FORMALIZING THE SAMPLING DESIGN SPACE OF DIFFUSION-BASED GENERATIVE MODELS VIA ADAPTIVE SOLVERS AND WASSERSTEIN-BOUNDED TIMESTEPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based generative models have achieved remarkable performance across various domains, yet their practical deployment is often limited by high sampling costs. While prior work focuses on training objectives or individual solvers, the broader sampling design problem, specifically solver selection and scheduling, remains largely governed by static heuristics. We propose SDM, a principled, training-free sampling framework that adapts both the numerical solver and the timestep schedule to the intrinsic properties of the diffusion trajectory. By analyzing the PF-ODE dynamics, we show that velocity variation is small in high-noise stages and increases near the data manifold, identifying intervals where solver order is most consequential. In parallel, we introduce an offline-calibrated adaptive scheduling method that explicitly controls the local Wasserstein discretization error and projects the calibrated trajectory to a prescribed NFE budget. We further extend the formulation to a mixed-transition Wasserstein error bound, providing a unified error-propagation view of adaptive scheduling and solver selection within the overall SDM framework. Across standard benchmarks, with extensions to modern ODE samplers, high-resolution synthesis, and text-to-image generation, SDM achieves improved sample quality compared to baseline methods, attaining an FID of 1.93 on CIFAR-10, 2.41 on FFHQ, and 1.98 on AFHQv2, with a reduced number of function evaluations compared to existing samplers. Our code is available at `https://anonymous.4open.science/r/sdm-3128`.

1 INTRODUCTION

Diffusion-based generative models have established themselves as the premier framework for high-fidelity synthesis, defining the state-of-the-art in image generation (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022), video synthesis (Ho et al., 2022; Singer et al., 2022), and 3D content creation (Poole et al., 2022; Lin et al., 2023). Fundamentally, these models cast generation as a differential equation problem: sampling is equivalent to numerically integrating a Probability Flow ODE (PF-ODE) that transports a simple prior distribution to the complex data distribution.

However, this theoretical elegance comes with a practical bottleneck. To minimize discretization error, the ODE must be simulated with fine-grained steps, leading to prohibitively high computational costs (measured in Number of Function Evaluations, NFE). While distillation methods (Salimans & Ho, 2022; Meng et al., 2023) and Consistency Models (Song et al., 2023; Luo et al., 2023; Wang et al., 2024; Kim et al., 2023; Yang et al., 2024) offer few-step generation, they require expensive retraining or fine-tuning pipelines. For the vast majority of pre-trained models, the central challenge remains: *How can we solve the generative ODE as efficiently as possible without modifying the model itself?*

Current approaches to this integration problem are largely dominated by static heuristics. Standard solvers (e.g., Euler, Heun, DPM-Solver++ (Lu et al., 2025)) and schedules (e.g., time-uniform, EDM (Karras et al., 2022)) treat the diffusion trajectory as homogeneous, applying the same numerical precision and step-size logic from start to finish. This ignores a critical geometric reality: the stiffness

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.

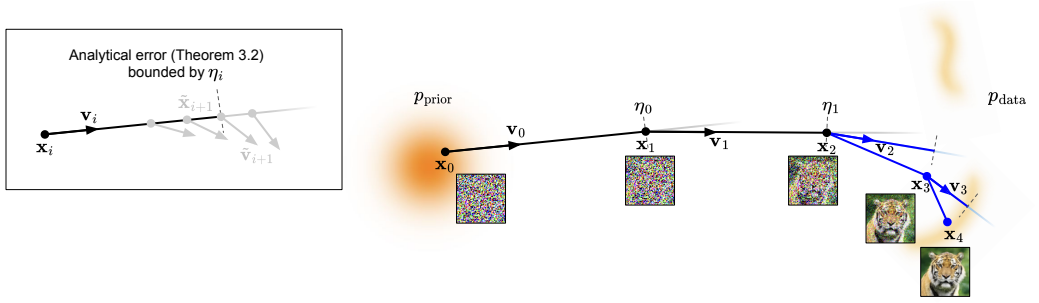


Figure 1: Overview of SDM. **(Inset)** We introduce an adaptive timestep optimization derived from an analytical upper bound on the Wasserstein error. Here, η_i serves as a controllable schedule that explicitly governs the allowable error budget at each step, tightening step sizes where the flow is most sensitive. **(Main)** Complementing this, SDM adapts the solver order to the trajectory's geometry. In the early high-noise regime (black), the flow is nearly linear, allowing efficient low-order integration. As the trajectory approaches the data manifold (blue), the curvature increases, necessitating high-order solvers. These two strategies act as independent and complementary components, jointly improving both sample quality and computational efficiency.

of the diffusion ODE is not constant. As we demonstrate in this work, the trajectory evolves from a nearly linear flow in high-noise regions (dominated by the prior) to a highly non-linear, curved path as it converges onto the low-dimensional data manifold. Treating these distinct regimes with a uniform strategy is inherently suboptimal.

In this work, we formalize the sampling design space through a geometric lens, proposing **SDM**, "**S**ampling Design Space for **D**iffusion **M**odels", a principled framework that adapts the numerical solver and timestep schedule to the intrinsic properties of the diffusion trajectory. Our contributions are two-fold:

(i) Curvature Analysis and Solver Allocation. We provide a rigorous analysis of the PF-ODE curvature under standard parameterizations. As illustrated in Figure 1, the trajectory curvature is negligible during early sampling stages but spikes significantly near the data manifold ($t \to 0$). Based on this, we propose a dynamic allocation strategy that deploys efficient first-order solvers early and transitions to stable higher-order solvers only when geometrically required. We provide theoretical justification for this solver allocation strategy and empirically validate its effectiveness in Section 4.

(ii) Wasserstein-Bounded Adaptive Scheduling. Moving beyond heuristic timestep selection, we introduce an optimization framework grounded in Wasserstein geometry. We derive an adaptive schedule that explicitly bounds the local Wasserstein distance between the discretized trajectory and the true continuous flow. We further introduce an *N-step resampling* procedure that projects this adaptive path onto a fixed NFE budget, allowing flexible control over the quality-efficiency trade-off.

Crucially, SDM is a *training-free* framework. It can be plugged into existing pre-trained diffusion models (e.g., EDM, Stable Diffusion) to improve the Pareto frontier of quality versus efficiency. We demonstrate that SDM achieves state-of-the-art performance across standard benchmarks, including an FID of 1.93 on CIFAR-10, 2.41 on FFHQ, and 1.98 on AFHQv2, surpassing established baselines that rely on static heuristics. Furthermore, we show that SDM can be extended to modern ODE samplers (e.g., DPM-Solver++, UniPC (Zhao et al., 2023)), and scales effectively to higher-resolution and text-to-image generation settings.

2 Preliminaries

2.1 Probability Flow ODE

Diffusion models are a class of generative model framework that gradually injects noise into data through a forward process and subsequently removes noise through a learned reverse process. The following generative process can be formulated as a stochastic differential equation (SDE) parameterized by a noise schedule, where $s(t)$ and $\sigma(t)$ denote the scale and noise term, respectively.

In addition to the stochastic reverse-time SDE, there exists a deterministic *probability flow ODE* that produces the same marginal distribution as the reverse SDE (Song et al., 2020):

$$dx = \left[\frac{\dot{s}(t)}{s(t)} x - s(t)^2 \dot{\sigma}(t) \sigma(t) \nabla_x \log p \left(\frac{x}{s(t)}; \sigma(t) \right) \right] dt. \tag{1}$$

Sampling from diffusion models is performed by discretizing this ODE into a finite number of steps from prior distribution at $t = T$ to the data distribution at $t = 0$ using numerical solvers.

2.2 Numerical Integration and Scheduling

Solving Equation (1) requires discretizing the time interval $[0, T]$ into a sequence $\{t_i\}_{i=0}^N$. The sample quality is governed by two coupled design choices: numerical solver and timestep schedules.

Numerical Solvers. Solvers navigate the trade-off between local precision and computational cost (NFE). First-order methods (e.g., Euler) are computationally cheap (1 NFE/step) but incur a discretization error of $O(h^2)$. To improve precision without increasing steps, high-order solvers such as DPM-Solver++ (Lu et al., 2025), UniPC (Zhao et al., 2023), and DEIS (Zhang & Chen, 2022) employ higher-order Taylor approximations or predictor-corrector schemes. While these methods significantly reduce discretization error ($O(h^k)$ for $k \geq 2$), they typically require complex multi-step history or additional evaluations per step, making their uniform application across the entire trajectory potentially inefficient.

Timestep Scheduling. Diffusion timesteps $\{t_i\}_{i=0}^N$ determines the step size Δt_i and the noise levels $\sigma(t_i)$. Early works employed heuristic time-based schedules, such as linear (Ho et al., 2020) or cosine schedules (Nichol & Dhariwal, 2021). More recently, Karras et al. (Karras et al., 2022) proposed a noise-level based polynomial schedule (EDM), which has become a standard baseline due to its empirical robustness. Subsequent studies have explored timestep optimization based on various criteria including score-based measures (Williams et al., 2024; Xue et al., 2024), KL divergence (Sabour et al., 2024; Park et al., 2024; Li et al., 2023), and Wasserstein-based error estimates (Hu et al., 2024). However, there does not exist a unified framework that integrates solver selection and timestep allocation based on the intrinsic geometry of the diffusion trajectory. Motivated by this gap, we propose a unified sampling design space that characterizes both of these dimensions.

3 SDM: Sampling via Adaptive Solvers and Timestep Schedulers

We investigate the sampling design space of pretrained diffusion models from a principled perspective. While prior works often rely on static heuristics, we propose a geometric framework, ***SDM***, which provides a theoretically grounded for both numerical solver selection and timestep scheduling based on the intrinsic properties of the diffusion trajectory.

In Section 3.1, we analyze the local stiffness of the probability flow ODE. While state-of-the-art exponential integrators (e.g., DPM-Solver++ (Lu et al., 2025)) achieve efficiency by analytically solving the linear component of the diffusion ODE, they fundamentally rely on fixed-order local approximations of the nonlinear residual. Our analysis generalizes this by explicitly quantifying the curvature of this residual, demonstrating that low-order solvers, such as Euler method, are sufficient for high-noise regimes, while higher-order corrections become necessary only in later stages. In Section 3.2, we introduce a Wasserstein-bounded adaptive strategy that optimizes the timestep schedule to minimize the discretization error under a fixed computational budget. We empirically show that this leads to consistent improvements in sample quality.

3.1 Adaptive ODE Solvers

3.1.1 Analysis of ODE curvature

To design an efficient adaptive sampling strategy, we must first quantify the geometric properties of the diffusion trajectory that govern numerical stability. The discretization error of any numerical solver is strictly bounded by the magnitude of the trajectory's higher-order derivatives, specifically the acceleration $\|\ddot{x}\|$. This quantity serves as a direct proxy for the *local stiffness* of the probability flow ODE (Krämer & Hennig, 2024).

Exponential integrators mitigate stiffness by explicitly solving the linear drift term. However, the discretization error is then dominated by the higher-order derivatives of the neural network prediction itself (the non-linear noise residual). Our curvature analysis focuses on this residual complexity, providing a solver-agnostic measure of difficulty that applies regardless of the integrator's baseline order. In Proposition 3.1, we derive the closed-form expression of $\|\ddot{x}\|$ for standard diffusion parameterizations.

Proposition 3.1. *Under EDM/VP/VE parameterizations, the second derivative of the probability flow ODE can be expressed as follows:*

EDM

$$\ddot{x} = -\frac{1}{\sigma^2} J_x(x - D_\theta) - \frac{1}{\sigma} D_\sigma, \tag{2}$$

VP

$$\begin{aligned} \ddot{x} = &\left(\frac{1}{4}B^2 - \frac{1}{2}\beta_d\right) x + \left[\frac{1}{2}\beta_d(1+\sigma^{-2}) - \frac{1}{4}B^2\left(1 + 2\sigma^{-2} + \sigma^{-4}\right)\right](x - sD_\theta) \\ &- \frac{1}{4}B^2 \frac{\sigma + 2\sigma^{-1} + \sigma^{-3}}{\sqrt{1+\sigma^2}} \left(J_x \epsilon_\theta + D_\sigma\right), \end{aligned} \tag{3}$$

VE

$$\ddot{x} = -\frac{1}{4\sigma^4}(I + J_x)(x - D_\theta) - \frac{1}{4\sigma^3} D_\sigma, \tag{4}$$

where J_x denotes the Jacobian of the composed map $D_\theta(x/s(t); \sigma(t))$ with respect to x, and D_σ denotes the partial derivative with respect to σ at fixed $x/s(t)$. Detailed proof and closed-form expressions for B, $\dot{\sigma}$, and $\ddot{\sigma}$ under the VP formulation are provided in Appendix B.

