SHARPEN YOUR FLOW: SHARPNESS AWARE SAMPLING FOR FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching models generate samples by numerically integrating a learned velocity field, with each integration step requiring a neural network evaluation. Fast generation therefore requires using a small fixed evaluation budget effectively: the key question is not only how to integrate the flow, but where the sampler should spend its steps. We propose SharpEuler, a training-free sampler that profiles a pretrained model offline by estimating where the learned velocity field changes most rapidly along calibration trajectories. This finite-difference estimate defines a solver-aware sharpness profile, which is smoothed and converted by a quantile transform into a timestep grid for any desired inference budget. At test time, sampling remains ordinary Euler integration with the same number of model evaluations as a uniform schedule. We justify SharpEuler using three principles: a numerical principle identifying trajectory acceleration as the leading source of Euler discretization error, a variational principle deriving sharpness-based power-law timestep densities, and a statistical guarantee showing that the finite-sample calibrated sampler is stable at the terminal distribution level. Our experiments show that SharpEuler improves sample quality at fixed budgets, reducing inter-mode leakage and increasing mode coverage.

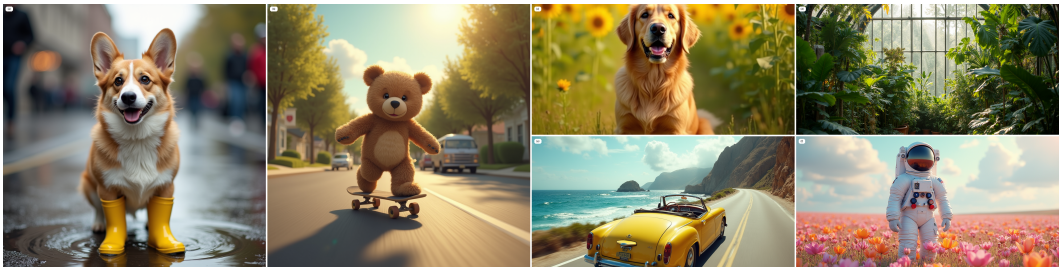


Figure 1: **Few-step text-to-image generation with SharpEuler.** Samples from the pretrained FLUX.1-dev model using only 12 network evaluations. SharpEuler uses ordinary Euler integration, but places its timesteps according to an offline-calibrated trajectory-sharpness profile.

1 INTRODUCTION

Flow-based generative models have become a widely used tool for modeling high-dimensional distributions (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2023). Their applications range from text-to-image (Esser et al., 2024) and text-to-video generation (Jin et al.) to scientific domains such as time series modeling (Hu et al., 2024; Lim et al., 2026), fluid flow forecasting (Lim et al., 2024), molecule generation (Morehead et al., 2026), and materials design (Sriram et al., 2024).As an instance of dynamical measure transport, sampling with a flow matching model requires integrating a learned velocity field that transports samples from a simple noise distribution to the data distribution.

This makes sampling a numerical integration problem. In classical numerical analysis, accuracy is often improved using higher-order or adaptive solvers (Hairer et al., 1993; Griffiths & Higham, 2010; Butcher, 2016). For neural generative models, however, each function evaluation is a forward pass through a large network (Lipman et al., 2024), and these evaluations dominate sampling cost in modern image and video models (Esser et al., 2024). Thus the relevant quantity is not only asymptotic

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.

solver accuracy, but sample quality per neural network evaluation. For this reason, many practical samplers use simple one-stage methods with a fixed budget. Forward Euler is attractive because a sampler with budget B uses exactly B model evaluations. If we do not add solver stages or use an online adaptive method with sample-dependent cost, the remaining question is then:

Where should a flow matching sampler spend its finite Euler budget?

Uniform and shifted schedules are simple and often effective, but they are not derived from the learned dynamics being integrated. Recent work has shown that timestep allocation can strongly impact sample quality, especially in the few-evaluation regime (Watson et al., 2021; Xue et al., 2024; Williams et al., 2024; Sabour et al., 2024). Other schedulers use information-theoretic criteria, such as entropy-based time reparameterizations that motivates the Entropic Time Scheduler (ETS) of (Stancevic et al., 2025). See also App. A for a detailed discussion of related work.

We propose SharpEuler, a training-free, solver-aware schedule calibration method for pretrained flow matching models. The method estimates where the learned trajectory is sharp (velocity field changes rapidly) rather than choosing this region by hand. In an offline calibration stage, we run fine-grained Euler trajectories on a set of representative prompts or samples, record consecutive velocity evaluations, and estimate how rapidly the vector field changes along the trajectory. Averaging and smoothing the finite-difference estimated accelerations gives a time-dependent sharpness profile. At inference time, we convert this profile into a monotone grid and run the same Euler sampler on that grid. The method does not change the model, require retraining, add solver stages, or increase the number of model evaluations. It only changes where the evaluations are placed.

SharpEuler is motivated by the local truncation error (LTE) of Euler integration along a flow trajectory. Euler's local truncation error is controlled by trajectory acceleration: regions where the learned velocity changes rapidly require smaller steps, while nearly linear regions can be traversed more coarsely. The same signal appears in backward error analysis (Hairer et al., 2006) as the leading modified-equation bias. Thus, rather than placing steps uniformly or by hand, SharpEuler estimates where the trajectory acceleration is large and places more evaluations in those regions. This perspective also explains why the method is most useful at small sampling budgets. With many evaluations, even a uniform grid can resolve most changes in the trajectory. However, with only a few evaluations, a poorly placed Euler step can cross a sharp region without evaluating the intervening dynamics, which can mix modes or leave parts of the data distribution underrepresented. We observe this behavior both on synthetic distributions, where mode coverage and leakage can be measured directly, and in high-dimensional text-to-image sampling, where calibrated timesteps improve semantic fidelity.

Contributions. Our main contributions are as follows.

- We propose SharpEuler, a training-free, solver-aware method for constructing timestep schedules for pretrained flow matching models. It uses offline calibration trajectories to estimate a sharpness profile and converts this profile into a fixed Euler grid for any inference budget.
- We provide three principles to justify SharpEuler. The *numerical* principle shows, via local truncation error and backward error analysis, that acceleration controls both the leading Euler error and the leading modified-equation bias. The *variational* principle shows that natural timestep allocation objectives yield power-law densities (Proposition 2), with the square-root case singled out by minimizing the leading Euler error proxy. The *statistical* principle gives a stability guarantee (Theorem 2), showing that the calibrated schedule approaches the best population schedule in the admissible class as the number of calibration trajectories grows, up to smoothing bias.
- We demonstrate that SharpEuler improves low-budget sampling without increasing the number of model evaluations. On synthetic multimodal data, it reduces inter-mode leakage, improves mode coverage and sample quality, especially at very small Euler budgets. We then demonstrate its advantage on high-dimensional text-to-image generation tasks.

2 SharpEuler: A Sharpness Aware Sampler

Background. Flow matching models define a continuous transformation from a simple noise distribution to the data distribution through an ODE driven by a velocity field (Lipman et al., 2024):

$$\frac{dx(t)}{dt} = u_\theta(x(t), t), \qquad x(0) \sim p_0, \qquad x(t) \in \mathbb{R}^d, \qquad t \in [0, 1]. \tag{1}$$

We start from a base distribution $p_0 = \mathcal{N}(0, I)$ at $t = 0$ and the ODE is integrated to $t = 1$ to obtain new samples. Let $T(x_0) := x(1; x_0)$ and $\mu_1 := T_\# p_0$ denote the learned flow map and its output distribution. Equivalently, we can use the reverse-time coordinate $s = 1 - t$ so that $s = 1$ corresponds to noise and $s = 0$ corresponds to data. If the learned velocity field is denoted by $v_\theta(x, s)$, then the forward-time velocity in eq. (1) is $u_\theta(x, t) = -v_\theta(x, 1 - t)$.

In practice, the ODE in eq. (1) cannot be solved exactly and must be discretized. Let $0 = t_0 < t_1 < \cdots < t_B = 1$ be an ascending forward-time schedule with B intervals. The Euler update is

$$x^{(k+1)} = x^{(k)} + (t_{k+1} - t_k) u_\theta(x^{(k)}, t_k), \qquad k = 0, \ldots, B - 1. \tag{2}$$

In the reverse-time implementation that we use, we consider the descending grid $s_k = 1 - t_k$, $1 = s_0 > s_1 > \cdots > s_B = 0$, and eq. (2) becomes

$$x^{(k+1)} = x^{(k)} - (s_k - s_{k+1}) v_\theta(x^{(k)}, s_k), \qquad k = 0, \ldots, B - 1. \tag{3}$$

The sampler performs exactly one evaluation of the vector field per interval, and therefore uses exactly B model evaluations. The integer B is often small in fast generation settings, because each evaluation of v_θ is a forward pass through a neural network. Once the model, solver, and budget B are fixed, the main remaining design choice is the time grid itself. A uniform schedule sets $s_k = 1 - k/B$, while other schedules redistribute the same number of evaluations toward selected parts of the trajectory. All such methods have the same online cost when they use the same value of B; they differ only in where the model evaluations are placed. This leads to the following problem setup.

Problem setup. Given a pretrained flow matching model v_θ and a fixed evaluation budget B, construct a monotone time grid $\mathcal{T}_B = \{s_k\}_{k=0}^B$, $1 = s_0 > s_1 > \cdots > s_B = 0$, that improves Euler sampling without changing the model, retraining, adding solver stages, or increasing the number of function evaluations (NFE). In the next section, we propose to choose this grid by measuring where the learned flow is sharp, and to allocate the fixed Euler budget accordingly.

SharpEuler. Given a pretrained flow matching model and an evaluation budget B, our goal is to construct a deterministic time grid for Euler sampling that allocates more model evaluations to regions where the learned dynamics are sharp, and fewer evaluations to regions where they are smooth. In view of this, we introduce *SharpEuler*, a sharpness-aware sampler. Its construction has two stages:

(i) **Offline stage.** We estimate a sharpness profile by computing fine-grained Euler trajectories on a small calibration set and measuring how rapidly the learned vector field changes along these trajectories. This stage is performed only once for a given model and calibration distribution.

(ii) **Online stage.** We convert this profile into a monotone grid and run Euler sampler on that grid, requiring the same number of model evaluations as a uniform Euler sampler with budget B.

Detailed algorithms for the two stages are given in App. B. The motivation is numerical: Euler's method is accurate when the trajectory is nearly linear over a step, but inaccurate when the trajectory changes rapidly. Thus a fixed step size may be harmless in smooth regions and too coarse in sharp ones, suggesting that the time grid should depend on the learned dynamics.

For the reverse-time ODE, the leading local error is controlled by the trajectory acceleration $\ddot{x}(t) = \frac{d}{dt} v_\theta(x(t), t)$. This derivative is taken along the sampled trajectory, so it reflects both explicit time variation in v_θ and variation induced as the state moves different velocity. Computing this quantity accurately is expensive for large neural networks, while classical adaptive error control would require additional model evaluations during sampling. We therefore estimate sharpness offline using finite differences of consecutive velocity evaluations along fine-grained Euler trajectories. This proxy measures where the learned trajectory changes direction or speed rapidly, and hence where a coarse Euler step is most likely to incur error. The resulting sharpness profile is model-, data-, and conditioning-distribution dependent. It is not an online error estimate for a particular sample, but an offline summary of where the velocity field of the pretrained model changes rapidly.

(i) Offline calibration. The offline stage turns the numerical intuition above into a computable signal. We would like to know, as a function of time, where the velocity field changes rapidly along typical sampling trajectories. Since we want a fixed-cost sampler, we estimate this information once from a calibration set and then reuse the resulting profile during inference. Let $\mathcal{C} = \{c_j\}_{j=1}^{M}$ be a calibration set of representative conditioning inputs, prompts, or samples. For unconditional models, we omit c_j from the notation. We choose a fine-grained descending reverse-time grid $1 = s_0 > s_1 > \cdots > s_N = 0$, where N is much larger than the evaluation budgets used at inference time. For each calibration input c_j, we run an N-step Euler trajectory and record the reverse-time velocity evaluations $V_{j,i} = v_\theta(x_{j,i}, s_i; c_j),\ i = 0, \ldots, N-1$. The reference trajectory itself is generated by $x_{j,i+1} = x_{j,i} - (s_i - s_{i+1})V_{j,i}$. These evaluations are already produced by the reference trajectory. We use their successive differences to estimate how rapidly the learned vector field changes along the path. For $i = 0, \ldots, N-2$, define

$$\widehat{a}_{j,i} = \frac{V_{j,i+1} - V_{j,i}}{d_i}, \qquad d_i = \frac{1}{2}\left(|s_i - s_{i+1}| + |s_{i+1} - s_{i+2}|\right) = \frac{1}{2}(s_i - s_{i+2}). \tag{4}$$

This is a finite difference approximation to the derivative of the velocity along the trajectory. The sign depends on the direction in which time is parameterized, but its norm does not. We therefore use $\|\widehat{a}_{j,i}\|$ as the local sharpness signal. Averaging over the calibration set gives a global profile $\bar{\mathcal{S}}_i = \frac{1}{M}\sum_{j=1}^{M} \|\widehat{a}_{j,i}\|,\ i = 0, \ldots, N-2$. This averaging is important. We do not want to adapt the sampler separately for each generated sample, since that would introduce online overhead and variable cost. Instead, we estimate a reusable profile for the model and calibration distribution. In this sense, the schedule is data informed but fixed at inference time.

Motivated by variational principles (see Sec. 3.2), we form the exponent-shaped sharpness signal

$$I_{\gamma,i} = \left(\bar{\mathcal{S}}_i + \varepsilon_a\right)^{\gamma}, \qquad \gamma > 0, \qquad i = 0, \ldots, N-2, \tag{5}$$

where $\varepsilon_a \geq 0$ is a small numerical floor. The raw profile can be noisy, especially for small calibration sets. We therefore smooth it with a Gaussian kernel of bandwidth σ, i.e. $\widetilde{I}_\gamma = \mathrm{Smooth}_\sigma\left(I_\gamma\right)$. We then normalize the smoothed values into discrete masses $\widehat{w}_i = \frac{\widetilde{I}_{\gamma,i}}{\sum_{r=0}^{N-2}\widetilde{I}_{\gamma,r}},\ i = 0, \ldots, N-2$. The support of this finite-difference signal is naturally centered at $\widetilde{s}_i = \frac{s_i + s_{i+2}}{2},\ i = 0, \ldots, N-2$. To be consistent with the forward-time notation, we set $\widetilde{t}_i = 1 - \widetilde{s}_i$. The pairs $(\widetilde{t}_i, \widehat{w}_i)$ are sorted in ascending $\widetilde{t}_i$, an empirical CDF is formed with a left endpoint anchor at $t = 0$, and the inverse CDF is evaluated by linear interpolation. This produces a deterministic schedule for any evaluation budget.

(ii) Online sampling. At inference time, the sampler is ordinary Euler integration on the calibrated grid. Let F_{emp}^{-1} denote the linearly interpolated inverse-CDF map obtained from the offline stage. For a budget B, we compute ascending forward-time quantiles $\widetilde{t}_b = F_{\mathrm{emp}}^{-1}\left(\frac{b}{B}\right),\ b = 0, \ldots, B$, and convert them to a descending reverse-time schedule $s_b = 1 - \widetilde{t}_b,\ b = 0, \ldots, B$. Given a conditioning input c and an initial noise sample $x^{(0)} \sim \mathcal{N}(0, I)$, we iterate $x^{(k+1)} = x^{(k)} - (s_k - s_{k+1})v_\theta(x^{(k)}, s_k; c),\ k = 0, \ldots, B-1$. The only difference from uniform sampling is the choice of the time grid $\{s_k\}$. The model is unchanged, no gradients are computed, and no online error estimates are used. The number of model evaluations is exactly B. The calibrated schedule can be viewed as a time reparameterization (Stancevic et al., 2025): the empirical sharpness profile defines a density over forward model time, and the sampler uses uniform quantiles in the corresponding CDF.

This distinction from classical adaptive solvers is important. Classical adaptive methods choose step sizes during integration and usually spend additional function evaluations to estimate or control local error. Sharpness aware sampling performs this adaptation once, offline, using representative trajectories. The resulting schedule is fixed before inference begins. Thus, the online cost remains deterministic and identical to that of a uniform Euler sampler with the same budget.

