h) = T(s_k) - hT'(s_k) + \tfrac{1}{2}h^2T''(s_k) + \mathcal{O}(h^3).$$

Substituting into g_k gives

$$\begin{aligned} g_k &= \tfrac{3}{2}T(s_k) - \tfrac{1}{2}T(s_{k-1}) \\ &= \tfrac{3}{2}T(s_k) - \tfrac{1}{2}\Big[T(s_k) - hT'(s_k) + \tfrac{1}{2}h^2T''(s_k) + \mathcal{O}(h^3)\Big] \\ &= T(s_k) + \tfrac{h}{2}T'(s_k) - \tfrac{1}{4}h^2T''(s_k) + \mathcal{O}(h^3). \end{aligned}$$

On the other hand, expanding T at $s_k + \frac{h}{2}$ yields

$$T\big(s_k + \tfrac{h}{2}\big) = T(s_k) + \tfrac{h}{2}T'(s_k) + \tfrac{1}{8}h^2T''(s_k) + \mathcal{O}(h^3).$$

Therefore

$$g_k = T\big(s_k + \tfrac{h}{2}\big) - \tfrac{3}{8}h^2T''(s_k) + \mathcal{O}(h^3),$$

so g_k and $T(s_k + h/2)$ differ by $\mathcal{O}(h^2)$. Since $\|T(s)\|_2 = 1$ and T is smooth, we have $\|g_k\|_2 = 1 + \mathcal{O}(h^2)$, hence normalizing g_k does not change its direction beyond $\mathcal{O}(h^2)$:

$$u_k^{\mathrm{AB2}} = \frac{g_k}{\|g_k\|_2} = T\big(s_k + \tfrac{h}{2}\big) + \mathcal{O}(h^2).$$

□

Thus Euler aligns its steps with $T(s_k)$, while AB2 aligns to leading order with $T(s_k + h/2)$, as if AB2 effectively looks ahead along the curve.

E.3 Local turning angles and curvature

The turning angle between two unit tangents separated by a small arc-length increment Δs is approximately the integral of curvature over that interval. More precisely, if $\theta(s)$ denotes the orientation of the tangent in the planar case, then $\theta'(s) = \kappa(s)$ and

$$\angle\big(T(s), T(s + \Delta s)\big) = \theta(s + \Delta s) - \theta(s) = \int_s^{s+\Delta s} \kappa(r)\,dr.$$

In higher dimensions one can define a similar notion using the Frenet frame; the leading-order dependence on $\int \kappa$ is the same.

Using this observation together with Lemma E.1, we can express the per-step turning angles for Euler and AB2 as integrals of curvature over slightly different arc-length intervals.

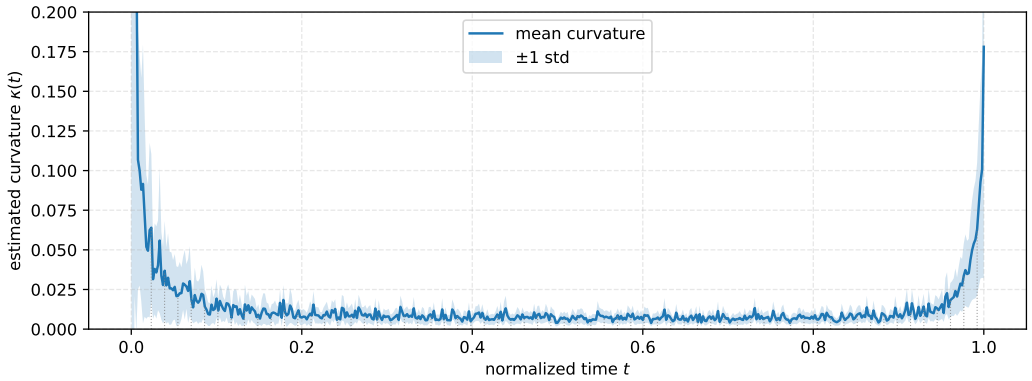


Figure 3: Estimated curvature profile along rectified-flow trajectories, using Euler with 512 steps as a high-accuracy proxy. The curve shows the batch-mean local curvature $\bar{\kappa}(t)$, with a shaded region indicating ± 1 standard deviation.

Euler. Since $u_k^{\mathrm{E}} = T(s_k)$ and $u_{k+1}^{\mathrm{E}} = T(s_{k+1})$, the Euler turning angle is

$$\theta_k^{\mathrm{E}} = \angle\big(T(s_k), T(s_{k+1})\big) \approx \int_{s_k}^{s_k+h} \kappa(r)\, dr. \tag{36}$$

AB2. By Lemma E.1, the AB2 step directions satisfy

$$u_k^{\mathrm{AB2}} = T\big(s_k + \tfrac{h}{2}\big) + \mathcal{O}(h^2), \qquad u_{k+1}^{\mathrm{AB2}} = T\big(s_{k+1} + \tfrac{h}{2}\big) = T\big(s_k + \tfrac{3h}{2}\big) + \mathcal{O}(h^2).$$

Hence the AB2 turning angle is

$$\theta_k^{\mathrm{AB2}} = \angle\Big(T\big(s_k + \tfrac{h}{2}\big), T\big(s_k + \tfrac{3h}{2}\big)\Big) \approx \int_{s_k+\frac{h}{2}}^{s_k+\frac{3h}{2}} \kappa(r)\, dr. \tag{37}$$

Euler and AB2 thus integrate curvature over two intervals of equal length h, but located at different positions along the curve.