The observed power-law relationship, i.e., $\ddot{x} \propto \sigma^{-p}$ with $p \geq 0$, in Equation (2)–Equation (4) implies that the PF-ODE trajectory becomes increasingly curved as the process approaches the data manifold. This theoretical insight motivates the use of a mixed-solver strategy that progressively adapts the solver order to the trajectory's changing stiffness.

3.1.2 Mixture of ODE Solvers

While Proposition 3.1 highlights the varying stiffness across the diffusion trajectory, utilizing this insight requires a practical metric to allocate solver types efficiently. Computing the exact second derivative $\|\ddot{x}\|$ involves Hessian-vector products, which are computationally prohibitive to evaluate repeatedly. To address this, we use discrete proxies based on vector-field variation, allowing us to construct a solver-order policy without additional model evaluations during inference.

Given diffusion timesteps $\{t_i\}_{i=0}^N$ such that

$$t_0 > t_1 > \cdots > t_N = 0, \tag{5}$$

let $v_{t_i} := v(x_{t_i}, t_i)$ denote the vector field evaluated at the (x_{t_i}, t_i). We define the *absolute local curvature*[1] as

$$\kappa_{\text{abs}}(t_i) := \frac{\|v_{t_{i+1}} - v_{t_i}\|}{\Delta t_i} \approx \|\ddot{x}_{t_i}\|, \tag{6}$$

where $\Delta t_i = t_i - t_{i+1}$. To obtain a scale-invariant measure of stiffness relative to the signal magnitude, we further introduce the *relative local curvature*,

$$\kappa_{\text{rel}}(t_i) := \frac{\kappa_{\text{abs}}(t_i)}{\|v_{t_i}\|}. \tag{7}$$

Computing $\kappa_{\text{rel}}(t_i)$ directly requires requires evaluating the vector field at both (x_{t_i}, t_i) and $(x_{t_{i+1}}, t_{i+1})$, which increases twice the number of function evaluations (NFE). To avoid this overhead, we hence propose a *cache-based curvature* that does not require additional network evaluations. Specifically, we define

$$\widehat{\kappa}_{\text{rel}}(t_i) := \frac{\|v_{t_i} - v_{t_{i-1}}\|}{\Delta \hat{t}_i \|v_{t_{i-1}}\|}, \tag{8}$$

[1] In this work, we use the term "curvature" to refer to the trajectory non-linearity (acceleration $\|\ddot{x}\|$) that governs the discretization error of ODE solvers.

where $\Delta\hat{t}_i$ is the interval between the actual evaluation points used by the EDM sampler. Since $v_{t_{i-1}}$ is computed at the previous timestep and v_{t_i} is required for the current update, the curvature $\widehat{\kappa}_{\mathrm{rel}}(t_i)$ can be obtained by simply reusing the cached evaluations, resulting in NFE $= 1$ per step. A detailed analysis justifying that $\widehat{\kappa}_{\mathrm{rel}}(t_i)$ provides an one-step delayed estimate of the relative local curvature is given in Appendix C. Based on the curvature profile of the sampling trajectory, we propose a *dynamic ODE solver selection* strategy by introducing threshold τ_κ that determines when to switch from low-order to high-order (e.g., Euler/Heun, DPM-Solver++(1M/2M)) solver updates.

3.2 Adaptive Timestep Schedulers

3.2.1 Adaptive step size based on Wasserstein distance

Given two probability distributions μ and ν, the squared 2-Wasserstein distance is defined as:

$$W_2^2(\mu,\nu) := \inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\pi}[\|X-Y\|^2], \tag{9}$$

where $\Pi(\mu,\nu)$ denotes the set of all couplings between μ and ν. In diffusion-based generative models, each numerical integration step induces a transport between probability distributions at adjacent timesteps. Within this framework, the step size directly controls the local Wasserstein distance induced by the sampler. Driven by this insight, we propose an *adaptive step size scheduling* algorithm that explicitly bounds the local Wasserstein distance at each sampling step.

Theorem 3.2 (Local Wasserstein Step-Size Condition)**.** *Given a diffusion timestep t_i and a step size $\Delta t_i > 0$, let $t_{i+1} = t_i - \Delta t_i$. Let $\Phi^*_{t_i\to t_{i+1}}$ denote the exact probability-flow transition from t_i to t_{i+1}, and let*

$$\Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}) = x_{t_i} - \Delta t_i\, v(x_{t_i}, t_i). \tag{10}$$

be the Euler transition. Suppose the along-trajectory velocity $v(x^*_s, s)$ is absolutely continuous on $[t_{i+1}, t_i]$, and define

$$S_i = \left(\mathbb{E}\left[\sup_{s\in[t_{i+1},t_i]} \left\|\frac{d}{ds}v(x^*_s,s)\right\|^2\right]\right)^{1/2}. \tag{11}$$

Then, the sufficient step-size condition

$$\Delta t_i \le \sqrt{\frac{2\eta}{S_i}} \tag{12}$$

ensures that the local Wasserstein distance does not exceed the given upper bound η.

The detailed proof is provided in Appendix D.1. The quantity S_i measures the local variation of the velocity field along the true trajectory. It is worth noting that S_i can be expressed as a function of local curvature measure defined in Equation (6), further motivating the joint design of adaptive solver allocation and timestep scheduling. Since S_i is intractable, we approximate it during an offline calibration phase using a trial Euler step $\Delta t_{\mathrm{trial}} := t_i - \tilde{t}_{\mathrm{trial}}$, evaluated at candidate $(x_{\tilde{t}_{\mathrm{trial}}}, \tilde{t}_{\mathrm{trial}})$:

$$\widehat{S}_i := \frac{\left\|v(x_{\tilde{t}_{\mathrm{trial}}}, \tilde{t}_{\mathrm{trial}}) - v(x_{t_i}, t_i)\right\|}{\Delta t_{\mathrm{trial}}}. \tag{13}$$

The corresponding empirical Euler local transition error proxy is then defined as

$$\hat{\eta}^{\mathsf{E}}_i := \frac{\Delta t^2_{\mathrm{trial}}}{2}\widehat{S}_i. \tag{14}$$

In practice, the timestep schedule is calibrated with the Euler-based local transition error term derived in Equation (14). We empirically find that the following method suffices to generate optimized schedules that yields improved model performance. Furthermore, despite being based on a first-order approximation, it should be noted that the first-order bound provides a conservative estimate that remains valid for higher order solvers, as the error term for second-order method, for instance, scales to $\mathcal{O}(\Delta t^3)$ compared to $\mathcal{O}(\Delta t^2)$ behavior of first-order methods (see Appendix D.2).

η-scheduling. Based on the local Wasserstein error budget η_i, we perform timestep scheduling by selecting the largest step size Δt_i that satisfies the corresponding error bound. The practical realization of our Wasserstein-bounded scheduling adopts a computationally efficient predictor-corrector strategy to enforce the local error budget derived in Theorem 3.2. To mitigate the computational cost of exhaustive search, we initialize the candidate $\tilde{t}_{i+1}$ from a pre-defined reference grid, effectively warm-starting the search procedure with a value proximal to the optimal solution. Given the current timestep t_i and the candidate $\tilde{t}_{i+1}$, we estimate the local velocity variation by $\widehat{S}_i = \|\tilde{v}_{t_{i+1}} - v_{t_i}\|/(t_i - \tilde{t}_{i+1})$. The candidate is then verified by checking whether the resulting local error satisfies the prescribed budget η_i. If the candidate violates the budget or, conversely, proves overly conservative, we refine $\tilde{t}_{i+1}$ using an *exponential backoff mechanism*. By expanding or contracting the step size via multiplicative factors, this procedure ensures logarithmic search complexity $O(\log \frac{\Delta t}{\delta})$ akin to classical backtracking methods like the Armijo-Wolfe conditions (Armijo, 1966; Wolfe, 1969; 1971). We provide the detailed algorithm in Appendix E.1.

Total Wasserstein Error Bound for Adaptive Solvers. We further demonstrate that taking the proposed step-size selection consequently bounds the total Wasserstein error under the mixed-solver strategy introduced in Section 3.1. In particular, as long as each selected numerical transition is Wasserstein-stable, the global Wasserstein error remains bounded by the accumulated local transition error budgets, propagated according to the stability of subsequent transitions. Thus, switching between different numerical solvers does not introduce additional error terms; it only changes the local transition error and the corresponding stability factor at each step. A formal bound and its proof are provided in Appendix D.3.

In practice, the difficulty of the sampling dynamics varies significantly across noise levels. To account for this, we allow the error tolerance term η_i to depend on the current noise level σ_i as the following:

$$\eta_i = (\eta_{\max} - \eta_{\min}) \left(\frac{\sigma_i}{\sigma_{\max}} \right)^p + \eta_{\min}, \tag{15}$$

where $p \geq 0$ controls how strongly the error budget is concentrated throughout different noise levels. The following design assigns a larger tolerance during the early, high-noise stages, while the error bound becomes more restrictive as $\sigma_i \to 0$.

3.2.2 N-STEP RESAMPLING

The proposed scheduling method provides a sequence of diffusion timesteps such that the local Wasserstein error remains below a prescribed threshold, but the resulting number of steps is not directly controlled. To obtain a schedule with a desired number of inference steps N, we therefore perform an additional resampling procedure.

Given a calibrated sequence of diffusion timesteps $\{t_i\}_{i=0}^M$ and corresponding noise levels $\{\sigma_i\}_{i=0}^M$, defined by scheduler σ such that $t_i = \sigma^{-1}(\sigma_i)$, we define an incremental cost $L(t_i, t_{i+1})$ and its corresponding cumulative geodesic length as

$$\Gamma = \sum_{i=0}^{M-1} \sqrt{L(t_i, t_{i+1})}. \tag{16}$$

As shown in prior work (Williams et al., 2024), an optimal N-step schedule can be obtained by uniformly discretizing the cumulative geodesic length, such that

$$\sqrt{L(t_i^*, t_{i+1}^*)} \approx \Gamma/N, \quad i = 0, ..., N-1, \tag{17}$$

We further show that the same conclusion holds and expand the following resampling procedure under a weighted incremental cost setting, as proven in Appendix D.4.

$$\tilde{L}(t_i, t_{i+1}) := w(t_i) L(t_i, t_{i+1}), \qquad w(t_i) > 0. \tag{18}$$

Hence, we define our *weighted* incremental cost function using the empirically measured local error proxy $\hat{\eta}_i$ from Equation (14) as

$$\tilde{L}(t_i, t_{i+1}) := w(t_i) \hat{\eta}_i. \tag{19}$$

Table 1: Quantitative results on unconditional generation for CIFAR-10, FFHQ, and AFHQv2 dataset, measured by FID and NFE. "SDM (Adaptive Scheduling)" denotes the Wasserstein-bounded timestep schedule for the solver indicated in the leftmost column. Rows under "SDM (Adaptive Solver)" use the curvature-based solver selection with the optimized threshold parameter τ_κ. The best FID within each solver block for a given dataset and parameterization are highlighted in **bold**, while the overall best FID in each column are highlighted in **underline**. Overall, the proposed SDM sampler achieves superior performance in both sample quality and computational efficiency over all baselines.

