PROJECTION BOTTLENECK IN ADAPTIVE DIFFUSION SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive timestep selection is widely used to accelerate sampling in continuous-time generative models, yet its effectiveness remains poorly understood. Existing methods generally posit that structured trajectory signals can be leveraged to derive improved discretization schemes. In this work, we show that this assumption can break down for projection-based step-wise adaptation under fixed sampling budgets. We find that trajectory-level signals, including hidden speed, curvature, and topological summaries, exhibit strong and consistent temporal structure across models and datasets. However, directly using these signals for step-wise adaptive control does not consistently improve sampling quality and can lead to severe degradation. Empirically, diverse controllers often produce identical timestep schedules and even identical generated samples. We identify an important mechanism underlying this behavior as a *projection bottleneck*: under discretization constraints, distinct adaptive proposals can be mapped onto the same feasible schedule through a many-to-one projection. We characterize this effect theoretically through active-set non-injectivity and local loss of controller degrees of freedom under saturation. Our results suggest that effective adaptive sampling depends not only on extracting trajectory information, but also on preserving that information during schedule construction, motivating alternative allocation strategies that avoid projection-induced information loss.

1 INTRODUCTION

Continuous-time generative modeling has emerged as a unified framework for modern generative models, including diffusion models, probability-flow ODEs, rectified flows, and flow matching Ho et al. (2020); Song et al. (2020b;a); Lipman et al. (2022); Liu et al. (2022); Ahamed et al. (2026). In these approaches, the generative process is formulated as the transport of samples from a simple prior distribution to the data distribution via a learned time-varying vector field. Although these models demonstrate powerful generative capabilities, their computational cost during the inference phase remains substantial, often requiring hundreds of neural network function evaluations (NFEs) to generate high-quality samples.

To enhance sampling efficiency, extensive research has focused on exploring adaptive time-step discretization techniques, including high-order solvers, learning-based samplers, and acceleration methods based on knowledge distillation Lu et al. (2022); Karras et al. (2022); Salimans & Ho (2022); Meng et al. (2023). These methods seek to allocate computational resources using trajectory-dependent signals-such as velocity magnitude, truncation error, or score norms-with the general goal of translating variations in trajectory dynamics into more effective time-step allocation schemes. Recent studies, conducted from theoretical, geometric, and trajectory-centric perspectives, have further demonstrated that diffusion dynamics are not merely confined to local denoising behaviors but also encapsulate rich global structures (Alonso et al., 2026; Chen et al., 2026; De Bortoli et al., 2022; Huang et al., 2022; Karczewski et al., 2025; Edelsbrunner et al., 2002; Carlsson, 2009). Nevertheless, it remains unclear how trajectory-level structure interacts with discretization constraints under a fixed computational budget, and whether distinctions captured by adaptive controllers survive the construction of executable timestep schedules.

In this work, we show that this assumption can break down for projection-based step-wise adaption under fixed sampling budgets, where the total number of NFE is prescribed in advance and

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.

timestep schedules must satisfy the same computational constraints. Specifically, we begin by analyzing trajectory-level dynamics using both geometric signals and topological summaries based on zigzag persistent homology. Across datasets and model families, these signals exhibit strong and consistent temporal structure, revealing distinct phases of the generative process. However, despite their pronounced and reproducible temporal structure, directly using these signals for step-wise timestep controls does not consistently improve sampling quality. Under strict matched-NFE settings, step-wise trajectory-aware controllers do not outperform simple handcrafted schedules and can substantially degrade performance. In highly saturated regimes, different controllers produce identical timestep schedules and even identical generated samples.

We identify a key mechanism underlying this behavior, which we term the **projection bottleneck**. In projection-based step-wise adaption, raw timestep proposals must be mapped to a feasible discrete schedule satisfying budget and step-size constraints. We show both empirically and theoretically that this projection can be many-to-one under active constraints, while increasing saturation reduces the local degrees of freedom through which controller variations can influence the executable schedule. As a result, distinctions present in raw controller proposals can be substantially suppressed during schedule construction. These findings highlight an important distinction in adaptive sampling: extracting trajectory-dependent information and preserving its influence during schedule construction are separate challenges.