3 Why SharpEuler: Three Justifying Principles

Throughout this section, we use the forward-time convention for clarity of analysis, and assume that the learned velocity field $u_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is sufficiently smooth on the region visited by sampled trajectories so that the displayed derivatives and Taylor expansions are well defined. Let $x(t; x_0)$ solve the forward-time ODE eq. (1) and define the acceleration sharpness $a(t; x_0) = \|\ddot{x}(t; x_0)\|$.

3.1 Numerical principle: acceleration controls Euler error and modified-equation bias

Principle I: The acceleration magnitude $\|\ddot{x}(t)\|$ is the quantity selected by both Euler local truncation error and backward error analysis.

The first justification comes from local truncation error.

Proposition 1 (Informal: Euler error selects acceleration)**.** *For the forward-time ODE eq.* (1)*, the leading local truncation error of one Euler step from the current state $x(t)$ is proportional to $h^2\ddot{x}(t)$. Hence, to leading order, the size of the one-step Euler error is controlled by $h^2\|\ddot{x}(t)\|$.*

See Proposition 3 for a detailed version. Indeed, Taylor expansion gives

$$x(t+h) - [x(t) + hu_\theta(x(t), t)] = \frac{h^2}{2}\ddot{x}(t) + \mathcal{O}(h^3) \tag{6}$$

as $h \to 0$, showing that the leading local error is $h^2\|\ddot{x}(t)\|/2$. Moreover, using the chain rule, $\ddot{x}(t) = \frac{d}{dt}u_\theta(x(t), t) = \partial_t u_\theta(x(t), t) + \nabla_x u_\theta(x(t), t)u_\theta(x(t), t)$. Therefore, finite differences of learned velocities along calibration trajectories provide a first-order local-error proxy for $a(t; x_0)$, i.e. $\frac{u_\theta(x(t_{k+1}),t_{k+1}) - u_\theta(x(t_k),t_k)}{\Delta t} = \ddot{x}(t_k) + \mathcal{O}(\Delta t)$. This motivates using acceleration as a solver-aware timestep allocation signal. Full derivations and the backward error analysis are in App. C.

3.2 Variational principle: power-law densities arise from optimal allocation

Principle II: Once a positive importance profile $I(t)$ is specified, natural variational allocation rules produce power-law timestep densities.

The second justification is variational. A timestep density $\rho(t)$ specifies how a fixed evaluation budget is distributed over sampling time. If $\rho(t)$ is small in a region where the importance profile $I(t)$ is large, then the sampler under-resolves an important part of the path. Thus a natural allocation rule should penalize assigning too little timestep density to high-importance regions.

The Euler local-error proxy gives one concrete example. Since a timestep density induces a local step size $h(t) \approx \frac{1}{B\rho(t)}$, and local truncation error scales quadratically in $h(t)$, the leading accumulated Euler-risk contains a term of the form $\int_0^1 \frac{I(t)}{\rho(t)}\, dt$. This corresponds to a power-law penalty against small $\rho(t)$. At the other extreme, if the goal is simply to match a prescribed importance profile, then the weighted cross-entropy objective $\mathcal{C}(\rho; I) := -\int_0^1 I(t) \log \rho(t)\, dt$ is natural. This motivates the following family, which interpolates between these two allocation principles.

Let $I : [0, 1] \to (0, \infty)$ be a positive importance profile and let ρ be a normalized timestep density. For $\beta \geq 0$, define

$$\psi_\beta(r) = \begin{cases} \dfrac{r^{-\beta} - 1}{\beta}, & \beta > 0, \\ -\log r, & \beta = 0, \end{cases} \qquad r > 0. \tag{7}$$

The objective

$$\min_{\rho>0,\ \int_0^1 \rho(t)\, dt = 1} \mathcal{F}_\beta(\rho; I), \qquad \mathcal{F}_\beta(\rho; I) = \int_0^1 I(t)\psi_\beta(\rho(t))\, dt \tag{8}$$

penalizes under-allocation of density in high-importance regions. The parameter β controls the strength of this penalty: $\beta = 0$ gives the logarithmic cross-entropy penalty, while $\beta = 1$ gives the Euler-risk penalty $\int I(t)/\rho(t)\, dt$, up to an additive constant and scale.

Proposition 2 (Variational allocation gives a power law)**.** *The unique minimizer of eq.* (8) *is*

$$\rho^\star_\beta(t) = \frac{I(t)^{1/(\beta+1)}}{\int_0^1 I(s)^{1/(\beta+1)}\, ds}. \tag{9}$$

Thus variational timestep allocation naturally produces power-law densities in the importance profile.

See Proposition 5 for a detailed version. The solution $\rho^\star_\beta(t) \propto I(t)^{1/(\beta+1)}$ shows how the variational family converts an importance profile into a timestep density. When $\beta = 0$, the scheduler directly matches the importance profile, $\rho^\star_0(t) \propto I(t)$. When $\beta = 1$, it produces the square-root law, $\rho^\star_1(t) \propto \sqrt{I(t)}$. Larger β produces flatter densities because the power $1/(\beta+1)$ becomes smaller.

This family recovers two timestep schedules of SharpEuler. First, let $\bar{a}(t) = \mathbb{E}_{x_0 \sim p_0} \|\ddot{x}(t; x_0)\|$ be the population acceleration sharpness. If ρ induces local step size $h(t) \approx \frac{1}{B\rho(t)}$, then the accumulated leading Euler local-error (or risk) proxy scales as $\mathcal{E}[\rho] \approx \frac{1}{2B} \int_0^1 \frac{\bar{a}(t)}{\rho(t)}\, dt$. This corresponds to $\beta = 1$ and $I(t) = \bar{a}(t)$, yielding $\rho^\star_{\mathrm{Euler}}(t) \propto \bar{a}(t)^{1/2}$.

Second, the profile-matching objective corresponds to $\beta = 0$, i.e. $\min_{\rho>0,\ \int \rho=1} \mathcal{C}(\rho; I)$. Its unique minimizer is $\rho_I(t) = \frac{I(t)}{\int_0^1 I(s)\, ds}$. Equivalently, with $\pi_I(t) = \frac{I(t)}{\int_0^1 I(s)\, ds}$, the objective equals a constant multiple of $H(\pi_I) + \mathrm{KL}(\pi_I \| \rho)$, and is minimized uniquely by $\rho = \pi_I$.

SharpEuler uses the solver-aware profile $I_\gamma(t) = \bar{a}(t)^\gamma$, hence $\rho_\gamma(t) \propto \bar{a}(t)^\gamma$. The special case $\gamma = 1/2$ is selected by the Euler-risk proxy.

SharpEuler with exponent γ can therefore be interpreted as profile-matching to the solver-aware importance profile

$$I_\gamma(t) = \bar{a}(t)^\gamma, \qquad \rho_\gamma(t) \propto \bar{a}(t)^\gamma. \tag{10}$$

Here γ shapes the sharpness profile, while the variational parameter β describes how a chosen profile is converted into a density. The implemented exponent schedules correspond to the profile-matching case $\beta = 0$. The square-root case $\gamma = 1/2$ is special: up to small smoothing, it coincides with the Euler-risk minimizer $\rho(t) \propto \bar{a}(t)^{1/2}$. For $\gamma \neq 1/2$, the schedule remains a valid profile-matching density, but is not generally Euler-risk optimal. In all cases, the underlying sharpness signal is justified by the numerical principle. Full derivations and details are given in App. C.

3.3 Statistical principle: statistical stability of the calibrated schedule

Principle III: With M calibration trajectories, the calibrated schedule is near its population oracle up to a finite-calibration term of order $\mathcal{O}(\sqrt{\log(N/\delta)/M})$ on a grid of size N.

The preceding principles are population-level statements that involve the population sharpness profile $\bar{a}(t) = \mathbb{E}_{x_0 \sim p_0} \|\ddot{x}(t; x_0)\|$. In practice, SharpEuler estimates this profile from finitely many calibration trajectories and then constructs an empirical timestep density. Thus we also need the resulting grid to be statistically stable. The theorem below provides such a guarantee under reasonable assumptions.

Theorem 1 (Informal: statistical stability for the square-root schedule $\gamma = 1/2$)**.** *Let $\widehat{w}$ be the schedule calibrated from M independent trajectories on a grid with N intervals, and let $w^\star$ be the population oracle schedule for the Euler local-error proxy $\mathcal{J}$ defined in App. C.3. Under boundedness, concentration, and Euler stability assumptions, with probability at least $1 - \delta$,*

$$W_1(\mu_{\widehat{w}}, \mu_1) \le \frac{1}{B} \left[C_{\mathrm{Eul}} \mathcal{J}(w^\star) + C_1 \sqrt{\frac{\log(N/\delta)}{M}} + C_2 \right]. \tag{11}$$

Here W_1 denotes the 1-Wasserstein distance, $\mu_{\widehat{w}}$ is the terminal law of the calibrated Euler sampler, μ_1 is the terminal law of the exact learned ODE flow, C_{Eul} is the Euler stability constant, C_1 collects concentration and schedule-stability constants, and C_2 collects smoothing, finite-grid, and finite-difference proxy bias terms.

See Theorem 2 for a detailed version and discussions for general γ cases. The bound separates the numerical sampling error into an oracle Euler term, a finite-calibration term of order $\sqrt{\log(N/\delta)/M}$, and a residual implementation-bias term. Thus SharpEuler preserves the $\mathcal{O}(1/B)$ Euler scaling while its calibrated schedule concentrates toward its population counterpart as M grows. The result controls numerical error relative to the learned ODE flow, not model error relative to the data distribution.

4 EXPERIMENTS

We evaluate whether SharpEuler improves fixed-budget Euler sampling by changing where, but not how many, model evaluations are placed. We report budgets in number of function evaluations (NFE), where one NFE corresponds to one evaluation of the learned vector field. Whenever possible,the model, initial noise, conditioning input, and solver are held fixed, so differences are attributable only to the timestep grid. Implementation details and additional results are given in App. D and E.

4.1 SYNTHETIC TWO-DIMENSIONAL EXPERIMENTS

We first evaluate SharpEuler in a controlled setting where geometric errors are visible and distributional discrepancies can be measured directly. We use two two-dimensional target distributions, *Branched Manifold* and *Rotated Grid*. The first tests whether a sampler resolves thin, curved structure; the second tests whether it preserves separated modes without introducing off-manifold mass.

Table 1 quantifies the behavior using Density, Coverage (Naeem et al., 2020), and squared 2-Wasserstein distance W_2^2. SharpEuler gives the best results in the low-budget regime. At $B = 8$, SharpEuler improves Coverage on both datasets and achieves the best Density on the Rotated Grid, while $\gamma = 1.5$ gives the highest Density on the Branched Manifold. The W_2^2 metric is consistently minimized by $\gamma = 0.5$, in agreement with the LTE-based choice: this exponent gives the best overall distributional match rather than the most concentrated samples. The Density and Coverage columns also show the effect of the exponent. On the Branched Manifold, increasing γ raises Density but can reduce Coverage and worsen W_2^2, indicating that the samples become more concentrated on parts of the target support. On the Rotated Grid, $\gamma = 1.0$ gives the best Coverage across all budgets and the best or near-best Density, while $\gamma = 1.5$ again loses coverage. These results suggest using $\gamma = 0.5$ when the goal is distributional agreement, and $\gamma = 1.0$ when the priority is a stronger

Table 1: Quantitative results on the two synthetic distributions. Density (D) measures sample fidelity, Coverage (C) measures support coverage, and W_2^2 measures distributional discrepancy. Higher is better for D and C; lower is better for W_2^2. Best values per metric and budget are shown in **bold**.

		$B = 8$			$B = 12$			$B = 16$			$B = 20$		
Dataset	Method	D↑	C↑	$W_2^2 \downarrow$	D↑	C↑	$W_2^2 \downarrow$	D↑	C↑	$W_2^2 \downarrow$	D↑	C↑	$W_2^2 \downarrow$
Branched Manifold	Uniform	0.139	0.357	0.020	0.163	0.398	0.016	0.214	0.456	0.013	0.265	0.512	0.013
	ETS	0.269	0.428	0.055	0.316	0.520	0.026	0.359	0.551	0.017	0.364	0.575	0.016
	SharpEuler γ=0.5	0.261	0.479	**0.017**	0.338	0.554	**0.013**	0.385	0.597	**0.012**	0.404	0.604	**0.012**
	SharpEuler γ=1.0	0.422	**0.582**	0.018	0.435	**0.598**	0.014	0.441	**0.620**	0.015	0.425	**0.613**	0.013
	SharpEuler γ=1.5	**0.510**	0.554	0.035	**0.474**	0.576	0.020	**0.458**	0.608	0.018	**0.456**	0.603	0.015
Rotated Grid	Uniform	0.426	0.581	0.022	0.705	0.771	0.015	0.810	0.685	0.012	0.855	0.699	0.010
	ETS	0.652	0.723	0.038	0.779	0.831	0.026	0.846	0.877	0.017	0.885	0.902	0.017
	SharpEuler γ=0.5	0.845	0.749	**0.020**	0.900	0.890	**0.013**	0.924	0.921	**0.011**	0.941	0.928	**0.009**
	SharpEuler γ=1.0	**0.913**	**0.872**	0.029	0.931	**0.924**	0.020	0.942	**0.934**	0.018	**0.953**	**0.939**	0.015
	SharpEuler γ=1.5	0.903	0.700	0.070	**0.940**	0.820	0.044	**0.951**	0.876	0.033	0.948	0.904	0.028

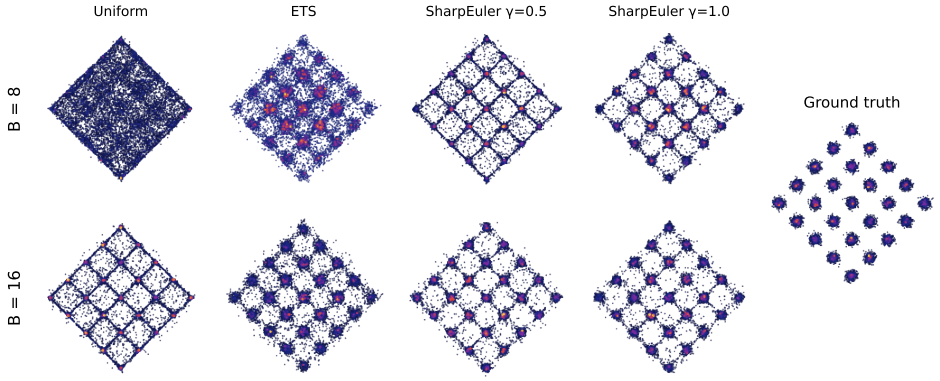


Figure 2: Generated samples on Rotated Grid at $B \in \{8, 16\}$, comparing Uniform, ETS, and SharpEuler with $\gamma \in \{0.5, 1.0\}$. SharpEuler has better visible support in the low-budget regime.

Table 2: FLUX.1-dev results at 1024×1024 resolution on 80 held-out prompts. RMSE measures pixel-level deviation from the 50-step pipeline reference, FID measures distributional discrepancy from the reference image set, and CLIP measures prompt-image alignment. Higher is better for CLIP; lower is better for RMSE and FID. Best values per metric and budget are shown in **bold**.

	$B = 8$			$B = 12$			$B = 16$			$B = 20$		
Method	RMSE↓	CLIP↑	FID↓	RMSE↓	CLIP↑	FID↓	RMSE↓	CLIP↑	FID↓	RMSE↓	CLIP↑	FID↓
Default Pipeline	44.38	0.32	124.31	39.97	0.32	104.77	35.95	**0.33**	90.32	28.58	**0.33**	74.20
Shifted α=3	43.76	0.32	120.84	39.70	0.32	104.01	35.96	**0.33**	89.35	28.55	**0.33**	74.05
SharpEuler γ=0.5	**24.63**	**0.33**	**80.61**	**16.51**	**0.33**	**48.65**	**12.18**	**0.33**	**32.76**	**12.40**	**0.33**	**28.56**



Figure 3: Representative FLUX.1-dev generations at $B = 16$, comparing the default uniform pipeline schedule and SharpEuler against the corresponding 50-step reference images. Relative to the pipeline schedule, SharpEuler more faithfully preserves scene composition, object structure, lighting, and fine visual details from the reference images. These differences are consistent with results in Table 3.

fidelity–coverage tradeoff. Additional comparisons with OSS-ave (Pei et al., 2025) and KTS (Li et al., 2026b) are reported in Table 4 in the appendix. Table 6 evaluates the effect of calibration-set size.