			Unconditional CIFAR-10 32×32		Unconditional FFHQ 64×64		Unconditional AFHQv2 64×64	
			VP	VE	VP	VE	VP	VE
Solver	**Schedule**							
Euler	EDM ($\rho = 7$)		7.61	7.75	4.59	4.76	2.93	3.21
	COS (Williams et al., 2024)		7.43	7.25	4.29	**4.46**	2.89	3.10
	SDM (Adaptive Scheduling)		**6.18**	**6.48**	**4.16**	4.51	**2.71**	**2.97**
		NFE	17	17	39	39	39	39
Heun	EDM ($\rho = 7$)		**1.96**	1.97	2.47	2.59	2.04	2.17
	COS (Williams et al., 2024)		2.08	2.10	2.48	2.61	**2.04**	2.16
	SDM (Adaptive Scheduling)		2.00	**1.95**	**2.41**	**2.52**	2.08	**2.13**
		NFE	35	35	79	79	79	79
SDM (Adaptive Solver)	EDM ($\rho = 7$)		**1.93**	1.97	2.48	**2.58**	**1.98**	**2.10**
		NFE	31	32	72	76	66	67
	SDM (Adaptive Scheduling)		**1.93**	1.97	2.47	**2.58**	**1.97**	**1.99**
		NFE	32	33	76	77	75	72

We then obtain an N-step schedule $\{t_i^*\}_{i=0}^N$ by uniformly discretizing the weighted cumulative geodesic length, which we denote the following procedure as the *N-step resampling* algorithm. For our implementation, we define the weighted function in terms of the noise level σ as

$$g(\sigma_i) := \left(\frac{\sigma_i}{\sigma_{\max}}\right)^{-q}, \qquad w(t_i) = g(\sigma_i)^2, \tag{20}$$

where parameter $q \geq 0$ controls the degree of emphasis on the later steps. Larger values of q allocate a greater fraction of the sampling budget to the low-σ regime. The offline resampling and online inference procedure is provided in Appendix E.1. We additionally provide proof that the Wasserstein-based local error interpretation remains valid after N-step resampling in Appendix D.5.

4 Experiments

4.1 Implementation Details

Datasets, Models, and Metrics. We evaluate our proposed SDM sampler across various standard benchmarks. For unconditional and conditional generation, we conduct experiments on CIFAR-10 (32×32), and FFHQ, AFHQv2, and ImageNet (64×64), following the baseline (Karras et al., 2022). We use pretrained models from EDM (Karras et al., 2022) with no additional training. For higher-resolution generation, we use EDM2-M (Karras et al., 2024) pretrained on ImageNet 512×512. Sample quality and sampling efficiency are measured using FID and the number of function evaluations (NFE), respectively. For text-to-image generation, we employ SDXL (Podell et al., 2024), Stable Diffusion 3 (Esser et al., 2024), and FLUX (Labs et al., 2025) with a guidance scale of 5.0, 7.0, and 3.5, respectively, and evaluate on COCO dataset (Lin et al., 2014), using FID and CLIP scores as our performance metrics.

Baselines We adopt the timestep discretization from the EDM schedule, where the noise levels $\{\sigma_i\}_{i=0}^N$ are defined as

$$\sigma_{i<N} = \left(\sigma_{\max}{}^{\frac{1}{\rho}} + \frac{i}{N-1}(\sigma_{\min}{}^{\frac{1}{\rho}} - \sigma_{\max}{}^{\frac{1}{\rho}})\right)^{\rho}, \sigma_N = 0. \tag{21}$$

with $\rho = 7$. For unconditional and conditional generation settings, we use a fixed number of timesteps for each dataset: 18 steps for CIFAR-10, 40 steps for FFHQ and AFHQv2, and 256 steps for ImageNet. We adopt stochastic settings with $S_{\text{churn}} = 40, S_{\text{min}} = 0.05, S_{\text{max}} = 50$, and $S_{\text{noise}} = 1.003$ for ImageNet. For baseline comparisons, we evaluate both EDM and Corrector Optimized Schedules (Williams et al., 2024), which we abbreviate as COS throughout the paper. We use 32 steps for EDM2 and 10 steps for text-to-image generation settings.

ODE Solvers and Timestep Scheduling. For adaptive solvers, we experiment with combinations of Euler/Heun and DPM-Solver++(1M/2M) (Lu et al., 2025) updates. During the offline scheduling for conditional generation on ImageNet 64×64, we remove the stochastic settings from EDM to accurately estimate the local velocity field variations and construct the Wasserstein bounded timestep sequence. During inference, however, we employ the same stochastic churn parameters as the EDM baseline. We use DPM-Solver++(2M) for SDXL and Euler for Stable Diffusion 3 and FLUX.

Hyperparameters. We set the threshold term τ_κ, Wasserstein error tolerance parameters $\eta_{\text{min}}, \eta_{\text{max}}, p$, and N-step resampling parameter q as our hyperparameters.

4.2 Results

Unconditional Generation. Quantitative results for unconditional generation are reported in Table 1. Across all datasets, solvers, and pre-trained model configurations, SDM consistently improves sample quality over prior baselines, demonstrating robust performance under both VP and VE parameterizations. Under the Euler solver, SDM with adaptive scheduling yields substantial gains. For Heun solver, SDM remains competitive or superior to existing methods, e.g., achieving FID 2.41 on FFHQ and 2.13 on AFHQv2. Applying the adaptive solver alone surpasses EDM baseline with Heun solvers, achieving FID 1.93 on CIFAR-10, 2.58 on FFHQ, and especially reducing the NFE by approximately 15–20% while achieving FID 1.98 on AFHQv2. Most notably, combining both adaptive solver with adaptive scheduling yields the best overall performance, reaching FID 1.97 for VP and 1.99 for VE configuration on AFHQv2.

Extension to Few-Step Generation. To evaluate the practical efficiency of SDM, we evaluate the model performance in few-step generation settings. Tables 7 and 8 present the quantitative results for 10-step and 20-step generation, respectively. Our results demonstrate that SDM consistently outperforms EDM baselines across both FFHQ and AFHQv2 datasets. Most notably, the integration of both adaptive scheduling and adaptive solver yields the strongest performance in terms of FID, confirming that our principled SDM framework remains robust and effective even under highly constrained sampling budgets.

Extension to Modern ODE solvers and High-Resolution Generation. We extend our proposed method in modern ODE solvers and high-resolution generation settings. As shown in Table 6, SDM schedules consistently outperform EDM baselines across standard benchmarks, even when integrated with state-of-the-art high-order solvers. We further validate the effectiveness of both adaptive solver and adaptive scheduling in Table 9, by decomposing the components and evaluating the model performance on ImageNet 512×512. SDM schedules provides the primary performance gain, followed by an additional improvement by adaptively increasing the solver order from 1M to 2M.

Extension to Text-to-Image Generation. We provide quantitative results of our proposed scheduling method on SDXL in Table 10, where SDM outperforms in FID against standard schedules, including time-uniform, EDM, AYS, and AutoDiffusion. We further evaluate SDM on Stable Diffusion 3 and

Table 2: Quantitative results on text-to-image generation with Stable Diffusion 3 (**Left**) and FLUX (**Right**). The baseline timestep schedule, denoted as "Uniform", includes the shifting term used by each corresponding model. SDM remains achieving the best model performance in FID while remaining competitive in CLIP for recent rectified flow-based text-to-image models.

Schedule	FID	CLIP
Uniform	23.41	**0.3162**
SDM	**22.06**	0.3151

Schedule	FID	CLIP
Uniform	24.41	**0.3112**
SDM	**23.87**	0.3102

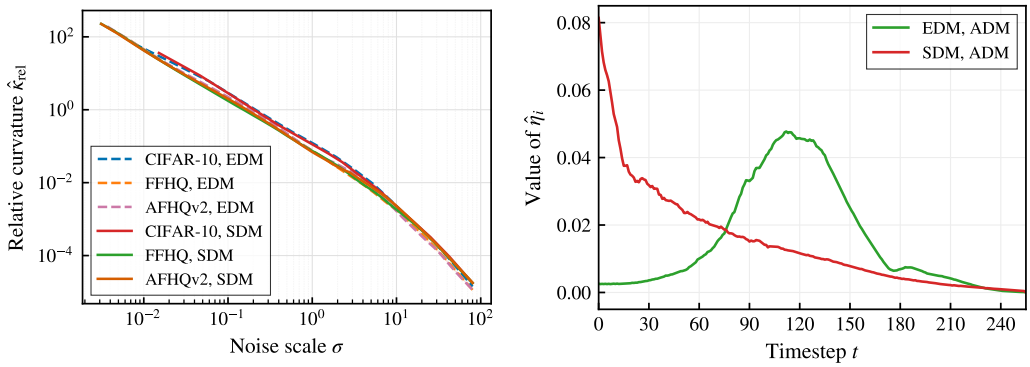


Figure 2: Empirical validation of trajectory geometry and Wasserstein error allocation. (**Left**) Relative curvature $\hat{\kappa}_{\text{rel}}$ as a function of noise level σ for standard benchmarks. The curvature exhibits an approximately linear correlation with noise levels σ in log scale, consistent with the theoretical derivation of second order probability flow ODE. (**Right**) Distribution of local Wasserstein error bound $\hat{\eta}_i$ over diffusion timesteps for ImageNet 64×64. EDM schedules exhibit an initially increasing trend with a subsequent decay, reaching maximum during the intermediate sampling stages. In contrast, SDM schedules allocate a larger portion of the error budget to the early high-noise stages, resulting in improved sample quality.

FLUX in Table 2, which are recent rectified flow-based text-to-image models, demonstrating superior performance in FID over baselines, hence demonstrating its generalizability to modern text-to-image generation settings. We note that quantitative metrics for text-to-image generation are often limited, as discussed in prior works (Podell et al., 2024). We find that improvements in sample quality are more clearly demonstrated through qualitative observation, which we provide in Figures 6 and 7.

4.3 Analysis of $\hat{\eta}_i$

We analyze the evolution of per-step error indicator $\hat{\eta}_i$ for both SDM schedules and EDM schedules. As shown in Figure 2, for EDM schedules, $\hat{\eta}_i$ initially increases during the early sampling stages and decreases afterwards, indicating that there exists unnecessary error budget expended during the intermediate stages, leaving room for additional improvement. In contrast, SDM schedules exhibit a monotonically decreasing trend in $\hat{\eta}_i$ throughout the sampling trajectory, suggesting that SDM schedules efficiently allocates more of the allowable error budget to the early stages of sampling, where dynamics are smoother, while enforcing tighter error control in later stage when the sampling trajectory becomes more nonlinear. As a result, SDM schedules achieve improved sample quality by devoting higher precision to the more critical stages of generation.

5 Conclusion

In this work, we introduce a sampling design space that improves both sample quality and sampling efficiency for pretrained diffusion-based generative models. By formulating the second-order behavior of the PF-ODE, we observe that the trajectory curvature remains negligible during the early sampling stage, which motivates the use of low-order ODE solvers in this regime, while reserving higher-order integration for later stages where the dynamics become more nonlinear. Based on this observation, we propose an adaptive solver that dynamically selects numerical solvers according to the local behavior of the sampling trajectory. In parallel, we introduce an adaptive scheduling method that optimizes diffusion timesteps based on a criterion derived from a Wasserstein error bound. Across extensive experiments against recent baselines, SDM demonstrates strong performance in both sampling quality and computational efficiency compared to baseline methods. Our sampling design currently focuses on optimizing visual quality, leaving the incorporation of text alignment objectives in text-to-image generation as an important direction for future work. Overall, SDM provides a strong and flexible framework for efficient diffusion sampling, and we believe its underlying principles can be extended to broader classes of diffusion-based generative models and applications.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of this paper, including improving clarity, grammar, and phrasing. We have reviewed all AI-assisted work, verifying the accuracy and consistency of the generated content with the underlying research. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide anonymous source code as supplementary material to facilitate the reproduction of our experiments. The paper describes the proposed method, theoretical analysis, and experimental setup, while the supplementary material provides the implementation and additional details needed to reproduce the reported results.

REFERENCES

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

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

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.

Ruiqi Gao et al. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Xixi Hu, Qiang Liu, Xingchao Liu, and Bo Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. *Advances in Neural Information Processing Systems*, 37:138836–138858, 2024.

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.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184. IEEE, 2024.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Nicholas Krämer and Philipp Hennig. Stable implementation of probabilistic ode solvers. *J. Mach. Learn. Res.*, 25(1), January 2024. ISSN 1532-4435.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7105–7114, 2023.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 300–309, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, pp. 1–22, 2025.

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.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14297–14306, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Yong-Hyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. Jump your steps: Optimizing sampling schedule of discrete diffusion models. *arXiv preprint arXiv:2410.07761*, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. *arXiv preprint arXiv:2404.14507*, 2024.

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

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

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

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

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

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

Philip Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

Philip Wolfe. Convergence conditions for ascent methods. ii: Some corrections. *SIAM Review*, 13(2): 185–188, 1971.

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.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

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.