To address this limitation, we reformulate adaptive sampling as a *global allocation* problem. Instead of adjusting timesteps sequentially, we allocate computation jointly across the entire trajectory, avoiding the repeated local projection that gives rise to the observed step-wise collapse. This formulation better preserves distinctions induced by trajectory signals and provides an alternative to repeated step-wise projection. We summarize our contributions as follows:

- **Trajectory-Level Analysis.** We show that generative trajectories exhibit consistent geometric and topological structure across models and datasets.
- **Failure of Step-wise Adaptive Control.** We demonstrate that under matched fixed-NFE budgets, projection-based trajectory -aware controllers can substantially degrade sampling quality and, in saturated regimes, collapse to identical schedules and outputs.
- **Projection Bottleneck.** We identify active-set saturation as a mechanism through which feasibility projection can suppress distinction between raw controller proposals and executable schedules.
- **Theoretical Characterization.** We characterize the bottleneck through active-set non-injectivity and local loss of controller degrees of freedom under feasibility projection.
- **Global Allocation Perspective.** We study joint budget allocation as an alternative to repeated step-wise projection, reducing schedule collapse while recovering competitive sampling quality.

2 BACKGROUND

Continuous-time generative models, including diffusion models and flow-based methods, generate samples by integrating a learned time-dependent vector field that maps a simple prior distribution to the data distribution. In practice, the sampling process typically involves discretizing time into a sequence of steps $\{t_i\}_{i=0}^{N}$ employing a numerical solver to generate a trajectory $\{x_{t_i}\}$.

Adaptive sampling aims to non-uniformly allocate computational resources across the time dimension by adjusting the step sizes $\Delta t_i = t_{i+1} - t_i$ based on trajectory-dependent sensitivity signals. A common implementation involves first computing a raw step-size proposal Δt_i^{raw} based on local metrics, followed by a projection step to ensure that feasibility constraints are satisfied:

$$\Delta t^{\text{final}} = \mathcal{P}(\Delta t^{\text{raw}}), \tag{1}$$

where $\mathcal{P}$ ensures $\sum_i \Delta t_i = 1$ and $\Delta t_i \in [\Delta t_{\min}, \Delta t_{\max}]$.

Under strict matched-NFE settings, the total number of steps N is fixed, making timestep allocation a constrained discretization problem. This formulation underlies most existing adaptive sampling strategies and serves as the basis for our analysis.

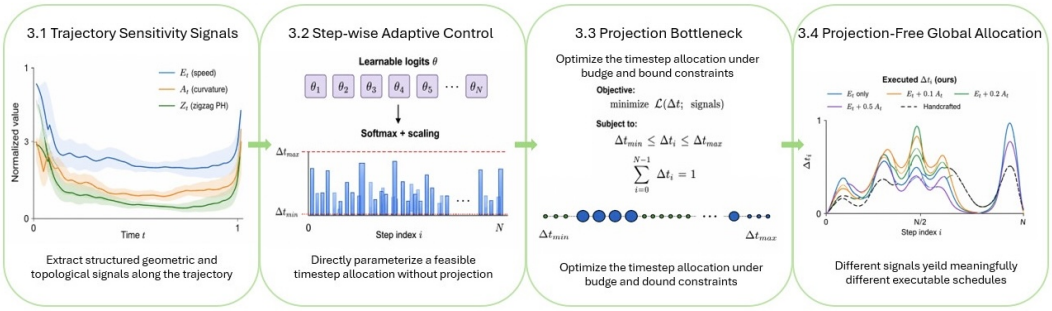


Figure 1: **Method overview.** We extract trajectory sensitivity signals 3.1, including geometric and topological descriptors that capture structured variation along the generative path. Step-wise adaptive controllers use these signals to construct local timestep proposals 3.2, whose subsequent feasibility projection can suppress controller distinctions under active constraints 3.3. Motivated by this projection bottleneck, we consider global schedule allocation, which distributes the fixed sampling budget jointly across the trajectory without repeated step-wise projection 3.4.