E.4 Dependence on the derivative of curvature

To compare (36) and (37), we approximate κ locally by a linear function around s_k:

$$\kappa(r) \approx \kappa_0 + \kappa_1(r - s_k), \quad \kappa_0 = \kappa(s_k), \quad \kappa_1 = \kappa'(s_k),$$

with higher-order terms of order $\mathcal{O}((r - s_k)^2)$.

Substituting this expansion into (36) gives

$$\theta_k^{\mathrm{E}} \approx \int_{s_k}^{s_k+h} \big(\kappa_0 + \kappa_1(r - s_k)\big)\, dr \tag{38}$$

$$= \kappa_0 h + \tfrac{1}{2}\kappa_1 h^2 + \mathcal{O}(h^3). \tag{39}$$

For AB2, using the same linear approximation in (37),

$$\theta_k^{\mathrm{AB2}} \approx \int_{s_k+\frac{h}{2}}^{s_k+\frac{3h}{2}} \big(\kappa_0 + \kappa_1(r - s_k)\big)\, dr \tag{40}$$

$$= \kappa_0 h + \kappa_1 h^2 + \mathcal{O}(h^3). \tag{41}$$

Subtracting (39) from (41) yields the leading-order relation

$$\theta_k^{\mathrm{AB2}} - \theta_k^{\mathrm{E}} = \tfrac{1}{2}\,\kappa'(s_k) h^2 + \mathcal{O}(h^3). \tag{42}$$

Proposition 1 (Local comparison of Euler and AB2 curvature)**.** *For sufficiently small step size h, the difference between the per-step turning angles of AB2 and Euler satisfies*

$$\theta_k^{\mathrm{AB2}} - \theta_k^{\mathrm{E}} = \tfrac{1}{2}\,\kappa'(s_k)h^2 + \mathcal{O}(h^3).$$

In particular, to leading order:

- *if $\kappa'(s_k) > 0$ (curvature increasing locally), then $\theta_k^{\mathrm{AB2}} > \theta_k^{\mathrm{E}}$;*
- *if $\kappa'(s_k) < 0$ (curvature decreasing locally), then $\theta_k^{\mathrm{AB2}} < \theta_k^{\mathrm{E}}$;*
- *if $\kappa'(s_k) \approx 0$, then the two angles coincide up to higher-order terms.*

Thus AB2 does not universally produce larger curvature than Euler; instead, it amplifies curvature in regions where the true curvature is increasing along the trajectory and attenuates it where curvature is decreasing.

E.5 Implications for our rectified-flow trajectories

Fig. 3 shows the empirical curvature profile of our rectified-flow model estimated from high-resolution Euler trajectories with 512 steps. The diagram is U-shaped, where curvature is large and decreasing near the noise endpoint, very small and nearly flat through most of the interval, and rises sharply near the data endpoint. In light of Proposition 1, this profile suggests the following:

- In the initial segment, where $\kappa'(s) < 0$, Euler may exhibit slightly larger turning than AB2.
- In the middle region, where $\kappa'(s) \approx 0$, Euler and AB2 have very similar turning angles.
- In the final segment, where curvature increases rapidly and $\kappa'(s) > 0$, AB2 accumulates larger turning than Euler because it evaluates an extrapolated tangent that looks ahead into this high-curvature region.

The integrated discrete curvature $\mathcal{C}_{\mathrm{disc}}$ is obtained by summing these local contributions over the entire path. In our experiments, the strong increase of curvature near the data endpoint dominates the initial decrease, so the net effect is that AB2 exhibits higher total discrete curvature than Euler at 64 steps, as observed in Table 1. This behavior is a direct consequence of how AB2 uses forward-looking information about the tangent field.

F Pseudocode for Sampling and Geometry Metrics

Algorithm 1 Sampling with geometric diagnostics

Input: sampler Step, RF vector field v_θ, label y, number of steps K
Sample $z_0 \sim \mathcal{N}(0, I)$; set $x_0 \leftarrow z_0$, $t_0 \leftarrow 0$
Initialize path length $L \leftarrow 0$, curvature $\mathcal{C} \leftarrow 0$
for $k = 0$ to $K - 1$ **do**
  $(x_{k+1}, t_{k+1}) \leftarrow \mathrm{Step}(v_\theta, x_k, t_k, y)$
  $\Delta x_k \leftarrow x_{k+1} - x_k, \quad \ell_k \leftarrow \|\Delta x_k\|_2$
  $L \leftarrow L + \ell_k$
  if $k > 0$ **then**
    $u_{k-1} \leftarrow \Delta x_{k-1} / \|\Delta x_{k-1}\|_2$
    $u_k \leftarrow \Delta x_k / \|\Delta x_k\|_2$
    $\theta_k \leftarrow \arccos(\langle u_{k-1}, u_k \rangle)$
    $\mathcal{C} \leftarrow \mathcal{C} + \theta_k$
  end if
end for
return $x_K, L, \mathcal{C}$