A Additional Preliminaries

A.1 Parameterizations

Since the introduction of diffusion models, various parameterizations of the model output have been proposed. x-prediction and ϵ-prediction formulation has first been introduced (Ho et al., 2020), where the later expressed as a reparameterization of the original training objective. Salimans and Ho (Salimans & Ho, 2022) further proposed v-prediction, where the model predicts a linear combination of data and noise, improving stability and sample quality. Flow-based generative models (Albergo & Vanden-Eijnden, 2022; Lipman et al., 2022; Liu et al., 2022) can also be interpreted within the v-prediction model framework, and recent work has actively explored the connection between diffusion models and flow-based formulations (Esser et al., 2024; Lipman et al., 2022; Gao et al., 2025).

Despite the popularity of alternative parameterizations, recent work (Li & He, 2025) has re-emphasized the importance of x-prediction, stating that directly predicting the clean image is more suitable for modeling low-dimensional manifolds embedded in high-dimensional space. This motivates us to reconsider the design space of EDM (Karras et al., 2022), which establishes a strong baseline by incorporating x-prediction formulation for training and sampling with additional pre-conditioning scheme.

B Derivation of ODE curvature

General PF-ODE second-order derivative. We consider the probability flow ODE (PF-ODE) formulation of diffusion models. Following (Song et al., 2020), the PF-ODE can be written as

$$dx = \left[\frac{\dot{s}}{s}x - s^2\dot{\sigma}\sigma\nabla_x \log p\Big(\frac{x}{s};\sigma\Big)\right] dt. \tag{22}$$

Let $y = x/s$. Under the x-prediction parameterization,

$$\nabla_y \log p(y;\sigma) = \frac{D_\theta(y;\sigma) - y}{\sigma^2}. \tag{23}$$

Hence, using the chain rule, we obtain

$$\nabla_x \log p(y;\sigma) = \frac{1}{s}\nabla_y \log p(y;\sigma). \tag{24}$$

which yields the equivalent ODE

$$\dot{x} = \frac{\dot{s}}{s}x + \frac{\dot{\sigma}}{\sigma}\left(x - sD_\theta(x/s;\sigma)\right). \tag{25}$$

Defining

$$A = \frac{\dot{s}}{s}, \quad \epsilon_\theta := \frac{x - sD_\theta(x/s;\sigma)}{\sigma}, \tag{26}$$

we obtain the following

$$\dot{x} = Ax + \dot{\sigma}\epsilon_\theta. \tag{27}$$

Derivation of $\dot{\epsilon}_\theta$. Recall from Equation (26) that

$$\epsilon_\theta = \frac{x - sD_\theta(x/s,\sigma)}{\sigma}, \qquad N := x - sD_\theta(x/s,\sigma), \tag{28}$$

so that $\epsilon_\theta = N/\sigma$. Differentiating yields

$$\dot{\epsilon}_\theta = \frac{\dot{N}\sigma - N\dot{\sigma}}{\sigma^2} = \frac{\dot{N}}{\sigma} - \frac{\dot{\sigma}}{\sigma}\epsilon_\theta. \tag{29}$$

Meanwhile, due to the definition of an auxiliary variable y,

$$\dot{y} = \frac{1}{s}\left(\dot{x} - \frac{\dot{s}}{s}x\right) = \frac{\dot{\sigma}}{s}\epsilon_\theta, \tag{30}$$

where the second inequality holds from $\dot{x} = \frac{\dot{s}}{s}x + \dot{\sigma}\epsilon_\theta$.

Therefore, differentiating $N = x - sD_\theta$ gives

$$\begin{aligned}\dot{N} &= \dot{x} - \dot{s}D_\theta - s\dot{D}_\theta \\ &= \dot{x} - \dot{s}D_\theta - s\left(J_y\dot{y} + D_\sigma\dot{\sigma}\right) \\ &= \dot{x} - \dot{s}D_\theta - sJ_x\dot{\sigma}\epsilon_\theta - sD_\sigma\dot{\sigma},\end{aligned} \tag{31}$$

where $J_x := \nabla_x D_\theta(x/s;\sigma)$ and $D_\sigma := \partial_\sigma D_\theta(x/s;\sigma)$.

Substituting Equation (31) into Equation (29) yields

$$\begin{aligned}\dot{\epsilon}_\theta &= \frac{\dot{x} - \dot{s}D_\theta - sJ_x\dot{\sigma}\epsilon_\theta - sD_\sigma\dot{\sigma}}{\sigma} - \frac{\dot{\sigma}}{\sigma}\epsilon_\theta \\ &= \frac{\left(\frac{\dot{s}}{s}x + \dot{\sigma}\epsilon_\theta\right) - \dot{s}D_\theta - sJ_x\dot{\sigma}\epsilon_\theta - sD_\sigma\dot{\sigma}}{\sigma} - \frac{\dot{\sigma}}{\sigma}\epsilon_\theta \\ &= \frac{\frac{\dot{s}}{s}(x - sD_\theta)}{\sigma} + \frac{\dot{\sigma}}{\sigma}\epsilon_\theta - \frac{s\dot{\sigma}}{\sigma}J_x\epsilon_\theta - \frac{s\dot{\sigma}}{\sigma}D_\sigma - \frac{\dot{\sigma}}{\sigma}\epsilon_\theta \\ &= \frac{\dot{s}}{s}\epsilon_\theta - \frac{s\dot{\sigma}}{\sigma}J_x\epsilon_\theta - \frac{s\dot{\sigma}}{\sigma}D_\sigma.\end{aligned} \tag{32}$$

Derivation of $\ddot{x}$. Taking the time derivative yields the second-order ODE

$$\ddot{x} = \dot{A}x + A\dot{x} + \ddot{\sigma}\epsilon_\theta + \dot{\sigma}\dot{\epsilon}_\theta \tag{33}$$

$$= (\dot{A} + A^2)x + (A\dot{\sigma} + \ddot{\sigma})\epsilon_\theta + \dot{\sigma}\dot{\epsilon}_\theta, \tag{34}$$

where the first-order derivative term is replaced using Equation (27).

Using $A = \dot{s}/s$, we have

$$\dot{A} + A^2 = \frac{\ddot{s}}{s}, \quad A\dot{\sigma} + \ddot{\sigma} = \ddot{\sigma} + \dot{\sigma}\frac{\dot{s}}{s}. \tag{35}$$

Substituting all terms, we obtain the general curvature expression

$$\ddot{x} = \frac{\ddot{s}}{s}x + \left(\ddot{\sigma} + 2\dot{\sigma}\frac{\dot{s}}{s}\right)\epsilon_\theta - \frac{s\dot{\sigma}^2}{\sigma}J_x\epsilon_\theta - \frac{s\dot{\sigma}^2}{\sigma}D_\sigma. \tag{36}$$

B.1 EDM FORMULATION

For EDM formulation (Karras et al., 2022), we set

$$\sigma(t) = t, \quad s(t) = 1. \tag{37}$$

Note that although EDM discretizes sampling timesteps based on ρ-schedule in the noise domain, the underlying probability flow ODE is parameterized directly by the noise level. Hence,

$$\dot{\sigma} = 1, \quad \ddot{\sigma} = 0, \quad \dot{s} = \ddot{s} = 0. \tag{38}$$

Substituting into Equation (36), the second derivative simplifies to

$$\ddot{x} = -\frac{1}{\sigma}(J_x\epsilon_\theta + D_\sigma) = -\frac{1}{\sigma^2}J_x(x - D_\theta) - \frac{1}{\sigma}D_\sigma. \tag{39}$$

B.2 VP FORMULATION

For the VP formulation, we use the parameterization

$$\sigma(t) = \sqrt{e^{u(t)} - 1}, \quad s(t) = \frac{1}{\sqrt{1 + \sigma(t)^2}}, \quad u(t) = \frac{1}{2}\beta_d t^2 + \beta_{\min} t. \tag{40}$$

For simplicity, we define

$$B(t) := \dot{u}(t) = \beta_{\min} + \beta_d t, \quad \dot{B}(t) = \beta_d. \tag{41}$$

Since $1 + \sigma(t)^2 = e^{u(t)}$, substituting to Equation (40), we have

$$s(t) = \frac{1}{\sqrt{1+\sigma(t)^2}} = e^{-u(t)/2}. \tag{42}$$

Derivatives of $\sigma(t)$. By differentiating $\sigma(t) = \sqrt{e^{u(t)} - 1}$, we obtain

$$\dot{\sigma} = \frac{e^u \dot{u}}{2\sqrt{e^u - 1}} = \frac{(1+\sigma^2)B}{2\sigma} = \frac{1}{2} B \left(\sigma + \sigma^{-1}\right). \tag{43}$$

Hence, we derive the second derivative of $\sigma(t)$ as

$$\begin{aligned} \ddot{\sigma} &= \frac{1}{2}\dot{B}(\sigma + \sigma^{-1}) + \frac{1}{2} B \frac{d}{dt}(\sigma + \sigma^{-1}) \\ &= \frac{1}{2}\beta_d(\sigma + \sigma^{-1}) + \frac{1}{2} B \dot{\sigma}(1 - \sigma^{-2}). \end{aligned} \tag{44}$$

Substituting $\dot{\sigma}$ from (46) gives

$$\ddot{\sigma} = \frac{1}{2}\beta_d(\sigma + \sigma^{-1}) + \frac{1}{4} B^2 (\sigma - \sigma^{-3}). \tag{45}$$

Derivatives of $s(t)$. Using $s(t) = e^{-u(t)/2}$, we directly obtain

$$\frac{\dot{s}}{s} = -\frac{1}{2}\dot{u} = -\frac{1}{2} B, \tag{46}$$

hence

$$\dot{s} = -\frac{1}{2} B s. \tag{47}$$

Taking the second derivative gives

$$\ddot{s} = \frac{d}{dt}\left(-\frac{1}{2} B s\right) = -\frac{1}{2}\dot{B} s - \frac{1}{2} B \dot{s} = -\frac{1}{2}\beta_d s + \frac{1}{4} B^2 s, \tag{48}$$

In other words,

$$\frac{\ddot{s}}{s} = \frac{1}{4} B^2 - \frac{1}{2}\beta_d. \tag{49}$$

Substitution into Equation (36). Substituting Equation (46) to our general PF-ODE second order expression, we obtain the following terms as

$$\begin{aligned} \ddot{\sigma} + 2\dot{\sigma}\frac{\dot{s}}{s} = \ddot{\sigma} - B\dot{\sigma} &= \frac{1}{2}\beta_d(\sigma + \sigma^{-1}) - \frac{1}{4} B^2 (\sigma + 2\sigma^{-1} + \sigma^{-3}), \\ \frac{s\dot{\sigma}^2}{\sigma} &= \frac{1}{4} B^2 \frac{\sigma + 2\sigma^{-1} + \sigma^{-3}}{\sqrt{1+\sigma^2}}. \end{aligned} \tag{50}$$

As a result, under VP parameterization we get

$$\begin{aligned} \ddot{x} = &\left(\frac{1}{4} B^2 - \frac{1}{2}\beta_d\right) x + \left[\frac{1}{2}\beta_d(1 + \sigma^{-2}) - \frac{1}{4} B^2 \left(1 + 2\sigma^{-2} + \sigma^{-4}\right)\right](x - s D_\theta) \\ &- \frac{1}{4} B^2 \frac{\sigma + 2\sigma^{-1} + \sigma^{-3}}{\sqrt{1+\sigma^2}} \left(J_x \epsilon_\theta + D_\sigma\right), \end{aligned} \tag{51}$$

where B, $\dot{\sigma}$, and $\ddot{\sigma}$ are defined in Equation (41), Equation (43), and Equation (45), respectively.