3 METHOD

Figure 1 summarizes our framework. We extract trajectory sensitivity signals and examine how step-wise feasibility projection affects their translation into executable schedules. Motivated by the resulting projection bottleneck, we consider a global allocation formulation that jointly distributes the fixed sampling budget without repeated step-wise projection.

We consider adaptive timestep scheduling for continuous-time generative models under a fixed function-evaluation budget. Let $x(t) \in \mathbb{R}^d$ denote a sampling trajectory governed by an ODE

$$\frac{dx(t)}{dt} = v_\theta(x(t), t), \qquad t \in [0, 1], \tag{2}$$

where v_θ denotes a diffusion probability-flow or rectified-flow vector field. In practice, the trajectory is approximated using a discrete grid

$$0 = t_0 < t_1 < \cdots < t_N = 1, \qquad \Delta t_i = t_{i+1} - t_i, \tag{3}$$

where N is the prescribed NFE.

The objective of adaptive scheduling is to allocate $\{\Delta t_i\}_{i=0}^{N-1}$ based on trajectory-dependent sensitivity signals under a fixed computational budget. This implicitly assumes that informative signals can be translated into effective timestep allocation. However, as we will show, discretization constraints can substantially weaken this correspondence: distinct raw proposals may collapse to nearly identical executable schedules after feasibility projection.

3.1 MULTI-SCALE TRAJECTORY SIGNALS

We construct multi-scale trajectory sensitivity signals that capture the evolution of generative dynamics across multiple scales, providing a unified characterization of *trajectory sensitivity*. Our goal is to quantify where finer discretization is required along the sampling path.

Representation trajectory. Let

$$h_i = \phi_\theta(x(t_i), t_i) \in \mathbb{R}^m \tag{4}$$

denote a hidden representation at timestep t_i. This induces a discrete trajectory

$$\mathcal{H} = \{h_0, h_1, \ldots, h_N\}, \tag{5}$$

which serves as a proxy for the underlying continuous-time dynamics.

Local geometric sensitivity. We first quantify local variations using first- and second-order finite differences:

$$E_i = \frac{\|h_i - h_{i-1}\|_2}{(\Delta t_{i-1} + \epsilon)(\|h_{i-1}\|_2 + \epsilon)}, \quad A_i = \frac{\|h_i - 2h_{i-1} + h_{i-2}\|_2}{(\Delta t_{i-1}^2 + \epsilon)(\|h_{i-1}\|_2 + \epsilon)}, \tag{6}$$

where $\epsilon > 0$ is a small numerical constant used for stability. The quantity E_i measures local trajectory variation, while A_i captures higher-order curvature and acceleration effects along the hidden-state trajectory.

To improve robustness against local fluctuations, we apply exponential moving average (EMA) smoothing:

$$\bar{E}_i = \beta_E \bar{E}_{i-1} + (1 - \beta_E) E_i, \qquad \bar{A}_i = \beta_A \bar{A}_{i-1} + (1 - \beta_A) A_i, \tag{7}$$

where $\beta_E, \beta_A \in [0, 1)$ control the smoothing strength. This yields a multi-scale measure of trajectory sensitivity, with larger values indicating regions of stronger observed trajectory variation and therefore serving as candidate signals for adaptive refinement.

Global structural evolution. Local geometric quantities do not capture higher-order structural changes along the trajectory. To incorporate global information, we define a topological signal over sliding windows

$$W_i = \{h_{i-k}, \ldots, h_i\}. \tag{8}$$

We construct a sequence of simplicial complexes over $\{W_i\}$ and apply zigzag persistent homology Chen & Gel (2025) to obtain persistence diagrams $\mathcal{D}_q(W_i)$. We summarize structural evolution via

$$Z_i = \Psi\big(\mathcal{D}_0(W_i), \mathcal{D}_1(W_i)\big), \tag{9}$$

where Ψ denotes a stable functional of total persistence. The resulting signal $\{Z_i\}$ captures global topological transitions over time.