Figure 2 provides visual results for generated samples at $B = 8$ and $B = 16$ for the the Rotated Grid task. SharpEuler produces samples that better match the visible support of the target distribution at the same NFE, but at $B = 16$, the gap between methods narrows.

4.2 Text-to-image generation with FLUX

We next evaluate whether SharpEuler improves sampling for a large pretrained rectified flow model in a high-dimensional vision setting. We use FLUX.1-dev (Black Forest Labs, 2024), a 12B-parameter rectified flow transformer for text-to-image generation, at its native 1024×1024 resolution.

Table 2 reports three complementary measures against the 50-step pipeline reference. RMSE measures pixel-level agreement with the reference sample for the same prompt and seed. FID measures distributional agreement between the generated image set and the reference set. CLIP score measures prompt-image semantic alignment. SharpEuler achieves the lowest RMSE and FID at every budget, with the largest gains in the low-NFE regime. For example, at $B = 12$, RMSE decreases from 39.97 to 16.51, and FID decreases from 104.77 to 48.65. At $B = 16$, FID decreases from 90.32 to 32.76. The shifted $\alpha = 3$ schedule remains close to the default uniform pipeline schedule across budgets, which indicates that the improvement is not explained by a generic monotone shift. CLIP scores stay within a narrow range across all methods. SharpEuler therefore improves agreement with the high-step reference while preserving prompt alignment at fixed NFE. On 1,000 held-out prompts, SharpEuler achieves the lowest KID (unbiased at finite sample sizes) and FID-1k; see Table 7.

Table 3: GPT-5.5 vision-judge comparison between SharpEuler $\gamma = 0.5$ and the default FLUX pipeline schedule. For each prompt, the judge compares both images against the same 50-step reference image and text prompt. Higher is better for scores, Δ, and pairwise wins.

Budget	SharpEuler score	Pipeline score	Mean $\Delta \uparrow$	Std. Δ	p-value	SharpEuler wins
$B = 8$	**7.98**	6.02	**1.96**	0.88	8.2×10^{-21}	**50/50**
$B = 12$	**8.74**	6.50	**2.24**	0.82	1.6×10^{-24}	**50/50**
$B = 16$	**9.04**	6.82	**2.22**	0.93	5.1×10^{-22}	**50/50**
$B = 20$	**9.14**	7.32	**1.82**	0.83	1.2×10^{-20}	**50/50**
$B = 24$	**9.35**	7.52	**1.83**	0.89	9.8×10^{-12}	**50/50**



Figure 4: Side-by-side temporal comparison on Pyramid Flow at a fixed budget of $B = 20$ function evaluations. Both methods use the same prompt and random seed. The columns show four equally spaced frames from the generated videos. Uniform Euler produces a poorly resolved butterfly, whereas SharpEuler produces a detailed butterfly that remains coherent throughout the sequence.

We also evaluate perceptual agreement with the high-step reference using GPT-5.5 as a vision judge. For each prompt, the judge receives the 50-step reference image, the SharpEuler image, the Pipeline image, and the text prompt, and returns both calibrated scores and a forced pairwise preference. Candidate order is randomized to avoid position bias; the judge is instructed to compare semantic content, composition, style, lighting, color, and salient spatial relationships rather than pixel identity. At $B = 16$, SharpEuler receives a mean score of 9.04, compared with 6.82 for Pipeline, giving a mean paired improvement of 2.22 points. Figure 3 shows representative FLUX generations at $B = 16$, comparing SharpEuler and the default pipeline schedule against the 50-step reference image.

4.3 Text-to-Video Generation with Pyramid Flow

We evaluate SharpEuler on the pretrained Pyramid Flow text-to-video model (Jin et al., 2025) at 384×640 resolution; see App. D.6 for details. Figure 4 shows four equally spaced frames generated from the same prompt and seed. Uniform Euler produces a small, poorly resolved butterfly, whereas SharpEuler generates a detailed butterfly and flower that remain coherent throughout the sequence.

5 Conclusion

In this work, we asked where a flow matching sampler should spend its finite Euler budget. SharpEuler answers this question by calibrating a timestep schedule from the learned dynamics of a pretrained model. It estimates where the velocity field changes rapidly along calibration trajectories, smooths the resulting sharpness profile, and converts it into a deterministic Euler grid for any inference budget. The deployed sampler remains unchanged except for the time grid: it uses the same model, the same Euler update, and the same NFE. We justify SharpEuler with numerical, variational, and statistical principles. Experiments on synthetic distributions and FLUX.1-dev show that this calibration improves low-budget sampling at fixed NFE, reducing mode leakage, improving coverage, and producing images closer to high-step references while preserving prompt alignment. These results suggest that efficient flow sampling depends not only on the solver, but also on how its evaluations are placed. Even without retraining or additional solver stages, a fixed-cost Euler sampler can improve when its steps are allocated according to how rapidly the learnt velocity field changes.

AI USE STATEMENT

In this work, we conceived and carried out the original research questions, core ideas, proposed methodology, theoretical contributions, experimental design, and overall execution; generative AI tools were used only in an assistive capacity.

We used generative AI tools to assist with editing and polishing the text, checking mathematical arguments and derivations, and assisting with the implementation and verification of computational methods. We additionally used generative AI tools for improving clarity and presentation, brainstorming and refining research ideas, searching for and identifying potentially relevant literature, assisting with coding and research workflow, and reviewing the manuscript for consistency, clarity, and completeness.

We have reviewed all AI-assisted work. We have independently checked all mathematical arguments, derivations, and claims. We have also verified suggested references against the original sources. The code was inspected and tested, and all reported experimental results were verified. All substantive research decisions, including the formulation of the method, theoretical claims, experimental protocols, interpretation of results, and conclusions, were made by us.

We take full responsibility for the final content of this work, including text, mathematical claims, experimental results, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work advances fundamental methodology for generative modeling and does not involve human subjects, sensitive personal data, or deployment in high-stakes applications. We do not identify any specific societal or ethical concerns arising from the contributions of this work.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate reproducibility. For the theoretical results, the Appendix provides the full mathematical settings, assumptions, statements, and proofs. For the empirical results, the Appendix provides detailed descriptions of the datasets and benchmarks, experimental protocols, model and training configurations, evaluation procedures, and relevant hyperparameters. Upon acceptance, we will release the source code required to reproduce our experiments in a public GitHub repository and provide the corresponding repository link in the paper.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Francis Bach. *Learning Theory from First Principles*. MIT Press, 2024.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

Black Forest Labs. FLUX.1, 2024. URL `https://github.com/black-forest-labs/flux`.

John Charles Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ODE-based diffusion sampling. *arXiv preprint arXiv:2405.11326*, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-First International Conference on Machine Learning*, 2024.

David F Griffiths and Desmond J Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*, volume 5. Springer Science & Business Media, 2010.

Aditi Gupta, Raphael A Meyer, Yotam Yaniv, Elynn Chen, and N Benjamin Erichson. Quantifying epistemic uncertainty in diffusion models. *arXiv preprint arXiv:2602.09170*, 2026.

Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving ordinary differential equations I: Nonstiff problems*. Springer, 1993.

Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. Flowts: Time series generation via rectified flow. *arXiv preprint arXiv:2411.07506*, 2024.

Hugging Face Diffusers. FluxPipeline documentation, 2024. URL `https://huggingface.co/docs/diffusers/api/pipelines/flux`. Default inference budget: 50 steps.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *The Thirteenth International Conference on Learning Representations*.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations*, volume 2025, pp. 23378–23402, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In *International Conference on Machine Learning*, pp. 18957–18973. PMLR, 2023.

Ziyun Li, Huancheng Hu, Soon Hoe Lim, Xuyu Li, Fei Gao, Enmao Diao, Zezhen Ding, Michalis Vazirgiannis, and Henrik Bostrom. A kinetic-energy perspective of flow matching. *arXiv preprint arXiv:2602.07928*, 2026a.

Ziyun Li, Huancheng Hu, Soon Hoe Lim, Xuyu Li, Fei Gao, Enmao Diao, Zezhen Ding, Michalis Vazirgiannis, and Henrik Bostrom. A kinetic energy perspective of flow matching. *arXiv preprint arXiv:2602.07928*, 2026b.

Soon Hoe Lim, Yijin Wang, Annan Yu, Emma Hart, Michael W Mahoney, Sherry Li, and N Benjamin Erichson. Elucidating the design choice of probability paths in flow matching for forecasting. *Transactions on Machine Learning Research*, 2024.

Soon Hoe Lim, Shizheng Lin, Michael W Mahoney, and N Benjamin Erichson. Is flow matching just trajectory replay for sequential data? *arXiv preprint arXiv:2602.08318*, 2026.

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Alex Morehead, Miruna Cretu, Antonia Panescu, Rishabh Anand, Maurice Weiler, Tynan Perez, Samuel Blau, Steven Farrell, Wahid Bhimji, Anubhav Jain, Hrushikesh Sahasrabuddhe, Pietro Liò, Tommi Jaakkola, Rafael Gómez-Bombarelli, Rex Ying, N. Benjamin Erichson, and Michael W. Mahoney. Zatom-1: A multimodal flow foundation model for 3d molecules and materials. *arXiv preprint arXiv:2602.22251*, 2026.

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)*, pp. 7176–7185, 2020.

NVIDIA BioNeMo. Entropic Flow Matching for Optimal Time Scheduling, 2024. URL `https://docs.nvidia.com/bionemo-framework/latest/main/examples/bionemo-moco/entropic_time_scheduler_tutorial_cfm/`. BioNeMo Framework Documentation, MoCo tutorial.

Jianning Pei, Han Hu, and Shuyang Gu. Optimal stepsize for diffusion sampling. *arXiv preprint arXiv:2503.21774*, 2025.

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*, pp. 42947–42975, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=TIdIXIpzhoI`.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

Anuroop Sriram, Benjamin K Miller, Ricky T Chen, and Brandon M Wood. Flowllm: Flow matching for material generation with large language models as base distributions. *Advances in Neural Information Processing Systems*, 37:46025–46046, 2024.

Dejan Stancevic, Florian Handke, and Luca Ambrogioni. Entropic time schedulers for generative diffusion models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.

Panos Tsimpos, Ren Zhi, Jakob Zech, and Youssef Marzouk. Optimal scheduling of dynamic transport. In *The Thirty Eighth Annual Conference on Learning Theory*, pp. 5441–5505, 2025.

Cédric Villani. *Optimal Transport: Old and New*. Springer, 2008.

Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2022.

Christopher Williams, Andrew Campbell, Arnaud Doucet, and Saifuddin Syed. Score-optimal diffusion schedules. *Advances in Neural Information Processing Systems*, 37:107960–107983, 2024.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8292–8301, 2024.

Zhantao Yang, Ruili Feng, Han Zhang, Yujun Shen, Kai Zhu, Lianghua Huang, Yifei Zhang, Yu Liu, Deli Zhao, Jingren Zhou, et al. Lipschitz singularities in diffusion models. In *ICLR*, 2024.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Aiqing Zhu, Pengzhan Jin, Beibei Zhu, and Yifa Tang. On numerical integration in neural ordinary differential equations. In *International Conference on Machine Learning*, pp. 27527–27547. PMLR, 2022.

APPENDIX

This appendix is organized as follows. In App. A we discuss related work in detail and position SharpEuler relative to recent works. In App. B we provide the detailed algorithms for SharpEuler. In App. C we provide full mathematical details and proofs for the three principles that we use to justisfy SharpEuler. In App. D-E we provide details on the experiments and additional results.

A RELATED WORK

Fast samplers and solver design. Much of the early work on fast generation was developed for diffusion and score-based models, where sampling also requires many sequential neural network evaluations. DDIM Song et al. (2021) showed that deterministic sampling can greatly reduce the number of denoising steps. DPM-Solver Lu et al. (2022) and UniPC Zhao et al. (2023) design higher-order solvers that exploit the structure of diffusion ODEs, while EDM Karras et al. (2022) clarified how solver choices, parameterization, and noise schedules interact. Other approaches use additional training to obtain very-few-step samplers, including Progressive Distillation Salimans & Ho (2022), Consistency Models Song et al. (2023), and Latent Consistency Models Luo et al. (2023). These works address the same practical bottleneck: sample quality per model evaluation. In contrast, we study pretrained flow matching models, keep the Euler solver fixed, and change only where a fixed number of evaluations are placed.

Timestep schedules and time reparameterization. A complementary line of work treats the timestep schedule itself as an object of design. Prior work has optimized discrete DDPM schedules using dynamic programming and ELBO objectives Watson et al. (2021), differentiated through sample quality to search for fast samplers Watson et al. (2022), optimized timesteps for a chosen numerical ODE solver Xue et al. (2024), and shown that the best schedule can depend on the solver, model, and data Sabour et al. (2024). Other work derives path-traversal costs and discretizes the path to minimize them Williams et al. (2024). Entropic time schedulers (ETS) Stancevic et al. (2025) also view schedules as time reparameterizations, but allocate steps using conditional-entropy information or rescaled conditional-entropy production. GITS Chen et al. (2024) uses dynamic programming over trajectory costs to select timesteps. These works show that uniform schedules are often not the right default. SharpEuler shares this motivation, but uses a different signal: it estimates trajectory sharpness of a pretrained flow matching model and converts this solver-aware profile into a fixed Euler grid.

Flow matching, path geometry, and regularity. Flow matching Lipman et al. (2023), rectified flow Liu et al. (2023), and large-scale rectified flow models Esser et al. (2024) have made continuous-time vector field models a practical alternative to traditional diffusion formulations. In these models, sampling is again an ODE discretization problem. Some recent work studies timestep or noise sampling choices during training Esser et al. (2024), while other work aims to reduce trajectory curvature during training Lee et al. (2023). Recent work has also studied time-dependent sensitivity in diffusion networks Yang et al. (2024), discretization error in diffusion sampling Benton et al. (2024), time parameterizations that improve velocity regularity in dynamic transport Tsimpos et al. (2025), and trajectory energy as a diagnostic for flow matching samples Li et al. (2026a). These works are related in spirit because they treat path geometry as important. SharpEuler uses this perspective at inference time: rather than changing the training path or model, it measures finite-difference acceleration along sampling trajectories, the signal selected by Euler local truncation error and backward error analysis.

Positioning of SharpEuler. SharpEuler is, to our knowledge, the first timestep scheduler for learned flow samplers that derives its schedule from an explicitly solver-aware Euler error proxy and estimates this profile using offline finite differences of the learned velocity field. While prior work has optimized timestep allocation using information-theoretic, spectral, or black-box objectives, SharpEuler targets the trajectory acceleration profile that controls Euler discretization error. This profile appears both in Euler local truncation error and in the leading backward error perturbation. The resulting square-root schedule is the closed-form minimizer of a continuous Euler-risk objective, and our statistical stability result shows that the empirically calibrated schedule is near-oracle for the generated distribution.

Limitations of SharpEuler. SharpEuler assumes that the pretrained model can be sampled accurately on grids different from the one used during training or deployment. This assumption may fail for

distilled models, or for models explicitly aligned to a prescribed sampling grid. For example, samplers used with Stable Diffusion variants are often tuned jointly with particular schedules, and changing the grid can violate the assumptions under which the model or sampler was calibrated. SharpEuler is also solver-specific. The schedule is derived from a leading local-error proxy for forward Euler, so the resulting grid is designed for a one-stage Euler update. Higher-order solvers, multistep solvers, or adaptive methods may require different error indicators and different timestep allocation rules. Extending sharpness-aware calibration to these solvers is an interesting direction for future work.

B Detailed Algorithms

In this section, we provide detailed algorithms for SharpEuler. We use the reverse-time convention (using s_k for the time indices), and consider a uniform reference grid ($\Delta_i := 1/N$ for all i).