B.3 VE FORMULATION

For VE formulation,

$$\sigma(t) = \sqrt{t}, \quad s(t) = 1, \tag{52}$$

Hence,

$$\dot{\sigma} = \frac{1}{2\sqrt{t}} = \frac{1}{2\sigma}, \quad \ddot{\sigma} = \frac{d}{dt}(\frac{1}{2\sigma}) = \frac{d}{d\sigma}(\frac{1}{2\sigma})\dot{\sigma} = -\frac{1}{4\sigma^3}, \quad \dot{s} = \ddot{s} = 0. \tag{53}$$

Substituting into equation 36, we obtain

$$\ddot{x} = -\frac{1}{4\sigma^3}(\epsilon_\theta + J_x \epsilon_\theta + D_\sigma) = -\frac{1}{4\sigma^4}(I + J_x)(x - D_\theta) - \frac{1}{4\sigma^3} D_\sigma. \tag{54}$$

C JUSTIFICATION OF THE PROXY CURVATURE ESTIMATOR $\widehat{\kappa}_{\mathrm{rel}}(t_i)$

Given the following definition of relative local curvature $\kappa_{\mathrm{rel}}(t_i)$ and its corresponding estimator $\widehat{\kappa}_{\mathrm{rel}}(t_i)$:

$$\kappa_{\mathrm{rel}}(t_i) := \frac{\|v_{t_{i+1}} - v_{t_i}\|}{\Delta t_i \|v_{t_i}\|}, \quad \widehat{\kappa}_{\mathrm{rel}}(t_i) := \frac{\|v_{t_i} - v_{t_{i-1}}\|}{\Delta \hat{t}_i \|v_{t_{i-1}}\|}, \tag{55}$$

we would like to prove the following equation

$$\kappa_{\mathrm{rel}}(t_{i-1}) = \widehat{\kappa}_{\mathrm{rel}}(t_i). \tag{56}$$

In other words, $\widehat{\kappa}_{\mathrm{rel}}(t_i)$ is a one-step delayed evaluation of $\kappa_{\mathrm{rel}}(t_i)$, hence being a valid estimator.

Specifically if we consider the case $S_{\text{churn}} = 0$ following the notation in (Karras et al., 2022), no intermediate perturbation is introduced, hence the sampler evaluates the drift term exactly at the grid points, i.e.,

$$\hat{x}_{t_i} = x_{t_i}, \quad \hat{t}_i = t_i. \tag{57}$$

Note that $(\hat{x}_{t_i}, \hat{t}_i)$ denotes the actual evaluation point used by the EDM sampler to compute the velocity v_{t_i}. Hence, the corresponding velocity field evaluations are numerically identical:

$$v(\hat{x}_{t_i}, \hat{t}_i) = v(x_{t_i}, t_i) = v_{t_i}. \tag{58}$$

Moreover, since

$$\Delta \hat{t}_i = t_i - t_{i+1} = \Delta t_i \tag{59}$$

the effective step size also remains unchanged, hence the proof.

D ADDITIONAL PROOFS

D.1 PROOF OF THEOREM 3.2

We provide a detailed proof of Theorem 3.2, which establishes a sufficient condition on step size Δt such that the Wasserstein distance between the true diffusion trajectory and its numerical approximation remains bounded. Let $\Phi^*_{t_i \to t_{i+1}}$ denote the exact PF-ODE transition map and $\Phi^{\mathsf{E}}_{t_i \to t_{i+1}}$ the Euler approximation. For a given state x_{t_i}, the exact and numerical updates are given by

$$x^*_{t_{i+1}} = \Phi^*_{t_i \to t_{i+1}}(x_{t_i}), \quad x^{\mathsf{E}}_{t_{i+1}} = \Phi^{\mathsf{E}}_{t_i \to t_{i+1}}(x_{t_i}). \tag{60}$$

Using the integral form of the ODE, we write

$$\Phi^*_{t_i \to t_{i+1}}(x_{t_i}) = x_{t_i} + \int_{t_i}^{t_{i+1}} v(x_\tau, \tau)\, d\tau, \tag{61}$$

and the Euler update as

$$\Phi^{\mathsf{E}}_{t_i \to t_{i+1}}(x_{t_i}) = x_{t_i} - \Delta t_i\, v(x_{t_i}, t_i). \tag{62}$$

The resulting discretization error is

$$\Phi^*_{t_i \to t_{i+1}}(x_{t_i}) - \Phi^{\mathsf{E}}_{t_i \to t_{i+1}}(x_{t_i}) = \int_{t_i}^{t_{i+1}} \big(v(x_\tau, \tau) - v(x_{t_i}, t_i)\big)\, d\tau. \tag{63}$$

Meanwhile, we define the auxiliary function

$$g(\tau) := v(x_\tau, \tau). \tag{64}$$

Then, by the fundamental theorem of calculus,

$$g(\tau) - g(t_i) = \int_{t_i}^{\tau} g'(s)\, ds, \tag{65}$$

which implies the bound

$$\|g(\tau) - g(t_i)\| \le |\tau - t_i| \sup_{s \in [t_{i+1}, t_i]} \|g'(s)\|. \tag{66}$$

Substituting this bound into Equation (63) yields

$$\begin{aligned}\left\|\Phi^*_{t_i\to t_{i+1}}(x_{t_i}) - \Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i})\right\| &\le \left|\int_{t_i}^{t_{i+1}} (\tau - t_i)\, d\tau\right| \sup_{s\in[t_{i+1},t_i]} \left\|\frac{d}{ds} v(x_s, s)\right\| \\ &= \frac{\Delta t_i^2}{2} \sup_{s\in[t_{i+1},t_i]} \left\|\frac{d}{ds} v(x_s, s)\right\|.\end{aligned} \tag{67}$$

We now derive the discretization error in relation to the 2-Wasserstein distance. Recall the definition of Wasserstein distance between two probability measures μ and ν:

$$W_2(\mu,\nu)^2 = \inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\pi}\left[\|X - Y\|^2\right]. \tag{68}$$

Let $x_t \sim p_t$ denote the random variable following the diffusion trajectory at time t. We then define the coupled random variables

$$x^*_{t_{i+1}} = \Phi^*_{t_i\to t_{i+1}}(x_{t_i}), \quad x^{\mathsf{E}}_{t_{i+1}} = \Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}). \tag{69}$$

where $x^*_{t_{i+1}}, x^{\mathsf{E}}_{t_{i+1}}$ corresponds to the exact and Euler-integrated flows, respectively. Due to the definition of Wasserstein distance, we obtain

$$W_2\left(p^*_{t_{i+1}}, p^{\mathsf{E}}_{t_{i+1}}\right) \le \left(\mathbb{E}\left[\left\|x^*_{t_{i+1}} - x^{\mathsf{E}}_{t_{i+1}}\right\|^2\right]\right)^{1/2} \le \frac{\Delta t_i^2}{2}\left(\mathbb{E}\left[\sup_{s\in[t_{i+1},t_i]}\left\|\frac{d}{ds} v(x^*_s, s)\right\|^2\right]\right)^{1/2}. \tag{70}$$

Therefore, requiring the Wasserstein distance to be bounded by a specified tolerance η,

$$W_2\left(p^*_{t_{i+1}}, p^{\mathsf{E}}_{t_{i+1}}\right) \le \eta, \tag{71}$$

leads to the following sufficient condition on the step size:

$$\Delta t_i \le \sqrt{\frac{2\eta}{S_i}}, \tag{72}$$

where

$$S_i = \left(\mathbb{E}\left[\sup_{s\in[t_{i+1},t_i]}\left\|\frac{d}{ds} v(x^*_s, s)\right\|^2\right]\right)^{1/2}. \tag{73}$$

Hence, the proof.

D.2 Extension of Theorem 3.2 for second-order methods

We extend the analysis to the second-order Heun method. Let $\Phi^*_{t_i\to t_{i+1}}$ denote the exact PF-ODE transition, $\Phi^{\mathsf{E}}_{t_i\to t_{i+1}}$ the Euler predictor, and $\Phi^{\mathsf{H}}_{t_i\to t_{i+1}}$ the Heun update. For a given state x_{t_i}, the Heun update is defined as

$$\Phi^{\mathsf{H}}_{t_i\to t_{i+1}}(x_{t_i}) = x_{t_i} - \frac{\Delta t_i}{2}\left(v(x_{t_i}, t_i) + v\left(\Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}),\, t_{i+1}\right)\right). \tag{74}$$

The discretization error can be written as

$$\begin{aligned}\Phi^*_{t_i\to t_{i+1}}(x_{t_i}) - \Phi^{\mathsf{H}}_{t_i\to t_{i+1}}(x_{t_i}) = &\left(\int_{t_i}^{t_{i+1}} v(x_\tau, \tau)\, d\tau + \frac{\Delta t_i}{2}\left[v(x_{t_i}, t_i) + v(x_{t_{i+1}}, t_{i+1})\right]\right) \\ &- \frac{\Delta t_i}{2}\left(v(x_{t_{i+1}}, t_{i+1}) - v\left(\Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}),\, t_{i+1}\right)\right).\end{aligned} \tag{75}$$

For the first term, defining $g(\tau) := v(x_\tau, \tau)$ as in Equation (64) and applying integration by parts twice, we obtain

$$\int_{t_i}^{t_{i+1}} g(\tau)\, d\tau + \frac{\Delta t_i}{2}\left[g(t_i) + g(t_{i+1})\right] = \frac{1}{2}\int_{t_i}^{t_{i+1}} (\tau - t_i)(t_{i+1} - \tau) g''(\tau)\, d\tau. \tag{76}$$

Hence,

$$\left\| \int_{t_i}^{t_{i+1}} g(\tau)d\tau + \frac{\Delta t_i}{2}\left[g(t_i) + g(t_{i+1})\right] \right\| \leq \frac{1}{2}\left| \int_{t_i}^{t_{i+1}} (\tau - t_i)(t_{i+1} - \tau)\, d\tau \right| \sup_{s\in[t_{i+1},t_i]} \|g''(s)\| = \frac{\Delta t_i^3}{12} \sup_{s\in[t_{i+1},t_i]} \|g''(s)\|. \tag{77}$$

Meanwhile, assuming that $v(\cdot, t_{i+1})$ is L-Lipschitz,

$$\left\| v(x_{t_{i+1}}, t_{i+1}) - v\left(\Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}), t_{i+1}\right) \right\| \leq L \left\| \Phi^{*}_{t_i\to t_{i+1}}(x_{t_i}) - \Phi^{\mathsf{E}}_{t_i\to t_{i+1}}(x_{t_i}) \right\|. \tag{78}$$

Using the Euler error bound from Theorem 3.2 and substituting into Equation (75) yields

$$\left\| \Phi^{*}_{t_i\to t_{i+1}}(x_{t_i}) - \Phi^{\mathsf{H}}_{t_i\to t_{i+1}}(x_{t_i}) \right\| \leq \Delta t_i^3 \left(\frac{1}{12} \sup_{s\in[t_{i+1},t_i]} \|g''(s)\| + \frac{L}{4} \sup_{s\in[t_{i+1},t_i]} \|g'(s)\| \right). \tag{79}$$

Finally, let $x_t \sim p_t$ and define

$$x^{*}_{t_{i+1}} = \Phi^{*}_{t_i\to t_{i+1}}(x_{t_i}), \quad x^{\mathsf{H}}_{t_{i+1}} = \Phi^{\mathsf{H}}_{t_i\to t_{i+1}}(x_{t_i}). \tag{80}$$

Then, the Wasserstein distance bound can be derived as

$$W_2\left(p^{*}_{t_{i+1}}, p^{\mathsf{H}}_{t_{i+1}}\right) \leq \left(\mathbb{E}\left[\left\| x^{*}_{t_{i+1}} - x^{\mathsf{H}}_{t_{i+1}} \right\|^2 \right] \right)^{1/2} \leq \Delta t_i^3 \left(\frac{Q_i}{12} + \frac{LS_i}{4} \right), \tag{81}$$

where

$$S_i = \left(\mathbb{E}\left[\sup_{s\in[t_{i+1},t_i]} \left\| \frac{d}{ds} v(x^{*}_s, s) \right\|^2 \right] \right)^{1/2}, \quad Q_i = \left(\mathbb{E}\left[\sup_{s\in[t_{i+1},t_i]} \left\| \frac{d^2}{ds^2} v(x^{*}_s, s) \right\|^2 \right] \right)^{1/2}. \tag{82}$$

D.3 Proof of Global Wasserstein Error Bound for Adaptive Solvers

Proposition D.1 (Total Wasserstein Error Bound for Adaptive Solvers)**.** *Let $\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}}$ be the selected numerical transition map, $(\cdot)_{\#}$ the pushforward operator, and $\mathcal{P}_2(\mathcal{Z}_i)$ the set of probability measures with finite second moment. Assume that each selected numerical transition is Wasserstein-stable, i.e.,*

$$W_2\left((\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}})_{\#}\alpha,\, (\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}})_{\#}\beta \right) \leq \Lambda_i W_2(\alpha, \beta), \quad \forall \alpha, \beta \in \mathcal{P}_2(\mathcal{Z}_i), \tag{83}$$

for some $\Lambda_i < \infty$. Then, the total Wasserstein error satisfies

$$W_2\left((\pi_N)_{\#}\mu_N,\, (\pi_N)_{\#}\nu_N \right) \leq e_0 \prod_{j=0}^{N-1} \Lambda_j + \sum_{i=0}^{N-1} \eta_i \prod_{j=i+1}^{N-1} \Lambda_j, \tag{84}$$

where $e_0 := W_2(\mu_0, \nu_0)$ and μ_i and ν_i denote the distributions obtained by applying the exact and numerical transition maps up to timestep t_i, respectively.