Unified multi-scale score. We combine local and global information into a single trajectory score

$$C_i = \bar{E}_i + \lambda_A \bar{A}_i + \lambda_Z Z_i, \tag{10}$$

where $\lambda_A, \lambda_Z \geq 0$ balance geometric and topological contributions. This yields a multi-scale measure of trajectory sensitivity, with larger values indicating regions that, in principle, require finer discretization treatment.

While these signals exhibit structured variation along the generative trajectory, we show in Sections 3.3 and 5 that distinctions induced by these signals can be substantially suppressed during the construction of executable timestep schedules.

3.2 Step-wise Adaptive Control

Recent work has explored adaptive routing and allocation mechanisms for improving diffusion efficiency (Yin et al., 2025). A common adaptive scheduling strategy constructs timestep allocations sequentially using trajectory-dependent sensitivity signals.

Given a trajectory sensitivity score C_i, the local timestep proposal is typically defined as

$$\Delta t_i^{\text{raw}} = \frac{\Delta t_i^{\text{base}}}{1 + \alpha C_i}, \tag{11}$$

where Δt_i^{base} denotes a reference schedule (uniform or phase-based), and $\alpha > 0$ controls adaptation strength.

This formulation follows the standard intuition that regions with higher trajectory sensitivity should receive finer temporal discretization.

Feasibility constraints. In practice, timestep allocations must satisfy strict discretization constraints:

$$\Delta t_i^{\text{final}} > 0, \qquad \sum_{i=0}^{N-1} \Delta t_i^{\text{final}} = 1, \tag{12}$$

with a fixed number of steps N and bounds $\Delta t_i \in [\Delta t_{\min}, \Delta t_{\max}]$. As a result, the raw proposals $\{\Delta t_i^{\text{raw}}\}$ are not directly executable.

Projection to feasible schedules. To enforce these constraints, practical implementations apply a projection bottleneck:

$$\Delta t^{\text{final}} = \mathcal{P}\left(\Delta t^{\text{raw}}; \Delta t_{\min}, \Delta t_{\max}, N\right), \tag{13}$$

which consists of clipping, normalization, and budget correction steps.

3.3 Projection Bottleneck

A key limitation of projection-based adaptive timestep scheduling can arise from the feasibility operation used to map raw proposals to executable schedules. Given raw timestep proposals Δt^{raw}, practical schedulers must construct a feasible schedule satisfying positivity, normalization, and bound constraints. This is achieved through a projection

$$\Delta t^{\text{final}} = \mathcal{P}(\Delta t^{\text{raw}}). \tag{14}$$

Although Δt^{raw} may encode rich variations induced by trajectory sensitivity signals, these distinctions need not be preserved after projection. In particular, when many coordinates approach the lower bound $\Delta t_{\min}$, active constraints can make the projection insensitive to variations along certain proposal directions, mapping distinct raw proposals to nearly identical executable schedules.

Formally, for two distinct proposals

$$\Delta t^{\text{raw}} \neq \tilde{\Delta} t^{\text{raw}}, \tag{15}$$

it is possible that

$$\mathcal{P}(\Delta t^{\text{raw}}) \approx \mathcal{P}(\tilde{\Delta} t^{\text{raw}}), \tag{16}$$

indicating that the mapping is effectively many-to-one. Consequently, distinct adaptive controllers may produce indistinguishable executed timestep sequences.

We term this phenomenon the *projection bottleneck*. The effect becomes particularly severe in the saturated regime, where a large fraction of timesteps are clamped at $\Delta t_{\min}$. In this regime, as the number of active constraints increases, fewer local degrees of freedom remain through which variations in raw proposals can influence the executable schedule.

As we demonstrate empirically in Section 5 and characterize theoretically in Section 4, this bottleneck provides a mechanism for the observed degradation of projection-based step-wise adaptive control under fixed discretization budgets.

Additional diagnostics and quantitative analyses are provided in Appendix B.

3.4 Global Schedule Allocation

The analysis above suggests that degradation of step-wise adaptive control can arise not only from the choice of trajectory signal, but also from how raw controller proposals are converted into executable schedules. In particular, repeated feasibility projection can suppress distinctions between proposals when many constraints become active. Motivated by this observation, we consider a global allocation formulation that constructs the timestep schedule jointly rather than through sequential local updates.