Algorithm 1 Offline sharpness profile calibration

Require: Calibration inputs $\{c_j\}_{j=1}^{M}$, model v_θ, reverse-time grid $1 = s_0 > \cdots > s_N = 0$, exponent $\gamma > 0$, smoothing bandwidth $\sigma > 0$, small floor $\varepsilon_a \geq 0$
Ensure: Empirical inverse-CDF map F_{emp}^{-1}

for $j = 1$ **to** M **do**
  Sample $x_j \sim \mathcal{N}(0, I)$
  for $i = 0$ **to** $N - 1$ **do**
    $V_{j,i} \leftarrow v_\theta(x_j, s_i; c_j)$
    $x_j \leftarrow x_j - (s_i - s_{i+1})V_{j,i}$
  end for
end for
for $i = 0$ **to** $N - 2$ **do**
  $d_i \leftarrow \frac{1}{2}\big(|s_i - s_{i+1}| + |s_{i+1} - s_{i+2}|\big)$
  $\widehat{a}_i \leftarrow \frac{1}{M}\sum_{j=1}^{M} \left\| \frac{V_{j,i+1} - V_{j,i}}{d_i} \right\|_2$
end for
Smooth the exponent-shaped signal:

$$\widetilde{I}_\gamma \leftarrow \mathrm{Smooth}_\sigma\left((\widehat{a} + \varepsilon_a)^\gamma\right).$$

Normalize masses:

$$\widehat{w}_i \leftarrow \frac{\widetilde{I}_{\gamma,i}}{\sum_{r=0}^{N-2} \widetilde{I}_{\gamma,r}}, \qquad i = 0, \ldots, N-2.$$

Assign forward-time support points:

$$\widetilde{t}_i \leftarrow 1 - \frac{s_i + s_{i+2}}{2}, \qquad i = 0, \ldots, N-2.$$

Sort $(\widetilde{t}_i, \widehat{w}_i)$ in ascending $\widetilde{t}_i$, build the empirical CDF, and add a left endpoint anchor at $t = 0$.
return the linearly interpolated inverse-CDF map $F_{\mathrm{emp}}^{-1} : [0, 1] \to [0, 1]$.

Algorithm 2 Online SharpEuler sampling

Require: Empirical inverse-CDF map F_{emp}^{-1}, budget B, model v_θ, conditioning input c
Ensure: Sample $x^{(B)}$

Compute forward-time quantiles and convert to reverse time:

$$\widetilde{t}_b \leftarrow F_{\mathrm{emp}}^{-1}(b/B), \qquad s_b \leftarrow 1 - \widetilde{t}_b, \qquad b = 0, \ldots, B.$$

Sample $x^{(0)} \sim \mathcal{N}(0, I)$
for $k = 0$ **to** $B - 1$ **do**
  $x^{(k+1)} \leftarrow x^{(k)} - (s_k - s_{k+1})v_\theta(x^{(k)}, s_k; c)$
end for
return $x^{(B)}$

Here Smooth_σ denotes normalized Gaussian convolution in index space with reflect padding:

$$\big(\mathrm{Smooth}_\sigma(x)\big)_i = \sum_{k=-r}^{r} K_k x_{i+k}, \qquad K_k = \frac{\exp(-k^2/(2\sigma^2))}{\sum_{j=-r}^{r}\exp(-j^2/(2\sigma^2))}, \qquad r = \max\{1, \lfloor 3\sigma \rfloor\},$$

where x_{i+k} is interpreted using reflect padding whenever $i + k \notin \{0, \dots, L-1\}$.

C Theoretical Justifications and Results

We assume throughout this section that the learned velocity field is sufficiently smooth on the region visited by the sampled trajectories. In particular, all derivatives and Taylor expansions below are assumed to be well defined and uniformly bounded along these trajectories.

In this section, we provide full details on the three principles that we use to justify SharpEuler: (I) numerical, (II) variational, and (III) statistical. Recall that, as in Section 3, we use the forward-time convention for the ODE sampler in our theoretical analysis.

C.1 Principle I: Trajectory Acceleration Arises Naturally From Local Truncation Error and Backward Error Analysis

Both local truncation error and backward error analyses are standard in numerical analysis of ODEs, but we include the details here for completeness and to highlight why trajectory acceleration is the relevant sharpness signal.

Let $x(t; x_0)$ denote the solution of the forward-time ODE

$$\dot{x}(t) = u_\theta(x(t), t), \qquad x(0) = x_0, \tag{12}$$

with $x_0 \sim p_0$. We define the pathwise acceleration magnitude by $a(t; x_0) = \|\ddot{x}(t; x_0)\|$. Note that we suppress conditioning variables for readability.

The following result shows that this quantity is precisely the leading quantity selected by the local truncation error of Euler discretization.

Proposition 3 (Euler local truncation error)**.** *Let $x : [t, t+h_0] \to \mathbb{R}^d$ be an exact solution of equation 12, and suppose $x \in C^3([t, t+h_0]; \mathbb{R}^d)$. If forward Euler is initialized at the current state $x(t)$ and takes one step of size $0 < h \le h_0$ in forward time, then*

$$x(t+h) - [x(t) + hu_\theta(x(t), t)] = \frac{h^2}{2}\ddot{x}(t) + R_3(t, h), \tag{13}$$

where $\|R_3(t,h)\| \le \frac{h^3}{6}\sup_{\xi\in[t,t+h]}\|x^{(3)}(\xi)\|$. In particular,

$$x(t+h) - [x(t) + hu_\theta(x(t), t)] = \frac{h^2}{2}\ddot{x}(t) + \mathcal{O}(h^3), \qquad \text{as } h \to 0. \tag{14}$$

Consequently, the leading-order one-step local truncation error is controlled by $h^2\|\ddot{x}(t)\|$.

Proof. By Taylor's theorem with remainder applied to the trajectory $x(\cdot)$, there exists a remainder $R_3(t, h)$ satisfying

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + R_3(t, h),$$

with

$$\|R_3(t,h)\| \le \frac{h^3}{6}\sup_{\xi\in[t,t+h]}\|x^{(3)}(\xi)\|.$$

Since $x(t)$ solves the forward-time ODE, $\dot{x}(t) = u_\theta(x(t), t)$, subtracting the Euler update $x(t) + hu_\theta(x(t), t)$ gives Eq. equation 13. Therefore,

$$\|x(t+h) - [x(t) + hu_\theta(x(t), t)]\| \le \frac{h^2}{2}\|\ddot{x}(t)\| + \frac{h^3}{6}\sup_{\xi\in[t,t+h]}\|x^{(3)}(\xi)\|.$$

Thus, as $h \to 0$, the leading term is $\frac{h^2}{2}\ddot{x}(t)$, so the leading local truncation error is governed by $h^2\|\ddot{x}(t)\|$, up to the constant factor $1/2$. □

$\psi_h(t, x) = x + hu_\theta(x, t)$, =

The same sharpness signal is also selected by backward error analysis Hairer et al. (2006). The central idea of backward error analysis is to interpret a numerical method not merely as an approximation to the original ODE, but as the exact flow, up to a prescribed order in the step size, of a nearby step-size-dependent ODE. Thus, instead of asking only how far one Euler step is from the exact flow of

$$\dot{x} = u_\theta(x, t),$$

one asks which modified differential equation Euler is solving. In the present setting, this means seeking a modified vector field

$$u_{\theta,h}(x, t) = u_\theta(x, t) + hu_{\theta,1}(x, t) + h^2 u_{\theta,2}(x, t) + \cdots$$

whose exact flow over one step of length h matches the Euler update to a prescribed order. The difference between $u_{\theta,h}$ and u_θ then describes the finite-step bias induced by the solver. Since few-step sampling uses non-negligible step sizes, the modified equation viewpoint gives a useful way to identify where the numerical method is most strongly biased. This viewpoint has also been used to study numerical integration in Neural ODEs Zhu et al. (2022).

The following proposition makes this connection explicit.

Proposition 4 (Informal: backward error interpretation)**.** *Consider the forward-time non-autonomous ODE*

$$\dot{x} = u_\theta(x, t), \tag{15}$$

where u_θ is sufficiently smooth (e.g., C^3 locally in (x, t)). Forward Euler applied to equation 15 has one-step map

$$\psi_h(t, x) = x + hu_\theta(x, t). \tag{16}$$

There exists a formal modified vector field, at the level of formal power series in h, of the form

$$u_{\theta,h}(x, t) = u_\theta(x, t) + hu_{\theta,1}(x, t) + \mathcal{O}(h^2), \tag{17}$$

whose exact time-h flow matches that of the Euler map up to local error $\mathcal{O}(h^3)$, as $h \to 0$. Its leading correction term is

$$u_{\theta,1}(x, t) = -\frac{1}{2}\left(\partial_t u_\theta(x, t) + \nabla_x u_\theta(x, t)u_\theta(x, t)\right). \tag{18}$$

Equivalently,

$$u_{\theta,h}(x, t) = u_\theta(x, t) - \frac{h}{2}\left(\partial_t u_\theta(x, t) + \nabla_x u_\theta(x, t)u_\theta(x, t)\right) + \mathcal{O}(h^2). \tag{19}$$

Along exact trajectories of equation 15, the leading modified equation perturbation is therefore proportional to $-\frac{h}{2}\ddot{x}(t)$.

Consequently, regions with large acceleration are, to leading order, precisely the regions where Euler is both locally less accurate and more strongly biased toward a modified dynamics.

Proof. Let the modified ODE be

$$\dot{\widetilde{x}} = u_{\theta,h}(\widetilde{x}, t), \tag{20}$$

with $u_{\theta,h}$ of the form equation 17. Starting from $\widetilde{x}(t) = x$, Taylor's theorem for the modified trajectory gives

$$\widetilde{x}(t + h) = x + hu_{\theta,h}(x, t) + \frac{h^2}{2}\left[\partial_t u_{\theta,h}(x, t) + \nabla_x u_{\theta,h}(x, t)u_{\theta,h}(x, t)\right] + \mathcal{O}(h^3) \tag{21}$$

as $h \to 0$. Substituting

$$u_{\theta,h}(x, t) = u_\theta(x, t) + hu_{\theta,1}(x, t) + \mathcal{O}(h^2)$$

into equation 21 and retaining terms through order h^2 gives

$$\widetilde{x}(t + h) = x + hu_\theta(x, t) + h^2\left[u_{\theta,1}(x, t) + \frac{1}{2}\left(\partial_t u_\theta(x, t) + \nabla_x u_\theta(x, t)u_\theta(x, t)\right)\right] + \mathcal{O}(h^3). \tag{22}$$

To match the forward Euler map

$$\psi_h(t,x) = x + hu_\theta(x,t)$$

up to $\mathcal{O}(h^3)$, the order-h^2 coefficient must vanish. Hence,

$$u_{\theta,1}(x,t) = -\frac{1}{2}\left(\partial_t u_\theta(x,t) + \nabla_x u_\theta(x,t) u_\theta(x,t)\right),$$

which proves equation 18 and equation 19. Finally, along an exact trajectory, by the chain rule,

$$\partial_t u_\theta(x(t),t) + \nabla_x u_\theta(x(t),t) u_\theta(x(t),t) = \frac{d}{dt} u_\theta(x(t),t) = \ddot{x}(t).$$

Therefore the leading backward error perturbation is proportional to $-h\ddot{x}(t)/2$. □

The preceding results show that the same quantity,

$$\|\partial_t u_\theta(x(t),t) + \nabla_x u_\theta(x(t),t) u_\theta(x(t),t)\| = \|\ddot{x}(t)\|, \tag{23}$$

is selected by two complementary numerical perspectives. From the local truncation error viewpoint, it controls the leading one-step Euler error. From the backward error viewpoint, it controls the leading perturbation in the modified vector field implicitly integrated by Euler. Therefore, regions with large trajectory acceleration are, to leading order, regions where Euler is both locally less accurate and more strongly biased toward modified dynamics. This provides a numerical justification for using velocity differences along calibration trajectories as a sharpness signal in SharpEuler.

C.2 Principle II: Power Scalings in SharpEuler Arise from Variational Allocation Problems

The risk objective induced by the leading Euler local truncation error (Euler-risk) and the profile-matching objective in the main paper can be viewed as two members of a variational family. We introduce this family first, and then recover the two cases relevant for SharpEuler as special cases.

Let $I : [0,1] \to (0,\infty)$ be a positive importance profile, and let $\rho(t)$ be a normalized timestep density. For $\beta \geq 0$, define

$$\psi_\beta(r) = \begin{cases} \dfrac{r^{-\beta} - 1}{\beta}, & \beta > 0, \\ -\log r, & \beta = 0, \end{cases} \qquad r > 0. \tag{24}$$

which is continuous at $\beta = 0$. Given I, define

$$\mathcal{F}_\beta(\rho; I) = \int_0^1 I(t)\psi_\beta(\rho(t))\, dt, \qquad \rho(t) > 0, \qquad \int_0^1 \rho(t)\, dt = 1. \tag{25}$$

For $\beta > 0$, this is, up to an additive constant independent of ρ, equivalent to

$$\mathcal{F}_\beta(\rho; I) = \frac{1}{\beta}\int_0^1 I(t)\rho(t)^{-\beta}\, dt + \text{constant}. \tag{26}$$

For $\beta = 0$, it becomes the weighted cross-entropy objective

$$\mathcal{F}_0(\rho; I) = -\int_0^1 I(t)\log\rho(t)\, dt. \tag{27}$$

The following proposition characterizes the unique minimizer of $\mathcal{F}_\beta(\rho; I)$ over the class of normalized densities $\rho > 0$.

Proposition 5 (Variational timestep density)**.** *Let $I : [0,1] \to (0,\infty)$ be continuous. For every $\beta \geq 0$, the unique minimizer of*

$$\min_{\rho(t)>0,\ \int_0^1 \rho(t)\, dt = 1} \mathcal{F}_\beta(\rho; I) \tag{28}$$

is

$$\rho^\star_\beta(t) = \frac{I(t)^{1/(\beta+1)}}{\int_0^1 I(s)^{1/(\beta+1)}\, ds}. \tag{29}$$

Proof. First suppose $\beta > 0$. Since the term $-\frac{1}{\beta}\int_0^1 I(t)\,dt$ does not depend on ρ, minimizing $\mathcal{F}_\beta$ is equivalent to minimizing

$$\frac{1}{\beta}\int_0^1 I(t)\rho(t)^{-\beta}\,dt \tag{30}$$

subject to $\int_0^1 \rho(t)\,dt = 1$. The Lagrangian is

$$\mathcal{L}(\rho,\lambda) = \frac{1}{\beta}\int_0^1 I(t)\rho(t)^{-\beta}\,dt + \lambda\left(\int_0^1 \rho(t)\,dt - 1\right), \tag{31}$$

where λ is the Lagrange multiplier. Taking the first variation with respect to ρ gives

$$-I(t)\rho(t)^{-\beta-1} + \lambda = 0. \tag{32}$$

Therefore

$$\rho(t) = \left(\frac{I(t)}{\lambda}\right)^{1/(\beta+1)}. \tag{33}$$

Using the normalization constraint,

$$1 = \int_0^1 \rho(t)\,dt = \lambda^{-1/(\beta+1)}\int_0^1 I(t)^{1/(\beta+1)}\,dt. \tag{34}$$

Thus

$$\lambda^{1/(\beta+1)} = \int_0^1 I(t)^{1/(\beta+1)}\,dt, \tag{35}$$

and hence Eq. equation 29 follows.

For $\beta = 0$, the objective is

$$\mathcal{F}_0(\rho; I) = -\int_0^1 I(t)\log\rho(t)\,dt. \tag{36}$$

The Lagrangian is

$$\mathcal{L}(\rho,\lambda) = -\int_0^1 I(t)\log\rho(t)\,dt + \lambda\left(\int_0^1 \rho(t)\,dt - 1\right). \tag{37}$$

The first variation gives

$$-\frac{I(t)}{\rho(t)} + \lambda = 0. \tag{38}$$

Thus $\rho(t) = \frac{I(t)}{\lambda}$, and normalizing gives

$$\rho_0^\star(t) = \frac{I(t)}{\int_0^1 I(s)\,ds}, \tag{39}$$

which agrees with Eq. equation 29 at $\beta = 0$.