We first introduce the necessary notation. For one-step solvers, the state is defined as $z_{t_i} = x_{t_i}$. For finite-memory solvers, we define $z_{t_i} = (x_{t_i}, r_{t_i}) \in \mathcal{Z}_i$, where r_{t_i} denotes the solver history. We assume that each $\mathcal{Z}_i$ is a Polish space equipped with a metric, all measures belong to $\mathcal{P}_2(\mathcal{Z}_i)$, and all transition maps are measurable. The projection onto the sample coordinate is denoted by $\pi_i(z_{t_i}) = x_{t_i}$, which is assumed to be 1-Lipschitz.

Let $\Phi^{*}_{t_i\to t_{i+1}} : \mathcal{Z}_i \to \mathcal{Z}_{i+1}$ denote the lifted exact PF-ODE transition and $\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}} : \mathcal{Z}_i \to \mathcal{Z}_{i+1}$ the selected numerical transition. We define

$$\mu_{i+1} = (\Phi^{*}_{t_i\to t_{i+1}})_{\#}\mu_i, \quad \nu_{i+1} = (\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}})_{\#}\nu_i, \tag{85}$$

and the local transition error as

$$\eta_i := W_2\left((\Phi^{*}_{t_i\to t_{i+1}})_{\#}\mu_i,\, (\Phi^{\mathsf{ODE}}_{t_i\to t_{i+1}})_{\#}\mu_i \right). \tag{86}$$

The lifted exact transition $\Phi^*_{t_i \to t_{i+1}}$ is chosen such that its sample marginal matches the exact PF-ODE flow. Namely, if $\phi^*_{t_i \to t_{i+1}} : \mathbb{R}^d \to \mathbb{R}^d$ denotes the exact PF-ODE transition from t_i to t_{i+1}, then

$$\pi_{i+1} \circ \Phi^*_{t_i \to t_{i+1}} = \phi^*_{t_i \to t_{i+1}} \circ \pi_i. \tag{87}$$

For finite-memory solvers, the lifted distribution μ_i stores the exact history variables, so that the local error term compares the numerical transition against the exact transition under exact input history.

Lemma D.2 (Wasserstein stability from Lipschitz continuity)**.** *Let $T : \mathcal{Z} \to \mathcal{Z}$ be Λ-Lipschitz. Then, for all $\alpha, \beta \in \mathcal{P}_2(\mathcal{Z})$,*

$$W_2(T_\# \alpha, T_\# \beta) \le \Lambda W_2(\alpha, \beta). \tag{88}$$

Proof. Let $\gamma \in \Pi(\alpha, \beta)$ be any coupling. Then $(T, T)_\# \gamma$ is a coupling of $T_\# \alpha$ and $T_\# \beta$. Hence,

$$W_2^2(T_\# \alpha, T_\# \beta) \le \int \|T(z) - T(z')\|^2 \, d\gamma(z, z') \le \Lambda^2 \int \|z - z'\|^2 \, d\gamma(z, z'). \tag{89}$$

Taking the infimum over $\gamma \in \Pi(\alpha, \beta)$ proves the claim. □

Proof of Proposition D.1. Define the local Wasserstein error at timestep t_i as

$$e_i := W_2(\mu_i, \nu_i). \tag{90}$$

Using

$$\mu_{i+1} = (\Phi^*_{t_i \to t_{i+1}})_\# \mu_i, \quad \nu_{i+1} = (\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}})_\# \nu_i, \tag{91}$$

we have

$$\begin{aligned} e_{i+1} &= W_2\left((\Phi^*_{t_i \to t_{i+1}})_\# \mu_i,\, (\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}})_\# \nu_i\right) \\ &\le W_2\left((\Phi^*_{t_i \to t_{i+1}})_\# \mu_i,\, (\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}})_\# \mu_i\right) + W_2\left((\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}})_\# \mu_i,\, (\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}})_\# \nu_i\right), \end{aligned} \tag{92}$$

where the first term is η_i, and the second term is bounded by the assumed Wasserstein stability of $\Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}}$, hence

$$e_{i+1} \le \eta_i + \Lambda_i e_i. \tag{93}$$

Unrolling this recursion yields

$$e_N \le e_0 \prod_{j=0}^{N-1} \Lambda_j + \sum_{i=0}^{N-1} \eta_i \prod_{j=i+1}^{N-1} \Lambda_j. \tag{94}$$

Finally, since π_N is 1-Lipschitz, applying Lemma D.2 with $T = \pi_N$ gives

$$W_2\left((\pi_N)_\# \mu_N,\, (\pi_N)_\# \nu_N\right) \le W_2(\mu_N, \nu_N) = e_N. \tag{95}$$

Combining the two inequalities proves Equation (84). □

Extensions and Practical Considerations. We provide several remarks to clarify how local error modeling and solver selection interact with the proposed Wasserstein error accounting framework.

Remark D.3 (Euler-proxy calibration versus solver-specific defects)*.* The default SDM implementation calibrates the timestep grid using the Euler-based proxy

$$\hat{\eta}_i^{\mathsf{E}} = \frac{\Delta t_i^2}{2} \widehat{S}_i. \tag{96}$$

This proxy is used as a computationally efficient signal to estimate local trajectory difficulty during offline schedule construction. However, it does not represent the true local error of a given solver. In the theoretical bound in Equation (84), the relevant quantity is the actual local defect η_i^m of the selected solver m, when such a bound is available.

Remark D.4 (Heun local transition error)*.* For Heun, under standard smoothness assumptions, Appendix D.2 gives

$$\eta_i^{\mathsf{H}} = \Delta t_i^3 \left(\frac{Q_i}{12} + \frac{L S_i}{4}\right), \tag{97}$$

where S_i and Q_i quantify first- and second-order variations along the trajectory. This example shows how solver-specific transition errors can be incorporated into Proposition D.1.

Remark D.5 (Finite-memory samplers). Multistep and predictor-corrector samplers can be written as one-step maps by augmenting the state. The history variable r_{t_i} may contain previous samples, denoiser outputs, model evaluations, and the corresponding integration variables. Once the update is written as

$$z_{t_{i+1}} = \Phi^{\mathsf{ODE}}_{t_i \to t_{i+1}}(z_{t_i}), \tag{98}$$

Proposition D.1 applies whenever this map satisfies the Wasserstein-stability condition. Solver-specific local error bounds, once computed, can be substituted directly into Equation (84).

Remark D.6 (Timestep-dependent solver allocation). In practice, SDM employs different numerical solvers across timesteps according to the estimated trajectory curvature. As a result, the overall sampler is implemented as a sequence of transition maps, e.g., Euler updates in early stages and Heun updates in later stages. Hence, Proposition D.1 remains valid under adaptive solver sequences, provided the selected transition map at each timestep satisfies the Wasserstein-stability condition.

D.4 Energy-Optimized Geodesic Path for Weighted Incremental Cost

Proposition D.7. *Assume the incremental cost admits the quadratic form*

$$L(t_i, t_{i+1}) \approx \delta(t_i)\Delta t_i^2, \quad \Delta t_i := t_i - t_{i+1}, \tag{99}$$

for some function $\delta(\cdot)$. Then the weighted cost can be written as

$$\tilde{L}(t_i, t_{i+1}) \approx \tilde{\delta}(t_i)\Delta t_i^2, \quad \tilde{\delta}(t) := w(t)\delta(t). \tag{100}$$

Consequently, the optimal schedule $\{t_i^*\}_{i=0}^N$ for the weighted objective traverses the diffusion path a constant speed with respect to the weighted geodesic length

$$\tilde{\Gamma} := \sum_{i=0}^{M-1} \sqrt{\tilde{L}(t_i, t_{i+1})} = \sum_{i=0}^{M-1} \sqrt{w(t_i)}\sqrt{L(t_i, t_{i+1})}, \tag{101}$$

i.e.,

$$\sqrt{\tilde{L}(t_i^*, t_{i+1}^*)} \approx \frac{\tilde{\Gamma}}{N}. \tag{102}$$

Proof. By Equation (18) and Equation (99),

$$\tilde{L}(t_i, t_{i+1}) = w(t_i)L(t_i, t_{i+1}) \tag{103}$$

$$\approx w(t_i)\delta(t_i)\Delta t_i^2 = \tilde{\delta}(t_i)\Delta t_i^2. \tag{104}$$

hence, the proof. The remainder follows by applying the same proof with δ replaced by $\tilde{\delta}$ (Williams et al., 2024). □

D.5 Proof of Wasserstein Error Bound after N-step Resampling

From the quadratic form in Equation (99), the weighted incremental cost satisfies

$$\sqrt{\tilde{L}(t_i, t_{i+1})} \approx \sqrt{\tilde{\delta}(t_i)}\Delta t_i. \tag{105}$$

We therefore define the continuous density-like function

$$\psi(t) := \sqrt{\tilde{\delta}(t)}. \tag{106}$$

The corresponding weighted cumulative geodesic length can then be expressed as

$$\tilde{\Gamma} = \int_0^{t_0} \psi(u)\, du. \tag{107}$$

Since ψ is continuous and monotone, the uniform discretization of the cumulative weighted geodesic length in Section D.4 implies that each resampled interval satisfies

$$\int_{t_{i+1}^*}^{t_i^*} \psi(u)\, du = \frac{\tilde{\Gamma}}{N}, \tag{108}$$

By the mean value theorem, there exists some $\xi_i \in [t^*_{i+1}, t^*_i]$ such that

$$\psi(\xi_i)\Delta t^*_i = \frac{\tilde{\Gamma}}{N}, \tag{109}$$

and hence

$$\Delta t^*_i = \frac{\tilde{\Gamma}}{N\psi(\xi_i)}. \tag{110}$$

Next, from Equation (19) and the definition of ψ, the weighted local error satisfies

$$\psi(t^*_i)^2(\Delta t^*_i)^2 = \tilde{L}(t^*_i, t^*_{i+1}) = w(t^*_i)\hat{\eta}^*_i = \frac{1}{2}w(t^*_i)\widehat{S}_{t^*_i}(\Delta t^*_i)^2 \tag{111}$$

Therefore,

$$\psi(t^*_i)^2 = \frac{1}{2}w(t^*_i)\widehat{S}_{t^*_i}. \tag{112}$$

Substituting Equation (110) into the local error proxy gives

$$\begin{aligned}\hat{\eta}^*_i &= \frac{1}{2}\widehat{S}_{t^*_i}\left(\frac{\tilde{\Gamma}}{N\psi(\xi_i)}\right)^2 \\ &= \frac{\tilde{\Gamma}^2}{N^2}\frac{\widehat{S}_{t^*_i}}{2\psi(\xi_i)^2} = \frac{\tilde{\Gamma}^2}{N^2 w(t^*_i)}\left(\frac{\psi(t^*_i)}{\psi(\xi_i)}\right)^2,\end{aligned} \tag{113}$$

using Equation (112). Finally, assuming that $\log\psi$ is K-Lipschitz, since $\xi_i \in [t^*_{i+1}, t^*_i]$,

$$\left|\log\frac{\psi(t^*_i)}{\psi(\xi_i)}\right| \le K|t^*_i - \xi_i| \le K\Delta t^*_i. \tag{114}$$

Consequently,

$$\left(\frac{\psi(t^*_i)}{\psi(\xi_i)}\right)^2 \le e^{2K\Delta t^*_i}, \tag{115}$$

and thus

$$\hat{\eta}^*_i \le e^{2K\Delta t^*_i}\frac{\tilde{\Gamma}^2}{N^2 w(t^*_i)}. \tag{116}$$

Thus, the local transition error after N-step resampling admits an upper bound of the same form as the original Wasserstein-based error bound, up to a factor controlled by the local variation of ψ, showing that the practical resampling procedure preserves the Wasserstein-based local error interpretation.

E Further Implementation Details

Experiments and scheduling algorithms for unconditional and conditional generation are implemented using a single NVIDIA RTX 5090 GPU, while a single NVIDIA A100 80GB GPU is used for high-resolution generation (EDM2-M) and text-to-image generation (SDXL, Stable Diffusion 3, FLUX).