Global allocation formulation. Instead of constructing timestep updates sequentially and applying post-hoc projection, we allocate the sampling budget jointly across the trajectory. Given trajectory-dependent scores $\{C_i\}_{i=0}^{N-1}$, we define non-negative weights

$$w_i = b_i \exp(-\alpha \widetilde{C}_i), \tag{17}$$

where $\widetilde{C}_i$ is a normalized score, $\alpha > 0$ controls sensitivity, and $b_i > 0$ is a base schedule encoding prior allocation (e.g., uniform or phase-based).

The timestep allocation is then obtained by normalization:

$$\Delta t_i = \frac{w_i}{\sum_{j=0}^{N-1} w_j}. \tag{18}$$

This normalization directly enforces positivity and the fixed total-budget constraint without requiring repeated step-wise projection.

Preserving trajectory-aware structure. Because the schedule is constructed jointly from the global weight vector, differences in trajectory-dependent scores directly influence the relative timestep allocation. Unlike step-wise adaptive control, the formulation avoids repeated local clipping and budget correction, thereby reducing the opportunity for active constraints to suppress distinctions between controller proposals.

From local control to global allocation. The fundamental distinction between step-wise adaptive control and the proposed global allocation formulation lies in how the sampling budget is enforced. Step-wise methods construct local proposals and subsequently enforce feasibility, which can introduce active-set saturation and cause distinct proposals to collapse to similar schedules. In contrast, global allocation distributes the total sampling budget jointly across all timesteps through the normalized weight vector in Equations 17 and 18.

This perspective shifts adaptive scheduling from sequential local control followed by feasibility correction to joint allocation of the fixed sampling budget.

Implications. The global formulation provides a complementary test of the projection bottleneck hypothesis. If the degradation of step-wise controllers is associated with projection-induced collapse, then avoiding repeated local projection should reduce schedule collapse and recover sampling performance. We evaluate this hypothesis empirically in Section 5.

4 Theoretical Characterization of the Projection Bottleneck

We formalize how feasibility projection can suppress distinctions between adaptive timestep proposals. Rather than requiring projection to be globally contractive, our analysis identifies two properties that are sufficient to explain the observed schedule collapse: non-injectivity and local loss of controller degrees of freedom under active constraints.

Setup. Let $\mathbf{u} \in \mathbb{R}^N$ denote raw timestep proposals. The feasible set is

$$\mathcal{S} = \left\{ \Delta\mathbf{t} \in \mathbb{R}^N \;\middle|\; \Delta t_i \in [\Delta t_{\min}, \Delta t_{\max}],\ \sum_{i=1}^{N} \Delta t_i = 1 \right\}. \tag{19}$$

The executed schedule is obtained via Euclidean projection,

$$\mathcal{P}(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{x} - \mathbf{u}\|_2^2. \tag{20}$$

Because $\mathcal{S}$ is closed and convex, $\mathcal{P}$ is well-defined and non-expansive. Our interest is therefore not in claiming global strict contraction, but in characterizing when active constraints make the mapping from controller proposals to executable schedules locally non-injective or insensitive to proposal perturbations.

Active-set non-injectivity.

Theorem 1. *Let $\mathbf{x}^\star \in \mathcal{S}$ lie on the boundary of the feasible set, and let $N_{\mathcal{S}}(\mathbf{x}^\star)$ denote the normal cone of $\mathcal{S}$ at $\mathbf{x}^\star$. For any $\mathbf{n}_1, \mathbf{n}_2 \in N_{\mathcal{S}}(\mathbf{x}^\star)$ with $\mathbf{n}_1 \neq \mathbf{n}_2$,*

$$\mathcal{P}(\mathbf{x}^\star + \mathbf{n}_1) = \mathcal{P}(\mathbf{x}^\star + \mathbf{n}_2) = \mathbf{x}^\star. \tag{21}$$

Hence, whenever one or more inequality constraints are active, $\mathcal{P}$ is non-injective in directions normal to the corresponding face of the feasible set.