Finally, note that ψ_β is strictly convex on $r > 0$ for every $\beta \geq 0$. Indeed, for $\beta > 0$,

$$\psi_\beta''(r) = (\beta+1)r^{-\beta-2} > 0, \tag{40}$$

and for $\beta = 0$,

$$\psi_0''(r) = \frac{1}{r^2} > 0. \tag{41}$$

Since $I(t) > 0$, the objective is strictly convex on the positive density constraint set. Thus, the minimizer is unique. □

Special case 1: Euler-risk objective. Let $\bar{a}(t) = \mathbb{E}_{x_0 \sim p_0} \|\ddot{x}(t; x_0)\|$ be the population mean acceleration sharpness. Given an Euler budget B, the local step size induced by ρ scales as

$$h(t) \approx \frac{1}{B\rho(t)}. \tag{42}$$

Using Eq. equation 42, the accumulated leading Euler local-error proxy scales as

$$\mathcal{E}[\rho] \approx \frac{1}{2B} \int_0^1 \frac{\bar{a}(t)}{\rho(t)}\, dt. \tag{43}$$

The factor $1/\rho(t)$, rather than $1/\rho(t)^2$, appears because the local error is accumulated over the number of steps assigned to each time region. A small interval dt receives approximately $B\rho(t)\, dt$ Euler steps, and each step contributes a leading local error of size $\frac{1}{2}h(t)^2\bar{a}(t)$. Therefore

$$B\rho(t)\, dt \cdot \frac{1}{2}\left(\frac{1}{B\rho(t)}\right)^2 \bar{a}(t) = \frac{1}{2B}\frac{\bar{a}(t)}{\rho(t)}\, dt. \tag{44}$$

The common factor $1/(2B)$ does not affect the optimal density. This Euler local-error objective is recovered from the unified family by taking

$$\beta = 1, \qquad I(t) = \bar{a}(t). \tag{45}$$

Then Eq. equation 29 gives

$$\rho^\star_{\mathrm{Euler}}(t) = \frac{\sqrt{\bar{a}(t)}}{\int_0^1 \sqrt{\bar{a}(s)}\, ds}. \tag{46}$$

Therefore, the Euler local-error analysis selects the square-root density

$$\rho(t) \propto \bar{a}(t)^{1/2}. \tag{47}$$

Special case 2: cross-entropy and profile matching. The profile-matching or cross-entropy objective is recovered by taking $\beta = 0$. Given any positive importance profile $I(t) > 0$, the minimizer is

$$\rho_I(t) = \frac{I(t)}{\int_0^1 I(s)\, ds}. \tag{48}$$

Equivalently, ρ_I is the unique minimizer of

$$\min_{\rho(t)>0,\ \int_0^1 \rho(t)\, dt = 1} -\int_0^1 I(t) \log \rho(t)\, dt. \tag{49}$$

Now, for $\beta = 0$, define the normalized importance density

$$\pi_I(t) = \frac{I(t)}{\int_0^1 I(s)\, ds}. \tag{50}$$

Then

$$-\int_0^1 I(t) \log \rho(t)\, dt = \left(\int_0^1 I(s)\, ds\right)\left[-\int_0^1 \pi_I(t) \log \rho(t)\, dt\right] \tag{51}$$

$$= \left(\int_0^1 I(s)\, ds\right)\left[H(\pi_I) + \mathrm{KL}(\pi_I \| \rho)\right]. \tag{52}$$

Since the entropy $H(\pi_I)$ does not depend on ρ, minimizing the cross-entropy objective is equivalent to minimizing the Kullback-Leibler divergence $\mathrm{KL}(\pi_I \| \rho)$. The unique minimizer is therefore $\rho = \pi_I$.

SharpEuler exponent schedules. SharpEuler with exponent $\gamma > 0$ can be viewed as an empirical estimator of the solver-aware importance profile

$$I_\gamma(t) = \left(\mathbb{E}_{x_0 \sim p_0} \|\ddot{x}(t; x_0)\|\right)^\gamma,$$

with associated profile density

$$\pi_{I_\gamma}(t) = \frac{I_\gamma(t)}{\int_0^1 I_\gamma(s)\, ds}.$$

As discussed earlier, the use of trajectory acceleration is motivated by the Euler local truncation error and by the backward error analysis, both of which identify the material derivative of the learned velocity field as the relevant sharpness signal.

Connection to entropy-based schedulers Stancevic et al. (2025). Entropy-based schedules can be viewed as using an information-theoretic importance profile. Following the notation of Stancevic et al. (2025), let $H[x_0 \mid x_t]$ denote the conditional differential entropy of the data endpoint x_0 given the intermediate state x_t,

$$H[x_0 \mid x_t] = -\mathbb{E}_{p(x_0,x_t)}\left[\log p(x_0 \mid x_t)\right].$$

In entropic time, the time change is

$$\phi_{\mathrm{ent}}(t) = H[x_0 \mid x_t],$$

and in rescaled entropic time,

$$\phi_{\mathrm{rent}}(t) = \int_0^t \sigma(\tau)\dot{H}[x_0 \mid x_\tau]\, d\tau.$$

After orienting time so that $\dot{\phi}(t) > 0$, uniform spacing in the new time variable induces the timestep density

$$\rho_\phi(t) = \frac{\dot{\phi}(t)}{\int_0^1 \dot{\phi}(s)\, ds}.$$

Thus entropy-based schedules use conditional-entropy production as their importance profile, whereas SharpEuler uses solver sharpness. The square-root SharpEuler density is special because it minimizes the Euler-risk objective and satisfies $\rho(t) \propto \bar{a}(t)^{1/2}$. An entropic schedule recovers this density only when its induced profile $\dot{\phi}(t)$ is proportional to $\bar{a}(t)^{1/2}$.

Specialization to discrete intervals. Let

$$0 = \tau_0 < \tau_1 < \cdots < \tau_N = 1, \qquad \Delta_i = \tau_{i+1} - \tau_i, \qquad i = 0, \ldots, N-1. \tag{53}$$

A density ρ induces interval masses

$$w_i = \int_{\tau_i}^{\tau_{i+1}} \rho(t)\, dt, \qquad \sum_{i=0}^{N-1} w_i = 1. \tag{54}$$

If the profile is approximately constant on each interval, with value $\phi_i > 0$, then the profile-matching density

$$\rho_\gamma(t) \propto \phi_i^\gamma, \qquad t \in [\tau_i, \tau_{i+1}), \tag{55}$$

induces weights

$$w_i^{(\gamma)} = \frac{\Delta_i \phi_i^\gamma}{\sum_{\ell=0}^{N-1} \Delta_\ell \phi_\ell^\gamma}. \tag{56}$$

In particular, for $\gamma = 1/2$,

$$w_i^{(1/2)} = \frac{\Delta_i \sqrt{\phi_i}}{\sum_{\ell=0}^{N-1} \Delta_\ell \sqrt{\phi_\ell}}. \tag{57}$$

Proposition 6 (Discrete profile matching)**.** *Fix $\gamma > 0$ and $\phi_i > 0$. The weights in Eq. equation 56 are the unique minimizer of*

$$\min_{w_i>0,\ \sum_i w_i = 1} -\sum_{i=0}^{N-1} \Delta_i \phi_i^\gamma \log w_i. \tag{58}$$

Proof. Let $q_i = \Delta_i \phi_i^\gamma$. The Lagrangian is

$$\mathcal{L}(w, \lambda) = -\sum_i q_i \log w_i + \lambda \left(\sum_i w_i - 1 \right), \tag{59}$$

where λ is the Lagrange multiplier. The first-order optimality condition gives

$$-\frac{q_i}{w_i} + \lambda = 0, \tag{60}$$

so $w_i = q_i/\lambda$. The normalization constraint gives

$$\lambda = \sum_i q_i, \tag{61}$$

and hence

$$w_i = \frac{\Delta_i \phi_i^\gamma}{\sum_{\ell=0}^{N-1} \Delta_\ell \phi_\ell^\gamma}. \tag{62}$$

Strict convexity of $-\log w_i$ gives uniqueness and concludes the proof. □

C.3 Principle III: Statistical Stability Guarantee For the Calibrated Schedule

In the main paper, we introduce a practical offline calibration procedure for learned flow samplers that estimates where the learned vector field is numerically sharp and converts this profile into a fixed-budget timestep schedule (SharpEuler). In this subsection, we provide a statistical stability guarantee to show that the calibrated schedule is near-oracle at the level of the generated distribution at the terminal time $t = 1$. We focus on the square-root schedule $\gamma = 1/2$, which is the exponent selected by the leading Euler local-error proxy. General exponent schedules are discussed in Remark 3.

Motivation. The scheduler is calibrated offline from finitely many ODE trajectories. Hence the schedule used at sampling time is random, since it depends on the calibration set. Our goal here is to show that this calibration randomness is controlled at the level of the final generated distribution.

Setting. We use the forward-time convention. Consider the learned flow ODE

$$\dot{x}(t) = u_\theta(x(t), t), \qquad t \in [0, 1], \tag{63}$$

where $x_0 \sim p_0$ is the initial noise sampled from the base distribution p_0. Let $T(x_0) := x(1; x_0)$ denote the learned flow map, and define the learned flow distribution $\mu_1 := T_\# p_0$.

Let us fix an N from now on, and let $0 = \tau_0 < \tau_1 < \cdots < \tau_N = 1$ be a reference grid, writing $\Delta_i := \tau_{i+1} - \tau_i$, $i = 0, \ldots, N-1$. A schedule is represented by the weights

$$w = (w_0, \ldots, w_{N-1}), \qquad w_i > 0, \qquad \sum_{i=0}^{N-1} w_i = 1. \tag{64}$$

The interpretation is that interval i receives fraction w_i of a total Euler budget B. Thus interval i receives approximately Bw_i steps, with local step size approximately $h_i \approx \frac{\Delta_i}{Bw_i}$. Let $T_w(x_0)$ be the Euler sampler output at $t = 1$ using schedule w, and let $\mu_w := (T_w)_\# p_0$ be the corresponding sampler distribution at $t = 1$.

Local-error risk and implemented sharpness. For each trajectory initialized at x_0, let $A_i(x_0) \geq 0$ be an interval-wise local-error coefficient on $[\tau_i, \tau_{i+1}]$.

A natural continuous coefficient is the interval average

$$A_i(x_0) = \frac{1}{\Delta_i} \int_{\tau_i}^{\tau_{i+1}} \|\ddot{x}(t; x_0)\| \, dt. \tag{65}$$

Alternatively, one may instead take a dominating coefficient such as

$$A_i(x_0) = \sup_{t \in [\tau_i, \tau_{i+1}]} \|\ddot{x}(t; x_0)\|. \tag{66}$$

The motivation comes from Euler local truncation error as discussed in Section 3. The leading local Euler error is controlled by $h^2 \|\ddot{x}(t)\|$. If interval i receives Bw_i steps of size $h_i \approx \frac{\Delta_i}{Bw_i}$, then the accumulated local-error proxy on interval i is

$$Bw_i A_i(x_0) \left(\frac{\Delta_i}{Bw_i} \right)^2 = \frac{1}{B} \frac{A_i(x_0) \Delta_i^2}{w_i}. \tag{67}$$

Ignoring the common factor $1/B$, define the population local-error risk

$$\mathcal{J}(w) := \mathbb{E}_{x_0 \sim p_0}\left[\sum_{i=0}^{N-1} \frac{A_i(x_0)\Delta_i^2}{w_i}\right]. \tag{68}$$

Equivalently,

$$\mathcal{J}(w) = \sum_{i=0}^{N-1} \frac{\mathcal{A}_i \Delta_i^2}{w_i}, \qquad \mathcal{A}_i := \mathbb{E}_{x_0 \sim p_0}[A_i(x_0)]. \tag{69}$$

The factor $1/w_i$ is essential: allocating more steps to interval i decreases its local-error contribution.

Smoothed empirical calibration. Suppose we have M independent calibration trajectories. For each $m = 1, \dots, M$, draw $x_0^{(m)} \sim p_0$ independently, and define $A_i^{(m)} := A_i(x_0^{(m)})$. Note that in implementation, $A_i^{(m)}$ is replaced by the finite-difference proxy.

The empirical raw profile is

$$\widehat{\mathcal{A}}_i = \frac{1}{M} \sum_{m=1}^{M} A_i^{(m)}. \tag{70}$$

The population raw profile is

$$\mathcal{A}_i = \mathbb{E}_{x_0 \sim p_0}[A_i(x_0)]. \tag{71}$$

For the square-root SharpEuler schedule, the exponent is applied after averaging over calibration trajectories:

$$\widehat{g}_i = \sqrt{\widehat{\mathcal{A}}_i + \varepsilon_a}, \qquad \varepsilon_a > 0. \tag{72}$$

Let $K_\sigma(i, \ell) \geq 0$ be Gaussian smoothing weights satisfying

$$\sum_{\ell=0}^{N-1} K_\sigma(i, \ell) = 1.$$

The profile is then smoothed:

$$\widehat{b}_i = \sum_{\ell=0}^{N-1} K_\sigma(i, \ell)\widehat{g}_\ell. \tag{73}$$

Since $\widehat{g}_i \geq 0$ and the smoothing weights are nonnegative, $\widehat{b}_i \geq 0$.

The calibrated weights are

$$\widehat{w}_i = \frac{\Delta_i \widehat{b}_i}{\sum_{\ell=0}^{N-1} \Delta_\ell \widehat{b}_\ell}. \tag{74}$$

The corresponding population square-root profile is $g_i^\star = \sqrt{\mathcal{A}_i + \varepsilon_a}$, and its smoothed version is

$$b_i^\star = \sum_{\ell=0}^{N-1} K_\sigma(i, \ell) g_\ell^\star. \tag{75}$$

Let us also define the effective empirical and population coefficients

$$\widehat{\phi}_i := \widehat{b}_i^2, \qquad \phi_i^\star := (b_i^\star)^2. \tag{76}$$

Then $\widehat{w}$ is the minimizer of the empirical effective Euler-risk proxy

$$\widehat{\mathcal{J}}_{\sigma, \varepsilon_a}(w) = \sum_{i=0}^{N-1} \frac{\widehat{\phi}_i \Delta_i^2}{w_i}. \tag{77}$$

Assuming that these weights are realized through the piecewise-constant scheduling density

$$\widehat{p}(t) = \frac{\widehat{w}_i}{\Delta_i}, \qquad t \in [\tau_i, \tau_{i+1}), \tag{78}$$

we now analyze this idealized interval-allocation schedule. The practical implementation constructs the schedule by forming an empirical CDF from the calibrated masses and evaluating its inverse by linear interpolation; this realizes the same allocation up to integer-rounding, interpolation, and finite-grid boundary effects.

C.3.1 THE MAIN THEOREM

Consider the schedule class

$$\mathcal{W}_\omega := \left\{ w \in \mathbb{R}^N : w_i \geq \omega,\ \sum_{i=0}^{N-1} w_i = 1 \right\}, \qquad 0 < \omega < 1/N. \tag{79}$$

The lower bound prevents degenerate schedules that allocate vanishing mass to an interval and makes the oracle comparison stable.

Let

$$w^\star \in \arg\min_{w \in \mathcal{W}_\omega} \mathcal{J}(w) \tag{80}$$

be the best raw population-risk schedule in this class. Define the population effective objective

$$\mathcal{J}_{\sigma,\varepsilon_a}(w) := \sum_{i=0}^{N-1} \frac{\phi_i^\star \Delta_i^2}{w_i}. \tag{81}$$

and the bias due to smoothing and other procedures

$$\beta_{\sigma,\varepsilon_a} := \max_{0 \leq i \leq N-1} |\phi_i^\star - \mathcal{A}_i| = \max_i \left| (b_i^\star)^2 - \mathcal{A}_i \right|. \tag{82}$$

This term measures the mismatch between the raw profile appearing in the Euler error proxy and the profile optimized by the algorithm.

We make the following assumptions.

(A1) Independent calibration trajectories. The calibration samples $x_0^{(1)}, \ldots, x_0^{(M)}$ are i.i.d. from p_0.

(A2) Bernstein conditions. For every $i = 0, \ldots, N-1$, there exist positive constants R and ν such that

$$|A_i(x_0) - \mathcal{A}_i| \leq R \tag{83}$$

almost surely, and

$$\mathrm{Var}_{x_0 \sim p_0}\left(A_i(x_0)\right) \leq \nu^2. \tag{84}$$

The boundedness assumption is a sufficient condition used to obtain a simple Bernstein concentration bound. In practice, it can be enforced by clipping the calibrated sharpness values, or replaced by an appropriate tail assumption with the corresponding concentration bound.