E.1 Algorithm for Adaptive Scheduling

We provide pseudocode for the overall adaptive scheduling algorithm. The scheduling is performed offline prior to inference, where η-scheduling and N-step resampling are implemented as a two-stage procedure. In practice, the offline scheduling algorithm requires only a single inference pass with a single batch of size 128. The calibration takes approximately 45s for CIFAR-10, 2 minutes for FFHQ and AFHQv2, and 10–30 minutes for the remaining experiments, including high-resolution and text-to-image generation settings. To assure robustness, we evaluate the model stability across independent pilot runs; the resulting FID for CIFAR-10, FFHQ, and AFHQv2 shows a standard deviation of only 0.0015, 0.0019, and 0.0014, respectively, confirming that the offline scheduling method is largely insensitive to initial sample choices.

η-scheduling

Input: Initial $(\mathbf{x}_{t_0}, t_0)$, model $\mathbf{v}_\phi$, error tolerance map η
$i \leftarrow 0$
$\mathbf{v}_{t_0} \leftarrow \mathbf{v}_\phi(\mathbf{x}_{t_0}, t_0)$
while $t_i > 0$ **do**
  $\tilde{t}_{i+1} \leftarrow \textsc{NextTimestep}(t_i)$
  repeat
    $\Delta t_i \leftarrow t_i - \tilde{t}_{i+1}$
    $\tilde{\mathbf{x}}_{t_{i+1}} \leftarrow \mathbf{x}_{t_i} - \Delta t_i \mathbf{v}_{t_i}$
    $\tilde{\mathbf{v}}_{t_{i+1}} \leftarrow \mathbf{v}_\phi(\tilde{\mathbf{x}}_{t_{i+1}}, \tilde{t}_{i+1})$
    $\widehat{S}_i \leftarrow \frac{\|\tilde{\mathbf{v}}_{t_{i+1}} - \mathbf{v}_{t_i}\|}{\Delta t_i}$ ▷ Equation (13)
    $\tilde{t}_{i+1} \leftarrow \textsc{LineSearch}(\widehat{S}_i, \tilde{t}_{i+1}, t_i)$
  until no further change in $\tilde{t}_{i+1}$
  $\Delta t_i \leftarrow \sqrt{2\eta_i / \hat{S}_i}$ ▷ Theorem 3.2
  $t_{i+1} \leftarrow t_i - \Delta t_i$
  $\mathbf{x}_{t_{i+1}} \leftarrow \mathbf{x}_{t_i} - \Delta t_i \mathbf{v}_{t_i}$
  $\mathbf{v}_{t_{i+1}} \leftarrow \mathbf{v}_\phi(\mathbf{x}_{t_{i+1}}, t_{i+1})$
  $i \leftarrow i + 1$
end while
Output: Initial schedule $\{t_i\}$

N-step resampling

Input: Initial schedule $\{t_i\}$, weights $\{w_i\}_{i=0}^{M-1}$, cost function $\{L(t_i, t_{i+1})\}_{i=0}^{M-1}$
for $i \leftarrow 0$ to $M-1$ **do**
  $\tilde{L}(t_i, t_{i+1}) \leftarrow w(t_i) L(t_i, t_{i+1})$
  $\tilde{\Gamma}(t_i) \leftarrow \sum_{j=0}^{i-1} \sqrt{\tilde{L}(t_j, t_{j+1})}$
end for
$\Gamma^{-1}(\cdot) \leftarrow \textsc{Interpolate}(\{\tilde{\Gamma}(t_i)\}_{i=0}^{M-1})$
for $i \leftarrow 0$ to $N-1$ **do**
  $t_i^* \leftarrow \Gamma^{-1}\left(\tilde{\Gamma}(t_{M-1}) \frac{i}{N}\right)$
end for
Output: N-step schedule $\{t_i^*\}$

Inference

Input: Initial $(\mathbf{x}_{t_0^*}, t_0^*)$, model $\mathbf{v}_\phi$, N-step schedule $\{t_i^*\}$
for $i \leftarrow 0$ to $N-1$ **do**
  $\mathbf{x}_{t_{i+1}^*} \leftarrow \textsc{SolverStep}(\mathbf{x}_{t_i^*}, t_i^*; \mathbf{v}_\phi)$
end for
Output: $\mathbf{x}_{t_N^*}$

E.2 Hyperparameter Settings

Table 3 represents the search grids for hyperparameters used throughout our experiments. For the step schedule based adaptive solver, we vary the curvature threshold τ_κ across datasets and timestep schedules. We observe that VP and VE models exhibit similar performance under the same threshold configuration on standard benchmarks, with the exception of AFHQv2, and therefore share the same τ_κ values. For AFHQv2, we select $\tau_\kappa = 2 \times 10^{-4}$ for VP parameterization and $\tau_\kappa = 1 \times 10^{-3}$ for VE parameterization, which yields the best FID performance.

For adaptive timestep scheduling, as shown in Table 4, we tune four hyperparameters, $\eta_{\min}$, $\eta_{\max}$, p, and q, which jointly control the overall Wasserstein error tolerance and its allocation across diffusion timesteps. We perform explicit tuning across different training configurations for CIFAR-10, but the model performance is largely insensitive to the hyperparameter grid. It is also worth noting that the choice of initial grid is not entirely heuristic. Specifically, $(\eta_{\min}, \eta_{\max})$ is determined by the minimum and maximum Wasserstein error values induced by the baseline schedule, thereby preserving the overall error range while allowing q to reallocate the error budget.

Table 3: Parameter search grid for threshold τ_κ for step schedule based adaptive solver selection.

Timestep Schedule	CIFAR-10	FFHQ	AFHQv2	ImageNet	Grid search
EDM	2×10^{-4}	1×10^{-4}	1×10^{-3}	1×10^{-4}	$\{2, 5, 10, 20, 50, 100\} \times 10^{-5}$
SDM	2×10^{-4}	1×10^{-4}	2×10^{-4}(VP) 1×10^{-3}(VE)	1×10^{-4}	$\{2, 5, 10, 20, 50, 100\} \times 10^{-5}$

Table 4: Parameter search grid for Wasserstein error tolerance and N-step resampling parameters for CIFAR-10.

Parameter	CIFAR-10				Grid search
	Uncond. VP	Uncond. VE	Cond. VP	Cond. VE	
$\eta_{\min}$	0.01	0.01	0.01	0.02	0.01, 0.02, 0.03, 0.04, 0.05
$\eta_{\max}$	0.40	0.40	0.40	0.10	0.10, 0.20, 0.30, 0.40, 0.50
p	1.0	1.0	1.0	1.0	0.8, 1.0, 1.2
q	0.1	0.25	0.1	0.25	0.1, 0.25

F Additional Results

In this section, we provide additional results for our proposed SDM sampler.

Table 5: Quantitative results on conditional generation for CIFAR-10 32×32 and ImageNet 64×64, measured by FID and NFE. "SDM (Adaptive Scheduling)" denotes the Wasserstein-bounded timestep schedule for the solver indicated in the leftmost column. Rows under "SDM (Adaptive Solver)" use the curvature-based solver selection with the optimized threshold parameter τ_κ. The best FID within each solver block for a given dataset and parameterization are highlighted in **bold**, while the overall best FID in each column are highlighted in **bold**.

			Conditional CIFAR-10 32×32		Conditional ImageNet 64×64
			VP	VE	ADM
Solver	**Schedule**				
Euler	EDM ($\rho = 7$)		7.09	6.75	3.48
	COS (Williams et al., 2024)		7.05	6.66	3.74
	SDM (Adaptive Scheduling)		**6.10**	**5.12**	**2.32**
		NFE	17	17	255
Heun	EDM ($\rho = 7$)		**1.82**	**1.78**	1.45
	COS (Williams et al., 2024)		1.92	1.87	**1.40**
	SDM (Adaptive Scheduling)		1.92	1.92	2.25
		NFE	35	35	511
SDM (Adaptive Solver)	EDM ($\rho = 7$)		**1.81**	**1.77**	**1.45**
		NFE	30	32	476
	SDM (Adaptive Scheduling)		**1.81**	**1.77**	2.25
		NFE	32	33	503

F.1 Quantitative Results for Conditional Generation

We report the quantitative results for conditional generation in Table 5. Under the Euler solver, SDM with adaptive scheduling achieves strong gains, obtaining state-of-the-art FID 6.10 and 5.12 on CIFAR-10 with VP and VE based models, respectively, and 2.32 on ImageNet. For adaptive solver settings, SDM further improves both sample quality and sampling efficiency, achieving FID 1.81 and 1.77 on CIFAR-10 with VP and VE based models, respectively, and 1.45 on ImageNet. Conditional generation is more setting-dependent than unconditional generation: SDM scheduling substantially improves Euler baselines, whereas gains under Heun or adaptive-solver settings are less consistent.

F.2 Quantitative Results for modern ODE solvers

Table 6: Quantitative results on modern ODE samplers, including DPM-Solver++ and UniPC, for CIFAR-10 32×32, FFHQ 64×64, and AFHQv2 64×64. Overall, SDM schedules achieves superior performance in FID compared to EDM baselines.

		Unconditional CIFAR-10 32×32		Unconditional FFHQ 64×64		Unconditional AFHQv2 64×64	
		VP	VE	VP	VE	VP	VE
Solver	**Schedule**						
DPM-Solver++(2M)	EDM ($\rho = 7$)	2.57	2.52	2.60	**2.71**	2.12	2.26
	SDM (Adaptive Scheduling)	**2.33**	**2.33**	**2.59**	2.73	**2.07**	**2.19**
UniPC	EDM ($\rho = 7$)	2.32	2.28	2.51	2.62	2.07	2.20
	SDM (Adaptive Scheduling)	**2.26**	**2.13**	**2.49**	**2.60**	**2.04**	**2.14**

F.3 Quantitative Results for Few-Step Generation

Table 7: Quantitative results on 10-step generation for FFHQ 64×64 and AFHQv2 64×64. Results for the adaptive solver are reported with the optimal threshold parameter τ_κ. Adaptive scheduling is applied to Euler, Heun, and SDM-based adaptive solvers. The best-performing configurations compared to the corresponding baselines are highlighted in **bold**, while the overall best results across all methods are highlighted in **<u>bold</u>**. Overall, the proposed SDM sampler consistently shows superior performance in both sample quality and computational efficiency for 10-step generation.

		Unconditional FFHQ 64×64		Unconditional AFHQv2 64×64	
		VP	VE	VP	VE
Solver	**Schedule**				
Euler	EDM ($\rho = 7$)	20.92	21.06	10.14	10.73
	SDM (Adaptive Scheduling)	**17.94**	**19.03**	**9.73**	**10.07**
Heun	EDM ($\rho = 7$)	5.27	5.44	2.99	3.16
	SDM (Adaptive Scheduling)	**<u>3.78</u>**	**<u>3.21</u>**	5.06	3.42
SDM	EDM ($\rho = 7$)	5.34	5.52	**<u>2.92</u>**	**3.02**
	SDM (Adaptive Scheduling)	**3.96**	**3.42**	3.28	**<u>2.50</u>**

Table 8: Quantitative results on 20-step generation for FFHQ 64×64 and AFHQv2 64×64. Results for the adaptive solver are reported with the optimal threshold parameter τ_κ. Adaptive scheduling is applied to Euler, Heun, and SDM-based adaptive solvers. The best-performing configurations compared to the corresponding baselines are highlighted in **bold**, while the overall best results across all methods are highlighted in **bold**. Overall, the proposed SDM sampler consistently shows superior performance in both sample quality and computational efficiency for 20-step generation.

		Unconditional FFHQ 64×64		Unconditional AFHQv2 64×64	
		VP	VE	VP	VE
Solver	**Schedule**				
Euler	EDM ($\rho = 7$)	8.64	8.75	4.61	5.04
	SDM (Adaptive Scheduling)	**7.37**	**8.13**	**4.21**	**4.73**
Heun	EDM ($\rho = 7$)	2.64	2.76	2.08	2.22
	SDM (Adaptive Scheduling)	**2.38**	**2.54**	2.31	2.23
SDM	EDM ($\rho = 7$)	2.65	**2.75**	**1.99**	**2.11**
	SDM (Adaptive Scheduling)	**2.43**	**2.53**	**2.01**	**2.09**

F.4 Quantitative Results for High Resolution Generation

Table 9: Quantitative results on high-resolution generation for ImageNet 512×512 using the EDM2-M pretrained model. (**Left**) FID comparison under EDM and SDM schedules with fixed solvers, DPM-Solver++(2M) and UniPC, respectively. (**Right**) FID comparison with changes in adaptive solver (DPM-Solver++(1M) $\rightarrow$ DPM-Solver++(2M)) and adaptive scheduling.