Proof. For Euclidean projection onto a closed convex set, $\mathbf{x}^\star = \mathcal{P}(\mathbf{u})$ if and only if

$$\mathbf{u} - \mathbf{x}^\star \in N_{\mathcal{S}}(\mathbf{x}^\star). \tag{22}$$

Therefore, for any $\mathbf{n} \in N_{\mathcal{S}}(\mathbf{x}^\star)$, the point $\mathbf{x}^\star + \mathbf{n}$ projects to $\mathbf{x}^\star$. Choosing two distinct normal vectors gives two distinct raw proposals with the same executable schedule.

□

Local loss of controller degrees of freedom. Consider a neighborhood in which the active set is fixed. Let $I_{\min}$ and $I_{\max}$ denote the coordinates clamped at $\Delta t_{\min}$ and $\Delta t_{\max}$, respectively, and let

$$F = \{1, \dots, N\} \setminus (I_{\min} \cup I_{\max})$$

denote the free coordinates.

Theorem 2. *Within a region of proposal space where the active set remains unchanged, the projection map is affine. Perturbations to active coordinates that lie in locally inactive controller directions do not affect the executable schedule, and the Jacobian of the projection has rank at most*

$$\operatorname{rank}(J_{\mathcal{P}}) \le \max(|F| - 1, 0). \tag{23}$$

Consequently, increasing the number of active constraints reduces the local number of proposal directions that can influence the executed schedule.

Proof. From a fixed active set, coordinates in $I_{\min} \cup I_{\max}$ are fixed at their corresponding bounds. The remaining coordinates are obtained by projecting the free proposal coordinates onto the affine constraint induced by $\sum_i \Delta t_i = 1$. When $|F| \ge 1$, the local Jacobian restricted to the free coordinates takes the form

$$J_F = I_{|F|} - \frac{1}{|F|}\mathbf{1}\mathbf{1}^\top, \tag{24}$$

while derivatives associated with active coordinates vanish as long as the active set does not change. Since

$$\operatorname{rank}(J_F) = |F| - 1, \tag{25}$$

the projection preserves at most $|F|-1$ local degrees of freedom. Thus, as more coordinates become saturated, fewer independent controller perturbations can alter the executable schedule. □

Implication for adaptive scheduling. These results characterize the projection bottleneck without requiring a global strict-contraction assumption. Projection onto the bounded simplex is nonexpansive, but active constraints create directions along which distinct controller proposals become indistinguishable. As saturation increases, the dimension of the locally preserved proposal space decreases, so increasingly different raw controller outputs can induce the same, or nearly the same, executable schedule. The results show the mechanism studied empirically in Section 5:

$$\text{trajectory signal} \longrightarrow \text{distinct raw proposals} \xrightarrow{\mathcal{P}} \textbf{collapsed executable schedules}. \tag{26}$$

The resulting loss of controller distinguishability does not imply that all adaptive sampling fails, nor that projection is globally contractive. Rather, it identifies a specific failure mode of bounded, projection-based step-wise adaptation: once many constraints become active, variations in the controller can lie in directions to which the executable schedule is locally insensitive.

5 Experimental Results: Projection Bottleneck Characterization

We empirically examine how structured trajectory signals are translated into executable timestep schedules under fixed discretization budgets. All experiments are conducted under strict matched-NFE settings across multiple datasets and generative model families. Detailed implementation settings, training configurations, evaluation protocols, and experiment scripts are provided in Appendix C

Trajectory sensitivity is structured. We first examine trajectory sensitivity along the sampling process. Across datasets (CIFAR-10, MNIST, AFHQ-Cat, Tiny ImageNet) and model families (DDPM, DDIM, Rectified Flow), both geometric quantities (E_t, A_t) and topological summaries (Z_t) exhibit consistent temporal structure. These signals show smooth early-stage dynamics followed by increasingly complex variations, indicating reproducible temporal variation rather than unstructured fluctuations. Figure 2 shows the trajectory signals of CIFAR-10 with Rectified Flow; additional trajectory patterns are provided in Appendix D.1.