(A3) Lower-bound and bounded effective-profile conditions. The closed-form empirical schedule $\widehat{w}$ lies in $\mathcal{W}_\omega$. Likewise, the unconstrained minimizer of the population effective objective $\mathcal{J}_{\sigma,\varepsilon_a}$ lies in $\mathcal{W}_\omega$. In addition, assume on the calibration event under consideration that

$$0 \leq \widehat{b}_i \leq G_{\max}, \qquad 0 \leq b_i^\star \leq G_{\max}, \qquad i = 0, \ldots, N-1. \tag{85}$$

A sufficient deterministic condition is that $0 \leq A_i(x_0) \leq A_{\max}$ almost surely, in which case one may take $G_{\max} = \sqrt{A_{\max} + \varepsilon_a}$. A sufficient condition for the lower-bound part is that $\Delta_i \geq \Delta_{\min} > 0$ and $\sqrt{\varepsilon_a} \leq \widehat{b}_i \leq G_{\max}$, $\sqrt{\varepsilon_a} \leq b_i^\star \leq G_{\max}$. Then

$$\widehat{w}_i = \frac{\Delta_i \widehat{b}_i}{\sum_\ell \Delta_\ell \widehat{b}_\ell} \geq \frac{\Delta_{\min}\sqrt{\varepsilon_a}}{G_{\max}}, \tag{86}$$

and similarly for the population effective weights. Hence one may choose

$$\omega \leq \frac{\Delta_{\min}\sqrt{\varepsilon_a}}{G_{\max}}. \tag{87}$$

(A4) Euler stability. For every $w \in \mathcal{W}_\omega$, the Euler sampler satisfies

$$\|T_w(x_0) - T(x_0)\| \le \frac{C_{\mathrm{Eul}}}{B} \sum_{i=0}^{N-1} \frac{A_i(x_0)\Delta_i^2}{w_i}, \tag{88}$$

where $C_{\mathrm{Eul}} > 0$ is a stability constant.

Assumption (A4) is a standard nonuniform Euler stability bound. For example, if u_θ is uniformly L-Lipschitz in x and we take $A_i(x_0) = \sup_{t\in[\tau_i,\tau_{i+1}]} \|\ddot{x}(t;x_0)\|$, then a discrete Grönwall argument gives (A4) with $C_{\mathrm{Eul}} = e^L/2$, up to the idealized interval-allocation approximation. We state (A4) as an assumption because learned flow fields may have endpoint singularities Li et al. (2026a) or problem-dependent stability constants.

Define $D_2 := \sum_{i=0}^{N-1} \Delta_i^2$. For $\delta \in (0,1)$, define

$$\eta_M(\delta) := \sqrt{\frac{2\nu^2 \log(2N/\delta)}{M}} + \frac{2R\log(2N/\delta)}{3M}, \tag{89}$$

and

$$\alpha_M(\delta) := \frac{\eta_M(\delta)}{2\sqrt{\varepsilon_a}}, \qquad \zeta_M(\delta) := 2G_{\max}\alpha_M(\delta) \tag{90}$$

With these assumptions and discussions in place, we have the following statistical guarantee in terms of the 1-Wasserstein distance W_1.

Theorem 2 (Statistical stability guarantee)**.** *Let $\delta \in (0,1)$, N, and M be given. Under Assumptions (A1)–(A4), with probability at least $1-\delta$ over the calibration set,*

$$W_1(\mu_{\widehat{w}}, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B}\left[\mathcal{J}(w^\star) + \frac{2D_2}{\omega}\left(\zeta_M(\delta) + \beta_{\sigma,\varepsilon_a}\right)\right]. \tag{91}$$

The high-probability statement above is over the random calibration trajectories. Conditional on the calibrated schedule, $\mu_{\widehat{w}}$ is the law induced by fresh samples $x_0 \sim p_0$. Equivalently, treating $G_{\max}$ and ε_a as fixed constants,

$$W_1(\mu_{\widehat{w}}, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B}\mathcal{J}(w^\star) + O_{\mathbb{P}}\left(\frac{C_{\mathrm{Eul}}D_2}{B\omega}\sqrt{\frac{\log N}{M}}\right) + \frac{C_{\mathrm{Eul}}D_2}{B\omega}\beta_{\sigma,\varepsilon_a}. \tag{92}$$

Remark 1. The oracle term $\frac{C_{\mathrm{Eul}}}{B}\mathcal{J}(w^\star)$ is the best Euler local-error proxy achievable by the schedule class. When all $\mathcal{A}_i > 0$ and the lower-bound constraint $w_i \ge \omega$ is inactive, the oracle problem is

$$\min_{w_i>0,\ \sum_i w_i = 1} \sum_{i=0}^{N-1} \frac{\mathcal{A}_i\Delta_i^2}{w_i}. \tag{93}$$

Its minimizer is

$$w_i^\star = \frac{\Delta_i\sqrt{\mathcal{A}_i}}{\sum_{\ell=0}^{N-1}\Delta_\ell\sqrt{\mathcal{A}_\ell}}, \tag{94}$$

and the optimal value is

$$\mathcal{J}(w^\star) = \left(\sum_{i=0}^{N-1}\Delta_i\sqrt{\mathcal{A}_i}\right)^2. \tag{95}$$

Thus, in the oracle case,

$$W_1(\mu_{w^\star}, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B}\left(\sum_{i=0}^{N-1}\Delta_i\sqrt{\mathcal{A}_i}\right)^2. \tag{96}$$

Since $\mathcal{J}(w^\star)$ does not depend on B, the oracle discretization term scales as $O(1/B)$, matching the standard global convergence rate of Euler integration.

The sharpness-aware schedule improves the constant in this rate by allocating more steps to intervals with larger local-error coefficients. The improvement is in the leading error constant: for the unconstrained oracle, applying Cauchy–Schwarz inequality:

$$\mathcal{J}(w^\star) = \left(\sum_i \Delta_i \sqrt{\mathcal{A}_i}\right)^2 \le \sum_i \Delta_i \mathcal{A}_i = \mathcal{J}(w^{\mathrm{unif}}),$$

where $w_i^{\mathrm{unif}} = \Delta_i$, corresponding to uniform step density in time. Thus the sharpness-aware oracle is never worse than uniform allocation for the local-error proxy, with larger gains when the sharpness profile is more nonuniform, while preserving the same $O(1/B)$ Euler rate.

As $M \to \infty$, the finite-calibration term vanishes at rate $O(M^{-1/2})$ up to a logarithmic factor in N, so the calibrated schedule approaches the population schedule, up to the remaining gap due to smoothing and other bias $\beta_{\sigma,\varepsilon_a}$.

Remark 2. The theorem compares two distributions at $t = 1$, i.e. $\mu_{\widehat{w}} = (T_{\widehat{w}})_{\#}p_0$, the law generated by the Euler sampler using the empirically calibrated schedule, and $\mu_1 = T_{\#}p_0$, the law generated by the learned ODE flow. It does not directly compare $\mu_{\widehat{w}}$ to the true data distribution. A full distribution bound would need the model error term:

$$W_1(\mu_{\widehat{w}}, p_{\mathrm{data}}) \le W_1(\mu_{\widehat{w}}, \mu_1) + W_1(\mu_1, p_{\mathrm{data}}). \tag{97}$$

The scheduler controls only the first term. At a high level, the final sampler error can be decomposed as:

final sampler error $\le$ best schedule error + finite calibration error + smoothing and other bias.

Here, we compare the flow induced by the calibrated schedule to the learned ODE flow. Thus the result controls numerical sampling error, rather than model error due to neural network approximation and optimization.

Remark 3 (General exponent schedules). The implementation also allows schedules of the form

$$\widehat{g}_i^{(\gamma)} = \left(\widehat{\mathcal{A}}_i + \varepsilon_a\right)^\gamma, \qquad \widehat{b}_i^{(\gamma)} = \sum_\ell K_\sigma(i,\ell)\widehat{g}_\ell^{(\gamma)},$$

followed by

$$w_i^{(\gamma)} = \frac{\Delta_i \widehat{b}_i^{(\gamma)}}{\sum_{\ell=0}^{N-1} \Delta_\ell \widehat{b}_\ell^{(\gamma)}}.$$

The theorem above corresponds to $\gamma = 1/2$, which is the exponent selected by the Euler local-error proxy. The statistical concentration part of the proof extends straightforwardly to these schedules, with constants depending on γ. However, only the square-root case $\gamma = 1/2$ is the minimizer of the Euler local-error proxy $\sum_i \frac{\mathcal{A}_i \Delta_i^2}{w_i}$. Therefore the near-oracle statement in Theorem 2 is stated for the square-root SharpEuler schedule. For $\gamma \neq 1/2$, the analogous guarantee should be interpreted as stability around the corresponding population exponent schedule, rather than as optimality for the Euler-error oracle.

C.3.2 PROOFS

We provide a detailed proof to Theorem 2.

First, we recall the continuous version of the minimization problem (see also equation 45-equation 46).

Proposition 7. *Let $a : [0,1] \to (0,\infty)$ be continuous. Among all normalized positive densities $\rho(t) > 0$, $\int_0^1 \rho(t)\,dt = 1$, the unique minimizer of $\int_0^1 \frac{a(t)}{\rho(t)}\,dt$ is*

$$\rho^\star(t) = \frac{\sqrt{a(t)}}{\int_0^1 \sqrt{a(s)}\,ds}. \tag{98}$$

Moreover, the minimum value is $\left(\int_0^1 \sqrt{a(t)}\,dt\right)^2$.

Proof. We solve the constrained optimization problem

$$\min_{\rho} J[\rho], \qquad J[\rho] = \int_0^1 \frac{a(t)}{\rho(t)}\, dt, \tag{99}$$

subject to $\rho(t) > 0$, $\int_0^1 \rho(t)\, dt = 1$. Introduce a Lagrange multiplier $\lambda \in \mathbb{R}$ for the normalization constraint and define

$$\mathcal{L}[\rho, \lambda] = \int_0^1 \frac{a(t)}{\rho(t)}\, dt + \lambda \left(\int_0^1 \rho(t)\, dt - 1 \right). \tag{100}$$

To compute the first variation, let $\eta(t)$ be a smooth perturbation and consider

$$\rho_\epsilon(t) = \rho(t) + \epsilon \eta(t), \tag{101}$$

for sufficiently small ϵ so that $\rho_\epsilon(t) > 0$. Then

$$\mathcal{L}[\rho_\epsilon, \lambda] = \int_0^1 \frac{a(t)}{\rho(t) + \epsilon\eta(t)}\, dt + \lambda \left(\int_0^1 [\rho(t) + \epsilon\eta(t)]\, dt - 1 \right). \tag{102}$$

Differentiating with respect to ϵ at $\epsilon = 0$ gives

$$\frac{d}{d\epsilon} \mathcal{L}[\rho_\epsilon, \lambda] \Big|_{\epsilon=0} = \int_0^1 \left(-\frac{a(t)}{\rho(t)^2} \right) \eta(t)\, dt + \lambda \int_0^1 \eta(t)\, dt \tag{103}$$

$$= \int_0^1 \left(-\frac{a(t)}{\rho(t)^2} + \lambda \right) \eta(t)\, dt. \tag{104}$$

At an interior constrained optimum, the first variation of the Lagrangian must vanish for all smooth perturbations η. Therefore,

$$-\frac{a(t)}{\rho(t)^2} + \lambda = 0 \qquad \text{for all } t \in [0, 1]. \tag{105}$$

Rearranging gives

$$\rho(t)^2 = \frac{a(t)}{\lambda}. \tag{106}$$

Since $\rho(t) > 0$ and $a(t) > 0$, we need $\lambda > 0$, and thus

$$\rho(t) = \frac{1}{\sqrt{\lambda}} \sqrt{a(t)}. \tag{107}$$

The normalization constraint determines λ. Substituting Eq. equation 107 into $\int_0^1 \rho(t)\, dt = 1$ gives

$$1 = \int_0^1 \frac{1}{\sqrt{\lambda}} \sqrt{a(t)}\, dt \tag{108}$$

$$= \frac{1}{\sqrt{\lambda}} \int_0^1 \sqrt{a(t)}\, dt. \tag{109}$$

Therefore,

$$\sqrt{\lambda} = \int_0^1 \sqrt{a(t)}\, dt, \tag{110}$$

and hence

$$\rho^\star(t) = \frac{\sqrt{a(t)}}{\int_0^1 \sqrt{a(s)}\, ds}. \tag{111}$$

This proves the claimed form of the optimizer.

We next compute the minimum value. Let $Z = \int_0^1 \sqrt{a(s)}\, ds$. Then

$$\rho^\star(t) = \frac{\sqrt{a(t)}}{Z}. \tag{112}$$

Substituting into the objective gives

$$J[\rho^\star] = \int_0^1 \frac{a(t)}{\rho^\star(t)}\,dt = \int_0^1 \frac{a(t)}{\sqrt{a(t)}/Z}\,dt = Z\int_0^1 \sqrt{a(t)}\,dt = Z^2 = \left(\int_0^1 \sqrt{a(t)}\,dt\right)^2. \quad (113)$$

It remains to justify uniqueness. For each fixed t, the function $r \mapsto \frac{a(t)}{r}$ is strictly convex on $r > 0$ since

$$\frac{d^2}{dr^2}\left(\frac{a(t)}{r}\right) = \frac{2a(t)}{r^3} > 0. \quad (114)$$

Since $a(t) > 0$, the integral functional is strictly convex over positive densities. The feasible set

$$\left\{\rho : \rho(t) > 0,\ \int_0^1 \rho(t)\,dt = 1\right\} \quad (115)$$

is convex. Therefore, any stationary point satisfying the constraint is the unique global minimizer. This completes the proof. □

We now state a discrete version of the minimization problem and characterize the corresponding solution.

Lemma 1. *Define the empirical effective proxy*

$$\widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(w) = \sum_{i=0}^{N-1} \frac{\widehat{\phi}_i \Delta_i^2}{w_i}. \quad (116)$$

The unique minimizer over the simplex $w_i > 0$, $\sum_i w_i = 1$ is

$$\widehat{w}_i = \frac{\Delta_i \widehat{b}_i}{\sum_{\ell=0}^{N-1} \Delta_\ell \widehat{b}_\ell}. \quad (117)$$

Under Assumption (A3), this minimizer lies in $\mathcal{W}_\omega$, and therefore it is also the minimizer over $\mathcal{W}_\omega$.

Proof. The proof is the discrete analogue of Proposition 7, applied to the coefficients $\widehat{\phi}_i = \widehat{b}_i^2$.

Since $\widehat{\phi}_i = \widehat{b}_i^2$, the Lagrangian is

$$\mathcal{L}(w, \lambda) = \sum_i \frac{\widehat{b}_i^2 \Delta_i^2}{w_i} + \lambda\left(\sum_i w_i - 1\right), \quad (118)$$

where λ is the Lagrange multiplier. The first-order optimality condition is

$$-\frac{\widehat{b}_i^2 \Delta_i^2}{w_i^2} + \lambda = 0. \quad (119)$$

Hence, $w_i^2 = \frac{\widehat{b}_i^2 \Delta_i^2}{\lambda}$. Since $w_i > 0$,

$$w_i = \frac{\Delta_i \widehat{b}_i}{\sqrt{\lambda}}. \quad (120)$$

Using $\sum_i w_i = 1$,

$$\sqrt{\lambda} = \sum_{\ell=0}^{N-1} \Delta_\ell \widehat{b}_\ell. \quad (121)$$

Therefore,

$$w_i = \frac{\Delta_i \widehat{b}_i}{\sum_{\ell=0}^{N-1} \Delta_\ell \widehat{b}_\ell}. \quad (122)$$

The objective is strictly convex on the positive simplex because $w \mapsto 1/w$ is strictly convex on $w > 0$. Therefore the minimizer is unique. Since $\mathcal{W}_\omega$ is a subset of the open simplex and the unconstrained global minimizer belongs to $\mathcal{W}_\omega$ under Assumption (A3), it is also the constrained minimizer over $\mathcal{W}_\omega$. □

We next recall Bernstein's inequality (see, e.g., Proposition 1.4 in Bach (2024)) in the form used below.

Lemma 2 (Bernstein's inequality)**.** *Let $Z_1, \dots, Z_n$ be independent random variables satisfying*

$$\mathbb{E}[Z_j] = 0, \qquad |Z_j| \le c \quad \textit{almost surely}. \tag{123}$$

Let $s^2 = \frac{1}{n}\sum_{j=1}^{n} \mathrm{Var}(Z_j)$. Then, for every $t > 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{j=1}^{n} Z_j\right| \ge t\right) \le 2\exp\left(-\frac{nt^2}{2s^2 + \frac{2ct}{3}}\right). \tag{124}$$

Moreover, for every $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\left|\frac{1}{n}\sum_{j=1}^{n} Z_j\right| \le \sqrt{\frac{2s^2\log(2/\delta)}{n}} + \frac{2c\log(2/\delta)}{3n}. \tag{125}$$

The same statement applies to noncentered variables after replacing Z_j by $Z_j - \mathbb{E}[Z_j]$.

Using this inequality and our assumptions, we obtain the following auxiliary result.