Solver	**Schedule**	**FID**
DPM-Solver++(2M)	EDM	2.48
DPM-Solver++(2M)	SDM	**2.36**
UniPC	EDM	2.45
UniPC	SDM	**2.37**

Solver	**Schedule**	**FID**
DPM-Solver++(2M)	EDM	2.48
SDM	EDM	**2.47**
DPM-Solver++(2M)	SDM	2.36
SDM	SDM	**2.34**

F.5 Quantitative Results for Text-to-Image Generation

Table 10: Quantitative results on text-to-image generation with SDXL. Comparison of SDM against standard schedules, including time-uniform, EDM, AYS, OTS (Xue et al., 2024), and AutoDiffusion. SDM demonstrates competitive performance against recent schedule optimization baselines. **Bold** and underlined values indicate the best and second-best results, respectively.

Schedule	**FID**	**CLIP**
Uniform	18.29	0.3209
EDM	18.10	**0.3238**
AYS	17.90	0.3205
OTS (Xue et al., 2024)	**17.49**	0.3219
AutoDiffusion	17.76	0.3182
SDM	17.72	0.3201

F.6 Ablation Studies

We further present ablation results on the following key components of the proposed SDM sampler for both unconditional and conditional generation settings in Section 4.

Threshold τ_κ. We investigate the effect of the threshold parameter τ_κ, which controls the activation of higher order solvers. We evaluate FID while varying τ_κ for each dataset, as shown in Figure 3. Based on the results, we set $\tau_\kappa = 2 \times 10^{-4}$ for CIFAR-10, $\tau_\kappa = 1 \times 10^{-4}$ for FFHQ and ImageNet, and $\tau_\kappa = 1 \times 10^{-3}$ for AFHQv2 under EDM schedules. For SDM schedules, we further adjust the threshold on AFHQv2, using $\tau_\kappa = 2 \times 10^{-4}$ for VP parameterization. Note that as long as $\tau_\kappa > 0$, the number of function evaluation is less than 2 per diffusion timestep, *i.e.*, NFE < 2.

Error tolerance and N-step resampling parameters. For adaptive scheduling, we perform a grid search over the error tolerance and resampling parameters on CIFAR-10, as detailed in Table 4. For the remaining datasets, we observe that performance is largely insensitive to the parameters. Consequently, we set parameters $\eta_{\min} = 0.02, \eta_{\max} = 0.20$, $p = 1.0$, and $q = 0.25$ for FFHQ and AFHQv2, and $\eta_{\min} = 0.001, \eta_{\max} = 0.01$, $p = 1.0$, and $q = 0.25$ for ImageNet.

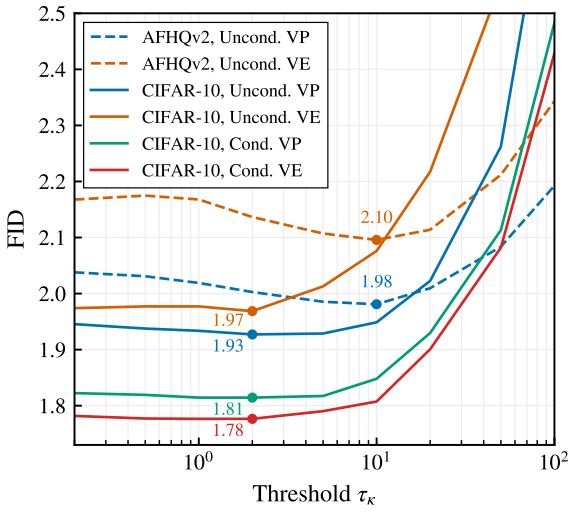


Figure 3: Ablation on threshold τ_κ. FID as a function of the curvature threshold τ_κ for CIFAR-10 (32×32, solid) and AFHQv2 (64×64, dashed), evaluated under unconditional and conditional settings using the curvature-thresholded adaptive solver. Markers denote the selected τ_κ values that yield the best model performance for each dataset and training configuration.

F.7 Qualitative Results

In this section, we present qualitative comparisons between the proposed SDM sampler and the baseline EDM sampler (Karras et al., 2022) across different datasets and training parameterizations.

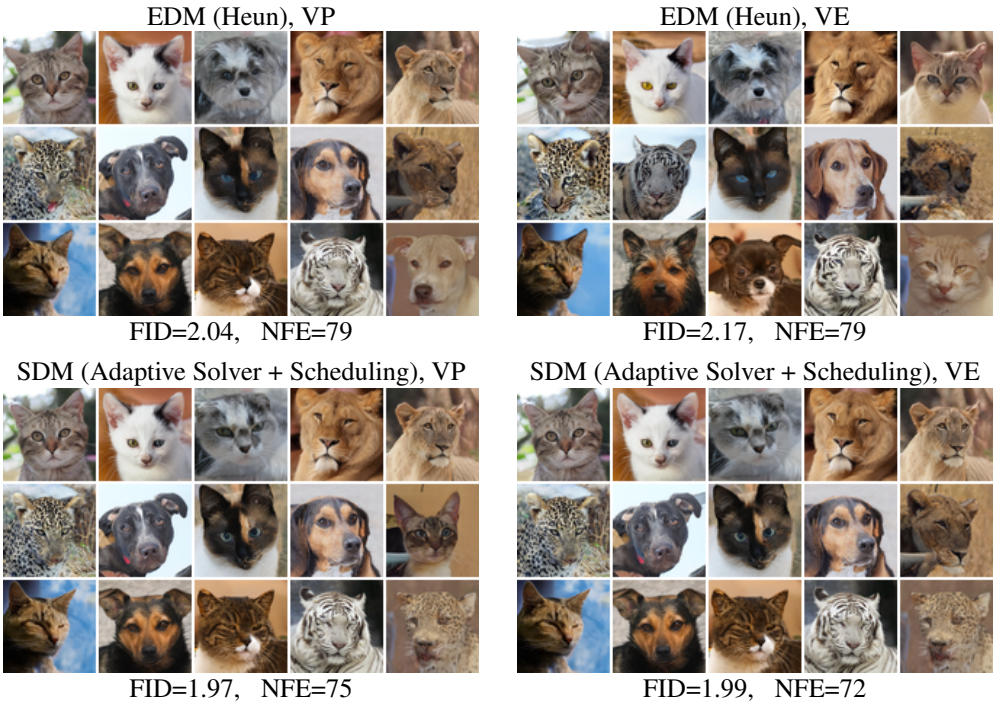


Figure 4: Qualitative comparison on AFHQv2 (64×64) across VP (left) and VE (right) parameterizations. The top row shows EDM (Heun), while the bottom row shows SDM with adaptive solver and scheduling. Each panel shows a 3×5 grid of generated samples; corresponding FID and NFE are reported below.

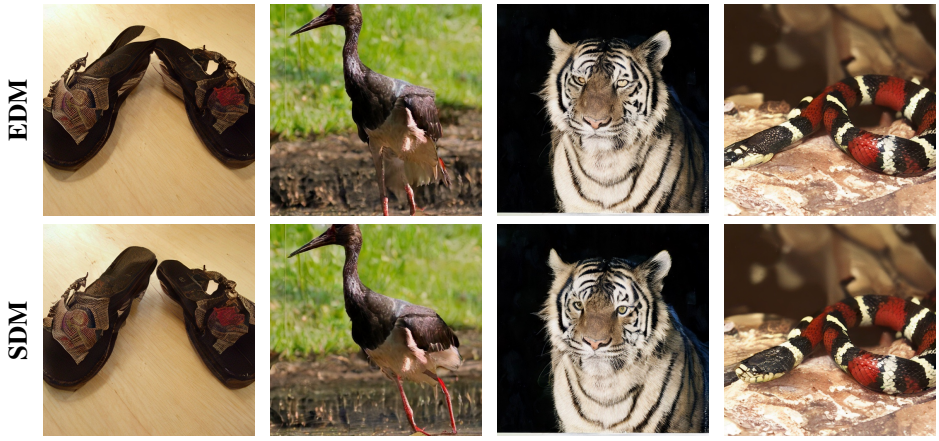


Figure 5: Qualitative comparison on ImageNet 512×512 using the pretrained EDM2-M model. For each pair of rows, the top and bottom row represents results from EDM and SDM schedules, respectively. Across the examples, the EDM schedule exhibits more noticeable structural inconsistencies in generated objects, whereas SDM produces more coherent object structures. All results are generated using DPM-Solver++(2M) as the numerical solver.



Figure 6: Qualitative comparison of different sampling schedules using SDXL for text-to-image generation. From left to right: Time uniform, EDM, AYS, and SDM. All results are generated using DPM-Solver++(2M) as the numerical solver.

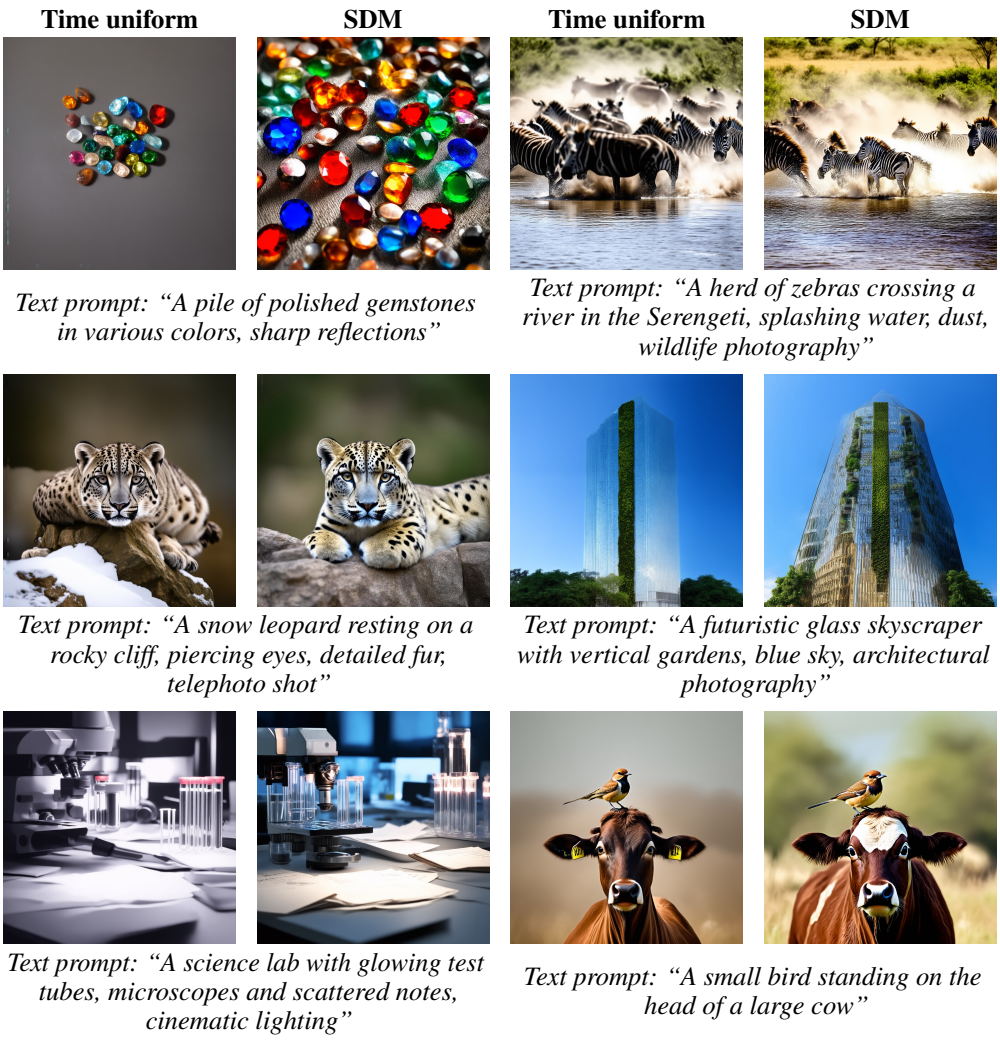


Figure 7: Qualitative comparison of different sampling schedules using Stable Diffusion 3 for text-to-image generation. From left to right: Time uniform and SDM. All results are generated using Euler as the numerical solver.