Lemma 3. *With probability at least $1-\delta$,*

$$\max_i \left|\widehat{\phi}_i - \phi_i^\star\right| \le \zeta_M(\delta). \tag{126}$$

Proof. We first prove concentration of the raw empirical profile $\widehat{\mathcal{A}}_i$. Fix $i \in \{0, \dots, N-1\}$ and define

$$U_i^{(m)} := A_i^{(m)} - \mathcal{A}_i, \qquad m = 1, \dots, M. \tag{127}$$

By Assumption (A1), the variables $U_i^{(1)}, \dots, U_i^{(M)}$ are independent. By Assumption (A2),

$$\mathbb{E}[U_i^{(m)}] = 0, \qquad |U_i^{(m)}| \le R, \qquad \mathrm{Var}(U_i^{(m)}) \le \nu^2. \tag{128}$$

for some positive constants R, ν. Moreover,

$$\widehat{\mathcal{A}}_i - \mathcal{A}_i = \frac{1}{M}\sum_{m=1}^{M} U_i^{(m)}. \tag{129}$$

Therefore, by Bernstein's inequality (Lemma 2), for every $\eta > 0$,

$$\mathbb{P}\left(\left|\widehat{\mathcal{A}}_i - \mathcal{A}_i\right| \ge \eta\right) \le 2\exp\left(-\frac{M\eta^2}{2\nu^2 + \frac{2}{3}R\eta}\right). \tag{130}$$

Taking a union bound over $i = 0, \dots, N-1$ gives

$$\mathbb{P}\left(\max_i \left|\widehat{\mathcal{A}}_i - \mathcal{A}_i\right| \ge \eta\right) \le 2N\exp\left(-\frac{M\eta^2}{2\nu^2 + \frac{2}{3}R\eta}\right). \tag{131}$$

Set $u = \log(2N/\delta)$. The right-hand side is at most δ for

$$\eta = \sqrt{\frac{2\nu^2 u}{M}} + \frac{2Ru}{3M}. \tag{132}$$

Substituting $u = \log(2N/\delta)$ gives

$$\max_i \left|\widehat{\mathcal{A}}_i - \mathcal{A}_i\right| \le \eta_M(\delta) \tag{133}$$

with probability at least $1-\delta$.

On this event, we next pass from the raw profile to the square-root profile. Since the map $r \mapsto \sqrt{r + \varepsilon_a}$ is $1/(2\sqrt{\varepsilon_a})$-Lipschitz on $r \ge 0$, we have

$$\max_i |\widehat{g}_i - g_i^\star| = \max_i \left|\sqrt{\widehat{\mathcal{A}}_i + \varepsilon_a} - \sqrt{\mathcal{A}_i + \varepsilon_a}\right| \tag{134}$$

$$\le \frac{1}{2\sqrt{\varepsilon_a}} \max_i |\widehat{\mathcal{A}}_i - \mathcal{A}_i| \le \frac{\eta_M(\delta)}{2\sqrt{\varepsilon_a}} = \alpha_M(\delta). \tag{135}$$

Now we pass through the smoothing operator. Since $K_\sigma(i,\ell) \geq 0$ and $\sum_{\ell=0}^{N-1} K_\sigma(i,\ell) = 1$, the smoothing operator is nonexpansive in the sup norm, so:

$$|\widehat{b}_i - b_i^\star| = \left| \sum_{\ell=0}^{N-1} K_\sigma(i,\ell)\,(\widehat{g}_\ell - g_\ell^\star) \right| \leq \sum_{\ell=0}^{N-1} K_\sigma(i,\ell) |\widehat{g}_\ell - g_\ell^\star| \leq \max_\ell |\widehat{g}_\ell - g_\ell^\star| \leq \alpha_M(\delta). \quad (136)$$

Therefore, $\max_i |\widehat{b}_i - b_i^\star| \leq \alpha_M(\delta)$.

Finally, recall that $\widehat{\phi}_i = \widehat{b}_i^2$, $\phi_i^\star = (b_i^\star)^2$. Using Assumption (A3), i.e. $0 \leq \widehat{b}_i \leq G_{\max}$ and $0 \leq b_i^\star \leq G_{\max}$, we get

$$|\widehat{\phi}_i - \phi_i^\star| = |\widehat{b}_i^2 - (b_i^\star)^2| = |\widehat{b}_i - b_i^\star||\widehat{b}_i + b_i^\star| \leq 2G_{\max}|\widehat{b}_i - b_i^\star| \leq 2G_{\max}\alpha_M(\delta) = \zeta_M(\delta). \quad (137)$$

Taking the maximum over i gives $\max_i |\widehat{\phi}_i - \phi_i^\star| \leq \zeta_M(\delta)$ with probability at least $1-\delta$. This completes the proof. □

Next, we will need the following series of lemmas.

Lemma 4. *Let*

$$w^\star_{\sigma,\varepsilon_a} = \arg\min_{w \in \mathcal{W}_\omega} \mathcal{J}_{\sigma,\varepsilon_a}(w). \quad (138)$$

On the event

$$\max_i |\widehat{\phi}_i - \phi_i^\star| \leq \zeta, \quad (139)$$

we have

$$\mathcal{J}_{\sigma,\varepsilon_a}(\widehat{w}) \leq \mathcal{J}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}) + \frac{2D_2}{\omega}\zeta. \quad (140)$$

Proof. For any $w \in \mathcal{W}_\omega$,

$$\left| \widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(w) - \mathcal{J}_{\sigma,\varepsilon_a}(w) \right| = \left| \sum_i \frac{(\widehat{\phi}_i - \phi_i^\star)\Delta_i^2}{w_i} \right| \leq \sum_i \frac{|\widehat{\phi}_i - \phi_i^\star|\Delta_i^2}{w_i} \leq \frac{\zeta}{\omega}\sum_i \Delta_i^2 = \frac{D_2}{\omega}\zeta. \quad (141)$$

Therefore,

$$\sup_{w \in \mathcal{W}_\omega} \left| \widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(w) - \mathcal{J}_{\sigma,\varepsilon_a}(w) \right| \leq \frac{D_2}{\omega}\zeta. \quad (142)$$

By Lemma 1 and Assumption (A3), $\widehat{w}$ minimizes $\widehat{\mathcal{J}}_{\sigma,\varepsilon_a}$ over $\mathcal{W}_\omega$. Hence

$$\widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(\widehat{w}) \leq \widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}). \quad (143)$$

Thus,

$$\mathcal{J}_{\sigma,\varepsilon_a}(\widehat{w}) \leq \widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(\widehat{w}) + \frac{D_2}{\omega}\zeta \leq \widehat{\mathcal{J}}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}) + \frac{D_2}{\omega}\zeta \leq \mathcal{J}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}) + \frac{2D_2}{\omega}\zeta. \quad (144)$$

□

Lemma 5. *For any $w \in \mathcal{W}_\omega$,*

$$|\mathcal{J}_{\sigma,\varepsilon_a}(w) - \mathcal{J}(w)| \leq \frac{D_2}{\omega}\beta_{\sigma,\varepsilon_a}. \quad (145)$$

Consequently, on the event of Lemma 4,

$$\mathcal{J}(\widehat{w}) \leq \mathcal{J}(w^\star) + \frac{2D_2}{\omega}(\zeta + \beta_{\sigma,\varepsilon_a}). \quad (146)$$

Proof. For any $w \in \mathcal{W}_\omega$,

$$|\mathcal{J}_{\sigma,\varepsilon_a}(w) - \mathcal{J}(w)| = \left| \sum_i \frac{(\phi_i^\star - \mathcal{A}_i)\Delta_i^2}{w_i} \right| \leq \sum_i \frac{|\phi_i^\star - \mathcal{A}_i|\Delta_i^2}{w_i} \leq \frac{\beta_{\sigma,\varepsilon_a}}{\omega}\sum_i \Delta_i^2 = \frac{D_2}{\omega}\beta_{\sigma,\varepsilon_a}. \quad (147)$$

Now,

$$\mathcal{J}(\widehat{w}) \le \mathcal{J}_{\sigma,\varepsilon_a}(\widehat{w}) + \frac{D_2}{\omega}\beta_{\sigma,\varepsilon_a}. \tag{148}$$

By Lemma 4,

$$\mathcal{J}_{\sigma,\varepsilon_a}(\widehat{w}) \le \mathcal{J}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}) + \frac{2D_2}{\omega}\zeta. \tag{149}$$

Since $w^\star_{\sigma,\varepsilon_a}$ minimizes $\mathcal{J}_{\sigma,\varepsilon_a}$ over $\mathcal{W}_\omega$,

$$\mathcal{J}_{\sigma,\varepsilon_a}(w^\star_{\sigma,\varepsilon_a}) \le \mathcal{J}_{\sigma,\varepsilon_a}(w^\star). \tag{150}$$

Using the bias bound again,

$$\mathcal{J}_{\sigma,\varepsilon_a}(w^\star) \le \mathcal{J}(w^\star) + \frac{D_2}{\omega}\beta_{\sigma,\varepsilon_a}. \tag{151}$$

Combining the above gives

$$\mathcal{J}(\widehat{w}) \le \mathcal{J}(w^\star) + \frac{2D_2}{\omega}\zeta + \frac{2D_2}{\omega}\beta_{\sigma,\varepsilon_a}. \tag{152}$$

Therefore,

$$\mathcal{J}(\widehat{w}) \le \mathcal{J}(w^\star) + \frac{2D_2}{\omega}(\zeta + \beta_{\sigma,\varepsilon_a}). \tag{153}$$

□

Lemma 6. *For any $w \in \mathcal{W}_\omega$,*

$$W_1(\mu_w, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B}\mathcal{J}(w). \tag{154}$$

Proof. By the definition of the 1-Wasserstein distance,

$$W_1(\mu_w, \mu_1) = \inf_{\pi \in \Pi(\mu_w, \mu_1)} \int \|x - y\| \, d\pi(x, y), \tag{155}$$

where $\Pi(\mu, \nu)$ denotes the set of all couplings of μ and ν. Using the shared-noise coupling

$$\pi_w = (T_w, T)_{\#} p_0, \tag{156}$$

we obtain

$$W_1(\mu_w, \mu_1) \le \mathbb{E}_{x_0 \sim p_0} \|T_w(x_0) - T(x_0)\|. \tag{157}$$

By Assumption (A4),

$$\|T_w(x_0) - T(x_0)\| \le \frac{C_{\mathrm{Eul}}}{B} \sum_i \frac{A_i(x_0)\Delta_i^2}{w_i}. \tag{158}$$

Taking expectation over $x_0 \sim p_0$,

$$\mathbb{E}\|T_w(x_0) - T(x_0)\| \le \frac{C_{\mathrm{Eul}}}{B}\mathbb{E}\left[\sum_i \frac{A_i(x_0)\Delta_i^2}{w_i}\right]. \tag{159}$$

The expectation on the right is exactly $\mathcal{J}(w)$. Therefore,

$$W_1(\mu_w, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B}\mathcal{J}(w). \tag{160}$$

□

We now have all the needed ingredients to prove Theorem 2.

Proof of Theorem 2. By Lemma 3, with probability at least $1 - \delta$,

$$\max_i |\widehat{\phi}_i - \phi_i^\star| \le \zeta_M(\delta). \tag{161}$$

On this high-probability event, Lemma 5 gives

$$\mathcal{J}(\widehat{w}) \le \mathcal{J}(w^\star) + \frac{2D_2}{\omega} \left(\zeta_M(\delta) + \beta_{\sigma,\varepsilon_a}\right). \tag{162}$$

By Lemma 6,

$$W_1(\mu_{\widehat{w}}, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B} \mathcal{J}(\widehat{w}). \tag{163}$$

Combining the two inequalities,

$$W_1(\mu_{\widehat{w}}, \mu_1) \le \frac{C_{\mathrm{Eul}}}{B} \left[\mathcal{J}(w^\star) + \frac{2D_2}{\omega} \left(\zeta_M(\delta) + \beta_{\sigma,\varepsilon_a}\right)\right]. \tag{164}$$

This completes the proof of the main theorem. □

D Experimental Details

D.1 Synthetic Datasets and Model Training

We evaluate SharpEuler on two synthetic two-dimensional distributions: *Branched Manifold* and *Rotated Grid*. These datasets are chosen to distinguish two failure modes of low-budget Euler sampling. Branched Manifold tests whether the sampler resolves thin, curved support, while Rotated Grid tests whether it preserves separated modes without placing excessive mass between them.

For each dataset, we train a small residual MLP flow matching model whose probability path is described by the linear interpolation

$$x_t = (1 - t)x_0 + tx_1, \qquad t \in [0, 1],$$

where x_0 is a noise sample and x_1 is a data sample. The regression target is

$$v^* = x_1 - x_0,$$

corresponding to the forward-time convention described in the main paper, integrating from noise to data. At inference, we sample with the corresponding reverse-time Euler update. All reported samples use EMA weights. The same trained model, random seed, and initial noise samples are used across schedules whenever possible, so that differences are attributable to the timestep grid.

D.2 Schedule Baselines

We compare schedules at fixed number of function evaluations (NFE), with budgets $B \in \{8, 12, 16, 20\}$ for the synthetic experiments. The sampler and model are unchanged across schedules.

Uniform. The Uniform baseline uses an equally spaced descending reverse-time grid from $s = 1$ to $s = 0$.

ETS. The ETS baseline is the entropy-driven schedule of Stancevic et al. (2025), adapted to our flow matching predictor following the BioNeMo MoCo implementation (NVIDIA BioNeMo, 2024). We use the absolute divergence when constructing the cumulative entropy estimate to ensure a monotone schedule.

SharpEuler. SharpEuler constructs a profile-aware density

$$\rho(t) \propto \phi(t)^\gamma,$$

where $\phi(t)$ is the smoothed finite-difference acceleration profile estimated from calibration trajectories. We evaluate $\gamma \in \{0.5, 1.0, 1.5\}$. The choice $\gamma = 0.5$ corresponds to the LTE-equidistributing density derived in Sec. 2. Larger values concentrate the grid more strongly in regions assigned high sharpness by the calibration profile.

D.3 Evaluation Metrics

For the synthetic experiments, we report Density and Coverage (Naeem et al., 2020), which are robust alternatives to the precision-recall metrics of Kynkäänniemi et al. (2019). Both metrics are computed using k-nearest-neighbor (NN) balls around real samples with $k = 5$.

Density. For each generated sample, Density counts how many real samples k-NN balls contain, averaged over generated samples and normalized by k. Higher Density indicates that generated samples lie in regions of high real sample density.

Coverage. Coverage is the fraction of real samples whose k-NN ball contains at least one generated sample. Higher Coverage indicates that the generator reaches a larger fraction of the target support.

Wasserstein-2 distance. We also report squared Wasserstein-2 distance W_2^2, estimated with Sinkhorn-regularized optimal transport (Villani, 2008; Cuturi, 2013). We use $n = 2000$ samples from each distribution and entropic regularization $\varepsilon = 0.01$. Lower W_2^2 indicates a closer distributional match.

D.4 FLUX: Calibration and Inference Details

We evaluate FLUX.1-dev (Black Forest Labs, 2024), a 12B-parameter rectified flow transformer for text-to-image generation, at its native 1024×1024 resolution. All samples are generated in `bfloat16` precision with classifier-free guidance scale 3.5.

We calibrate SharpEuler once using 64 prompts spanning landscapes, urban scenes, wildlife, and abstract content. For each prompt, we run the default 50-step FLUX.1-dev pipeline in `diffusers` (Hugging Face Diffusers, 2024). During calibration, we record the velocity predictions used by `scheduler.step`, take finite differences of consecutive predictions to estimate acceleration along the trajectory, average these estimates across prompts, and smooth the resulting profile with a Gaussian kernel of bandwidth $\sigma = 1.0$ on the 49-point midpoint grid. The calibrated profile is then fixed and reused for all held-out prompts and all inference budgets.

We compare SharpEuler with $\gamma = 0.5$ against two profile-agnostic schedules. The first is the default FLUX pipeline schedule, which applies the resolution-dependent μ-shift used by `diffusers`. The second is a static shifted schedule with $\alpha = 3$, following the SD3-style FlowMatch Euler shift (Esser et al., 2024),

$$t_{\text{shifted}} = \frac{\alpha t}{1 + (\alpha - 1)t}.$$

For SharpEuler, we use a tail-aligned variant in which the final step is replaced by the corresponding pipeline step. This preserves the hand-tuned low-noise tail of the FLUX sampler, where the calibrated finite-difference profile is empirically noisier.

We omit ETS from the FLUX experiments because its flow matching adaptation requires Jacobian-vector products through the transformer at candidate timesteps, which is too expensive at 1024×1024 resolution.

D.5 GPT-5.5 Vision-Judge Protocol

We use GPT-5.5 as a vision judge to compare SharpEuler and the default FLUX pipeline schedule against the same 50-step reference image. For each prompt, the judge receives four inputs: the text prompt, the 50-step reference image, the SharpEuler image, and the Pipeline image. The two candidate images are shown in randomized order to reduce position bias.

The judge returns two types of output. First, it assigns each candidate an independent integer score from 1 to 10, where higher scores indicate closer agreement with the reference image in the intended generative sense. Second, it returns a forced pairwise preference, allowing ties only when the candidates are effectively indistinguishable. The rubric instructs the judge to compare prompt-grounded semantic content, global composition, scene layout, object identity, style, viewpoint, lighting, color palette, and salient spatial relationships. It also instructs the judge not to over-penalize harmless stochastic differences or exact pixel-level deviations.

For each budget, we report the mean SharpEuler score, mean Pipeline score, the paired score difference ($\Delta = \text{score}_{\text{SharpEuler}} - \text{score}_{\text{Pipeline}}$), the standard deviation of Δ across prompts, a paired t-test p-value for the score differences, and the number of pairwise wins for SharpEuler. Statistical tests are computed over prompt-level paired comparisons.

D.6 Pyramid Flow: Calibration and Inference Details

We calibrate one three-stage sharpness profile using eight prompts and a dense 60-step trajectory per stage, then reuse it without modification on held-out prompts. At a total budget of $B = 20$, both uniform Euler and SharpEuler allocate $(7, 7, 6)$ evaluations across the three stages and differ only in their within-stage timestep placement.

D.7 Calibration Cost

SharpEuler incurs a one-time offline calibration cost; inference then uses ordinary Euler integration at the prescribed NFE, with no additional model evaluations. For the synthetic models, calibration uses 2,000 trajectories with 1,000 dense steps. The calibration times measured alongside OSS-ave are reported in Table 4. For FLUX.1-dev, calibration on 64 prompts takes approximately 61 minutes on one A100 GPU. The resulting profile can be reused across prompts and NFE budgets.

E Additional Results

E.1 Sharpness Profiles

Figure 5 shows calibrated sharpness profiles for three synthetic datasets: Branched Manifold (left), Rotated Grid (middle), and Spiral (right). The top row shows the target data distributions. The middle row shows the calibrated sharpness profile together with the resulting timestep grid for $\gamma = 0.5$ at budget $B = 16$, while the bottom row shows the corresponding profile and grid for $\gamma = 1.0$.

The profiles differ across datasets, indicating that the calibrated sharpness signal depends on both the learned model and the geometry of the target distribution. Thus, the induced timestep grids are also dataset dependent. Increasing γ concentrates grid points more strongly in regions assigned with high sharpness, producing visibly different timestep placements even at the same evaluation budget. Since the calibrated profiles are derived from the local behavior of the learned flow, an interesting direction for future work is to study whether the induced timestep densities correlate with uncertainty or ambiguity in the generative trajectory Gupta et al. (2026).

E.2 Gamma Ablation

We ablate the exponent γ in the schedule density $\rho(t) \propto \phi(t)^\gamma$ on Branched Manifold and Rotated Grid. We sweep $\gamma \in \{0.5, 0.8, 1.0, 1.2, 1.5, 1.7, 2.0\}$ and evaluate Density and Coverage at budgets $B \in \{5, 8, 12, 16, 20\}$. Results are shown in Figure 6.

The sweep shows a consistent fidelity–coverage tradeoff. Increasing γ tends to increase Density, because the resulting grid concentrates more evaluations at times with high estimated sharpness. Coverage behaves differently: it usually peaks at an intermediate value, often around $\gamma \in [1.0, 1.2]$, and decreases for larger γ. This indicates that overly concentrated grids can improve local sample fidelity while leaving parts of the target support under-sampled. The effect is strongest at the smallest budget, $B = 5$, where each timestep placement has high leverage, and weakens as the budget increases. Overall, $\gamma = 0.5$ remains the LTE-prescribed default, while $\gamma \in [1.0, 1.5]$ often gives the best empirical Density–Coverage tradeoff on these synthetic datasets.

E.3 Additional Synthetic Baselines

We compare SharpEuler with OSS-ave, an offline schedule optimizer for deterministic ODE sampling, and KTS, a training-free method that modifies the velocity field during sampling. Table 4 reports results at matched budgets of 8 and 16 model evaluations. OSS-ave is optimized separately for each budget, whereas SharpEuler calibrates a continuous profile that can be reused across budgets. KTS requires no offline calibration.

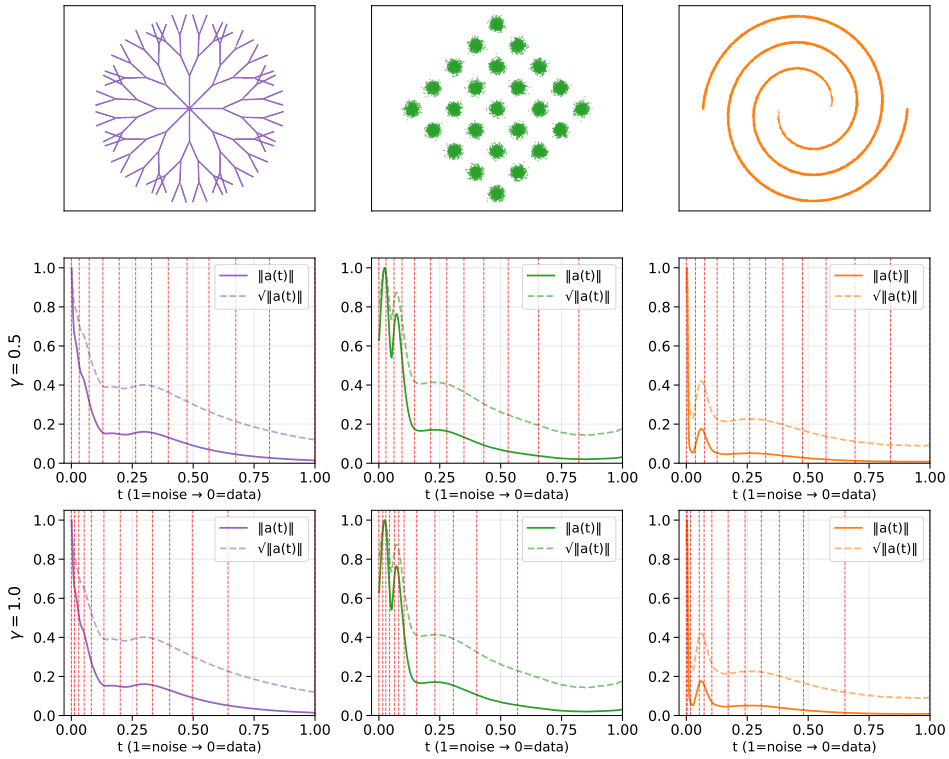
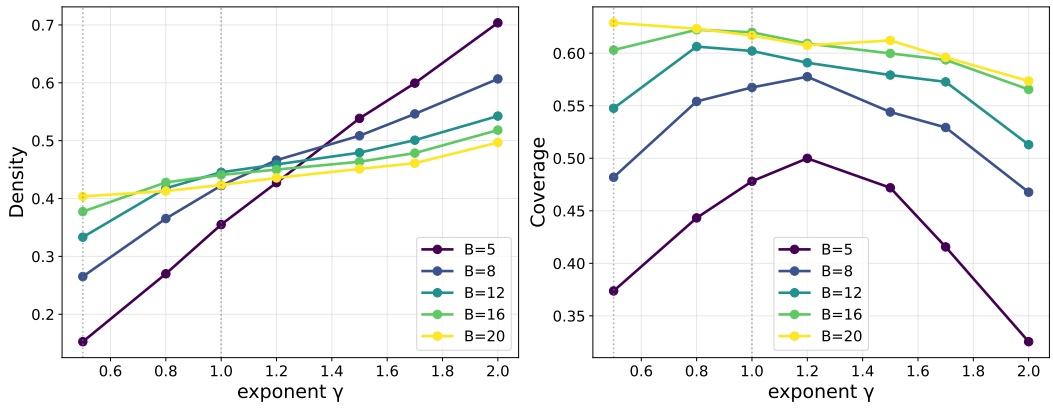


Figure 5: Calibrated sharpness profiles and induced timestep grids for Branched Manifold (left), Rotated Grid (middle), and Spiral (right). The top row shows target samples. The middle and bottom rows show the sharpness profile and resulting $B = 16$ grid for $\gamma = 0.5$ and $\gamma = 1.0$, respectively. The profiles and induced grids vary across datasets, and larger γ concentrates timestep placements more strongly in high-sharpness regions.

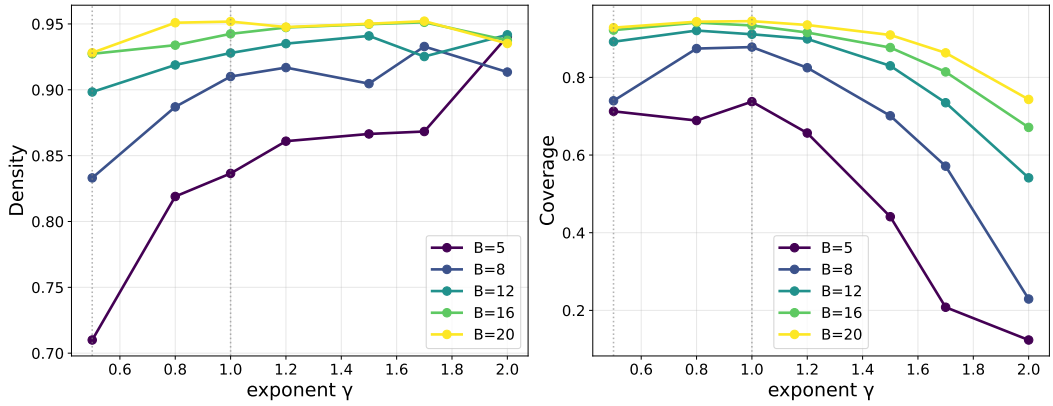
Table 4: Additional baseline comparisons on the synthetic datasets. All methods are evaluated at matched NFE. Higher is better for Density and Coverage; lower is better for W_2^2. Calibration time is reported for methods with an offline calibration stage. Best metric values within each dataset and budget are shown in **bold**.

Dataset	Method	NFE	Density↑	Coverage↑	$W_2^2 \downarrow$	Calibration time
Branched Manifold	SharpEuler	8	**0.888**	**0.771**	0.0527	1.14 s
	OSS-ave	8	0.776	0.656	**0.0519**	16.32 s
	KTS	8	0.769	0.657	0.0565	–
	SharpEuler	16	**0.926**	**0.840**	**0.0395**	1.01 s
	OSS-ave	16	0.818	0.734	0.0397	26.55 s
	KTS	16	0.889	0.822	0.0411	–
Rotated Grid	SharpEuler	8	**0.836**	**0.574**	0.0237	0.83 s
	OSS-ave	8	0.328	0.477	**0.0226**	13.48 s
	KTS	8	0.293	0.456	0.0268	–
	SharpEuler	16	**0.882**	**0.730**	**0.0133**	0.62 s
	OSS-ave	16	0.729	0.568	0.0135	22.32 s
	KTS	16	0.800	0.560	0.0142	–

SharpEuler achieves the highest Density and Coverage in all four dataset–budget settings. It also has the lowest W_2^2 at 16 NFE; at 8 NFE, OSS-ave has a slightly lower W_2^2. The reported calibration times for OSS-ave are higher than those for SharpEuler.



(a) Branched Manifold



(b) Rotated Grid

Figure 6: Density and Coverage as functions of the exponent γ for Branched Manifold and Rotated Grid, across budgets $B \in \{5, 8, 12, 16, 20\}$. Increasing γ generally improves Density, while Coverage is maximized at intermediate values and can decrease when the grid becomes too concentrated. Dashed vertical lines mark $\gamma = 0.5$, the LTE-prescribed choice, and $\gamma = 1.0$.

E.4 Higher Order Solver: Heun's Method

We test whether sharpness-based timestep placement also benefits Heun's method. We recompute the acceleration profile using dense Heun trajectories, construct a fixed grid from that profile, and compare it with a uniform grid at matched NFE. Since Heun uses two velocity evaluations per step, budgets of 8 and 16 NFE correspond to 4 and 8 integration steps, respectively. As shown in Table 5, the sharpness-based grid improves Density and Coverage on both datasets at both budgets. It also reduces W_2^2 on Rotated Grid, while W_2^2 is slightly higher on Branched Manifold. These results suggest that the calibrated grid can help beyond Euler, but they do not establish optimality for Heun: the acceleration-based allocation is motivated by Euler's leading discretization error, and a principled Heun schedule would require a solver-specific error proxy.

E.5 Sensitivity to Calibration-Set Size

We vary the number of calibration trajectories M and report mean and standard deviation across repeated calibration sets in Table 6. On Branched Manifold, Density and Coverage increase as M grows, while their run-to-run variability generally decreases. On Rotated Grid, performance is already stable with a small calibration set. The W_2^2 values show no consistent improvement with increasing M. These results illustrate that the useful calibration size depends on the variability of the trajectory profiles and the target distribution.

Table 5: Heun sampling on the synthetic datasets at matched model evaluation budgets. At 8 and 16 NFE, Heun takes 4 and 8 steps, respectively. The sharpness schedule is calibrated using dense Heun trajectories. Higher is better for Density and Coverage; lower is better for W_2^2. Best values within each dataset and budget are shown in **bold**.

Dataset	Method	NFE	Density↑	Coverage↑	$W_2^2 \downarrow$
Branched Manifold	Uniform + Heun	8	0.703	0.776	**0.022**
	Sharpness + Heun	8	**0.752**	**0.851**	0.024
	Uniform + Heun	16	0.792	0.862	**0.026**
	Sharpness + Heun	16	**0.879**	**0.871**	0.028
Rotated Grid	Uniform + Heun	8	0.448	0.622	0.016
	Sharpness + Heun	8	**0.718**	**0.922**	**0.008**
	Uniform + Heun	16	0.586	0.831	0.009
	Sharpness + Heun	16	**0.931**	**0.840**	**0.007**

Table 6: Sensitivity to the number of calibration trajectories M on the synthetic datasets. Values are mean $\pm$ standard deviation across repeated calibration sets. Higher is better for Density and Coverage; lower is better for W_2^2.

Dataset	M	Density↑	Coverage↑	$W_2^2 \downarrow$
Branched Manifold	2	0.3558 ± 0.0224	0.5746 ± 0.0170	$\mathbf{0.01234 \pm 0.00052}$
	8	0.3647 ± 0.0184	0.5823 ± 0.0106	0.01239 ± 0.00050
	32	0.3743 ± 0.0174	0.5883 ± 0.0122	0.01243 ± 0.00058
	128	$\mathbf{0.3799 \pm 0.0101}$	$\mathbf{0.5908 \pm 0.0094}$	0.01244 ± 0.00057
Rotated Grid	2	0.9228 ± 0.0082	0.9205 ± 0.0071	0.01278 ± 0.00141
	8	0.9198 ± 0.0076	0.9212 ± 0.0052	$\mathbf{0.01263 \pm 0.00121}$
	32	$\mathbf{0.9241 \pm 0.0040}$	$\mathbf{0.9263 \pm 0.0020}$	0.01282 ± 0.00146
	128	0.9222 ± 0.0052	0.9239 ± 0.0035	0.01275 ± 0.00128

E.6 FLUX Evaluation

We evaluate FLUX.1-dev on 1,000 held-out prompts, disjoint from the 64 calibration prompts. KID is our primary distributional metric because its squared-MMD estimator is unbiased at finite sample sizes; we report FID-1k as supporting evidence. Table 7 shows that SharpEuler obtains the lowest value on both metrics.

Table 7: Distributional evaluation of FLUX.1-dev at 1024×1024 on 1,000 held-out prompts. Lower is better.

Method	KID↓	FID-1k↓
Default FLUX pipeline	0.002949	63.758
Shifted α=3	0.002004	59.677
SharpEuler	**0.000537**	**42.117**