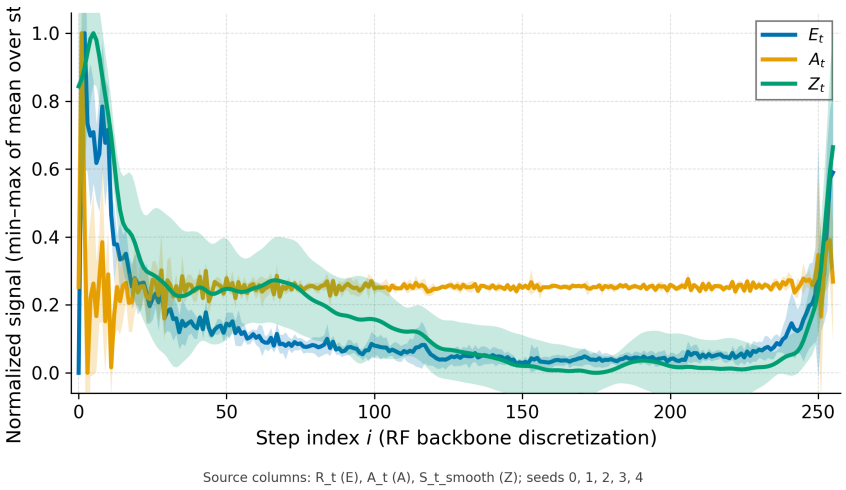


Figure 2: **Hidden speed (E_t), curvature (A_t), and topology (Z_t) exhibit strong and consistent temporal structure along the sampling trajectory.**

Table 1: FID (↓) on CIFAR-10 with Rectified Flow under matched NFE. Mean ± std over 5 seeds.

Method	**30**	**50**	**100**
Handcrafted	**34.19**±.17	**32.92**±.42	**31.93**±.28
Uniform	34.72±.21	33.26±.46	32.16±.29
Phase-Topo.	49.04±9.22	35.07±1.59	32.00±.25
Exact Zigzag	37.03±1.19	34.68±.99	32.45±.32
Dynamic Zigzag	37.67±.34	37.52±.50	–

Table 2: FID (↓) on CIFAR-10 under matched NFE. Mean ± std over 5 seeds.

Method	**30**	**50**
Handcrafted	**34.10**±.37	**32.92**±.35
Uniform	34.72±.21	33.26±.46
Step-wise E_t	82.74±.72	76.03±.64
Step-wise E_t+A_t	125.24±1.32	133.64±1.57
Global E_t	34.21±.49	32.99±.43
Global E_t+A_t	34.25±.39	33.00±.44

Step-wise adaptive control fails under fixed NFE. Despite the structured nature of these signals, directly incorporating them into adaptive step-size control does not improve sampling quality. As shown in Table 1, handcrafted phase schedules consistently outperform trajectory-aware adaptive schedules across matched step budgets, especially in low-NFE regimes. Topology-based and zigzag-based variants often degrade performance, suggesting that richer trajectory signals alone are insufficient for effective timestep allocation.

Table 2 further isolates the source of failure. Geometry-based step-wise controllers dramatically degrade performance, increasing FID from 34.10 to over 125 at NFE=30. In contrast, the proposed global allocation formulation restores performance to the level of handcrafted schedules, indicating that schedule construction is an important contributor to the observed degradation. At higher budgets, the gap between methods narrows, suggesting that timestep allocation becomes less critical once the trajectory is sufficiently resolved.

Projection collapses adaptive schedules. To understand this failure, we analyze the executed timestep schedules. Figure 3 shows that distinct controllers produce different raw timestep proposals, yet collapse to nearly identical schedules after projection onto the feasible set. Most timesteps are clamped to $\Delta t_{\min}$, while the remaining budget concentrates in a small number of large steps. This indicates that the projection can substantially suppress distinctions between raw controller proposals and executable schedules.

Projection eliminates controller distinguishability. As a consequence, different controllers become indistinguishable after projection. In multiple settings, variants such as E_t, $E_t + 0.1A_t$, and $E_t + 0.2A_t$ yield identical FID values under fixed NFE. Inspection of the resulting schedules and

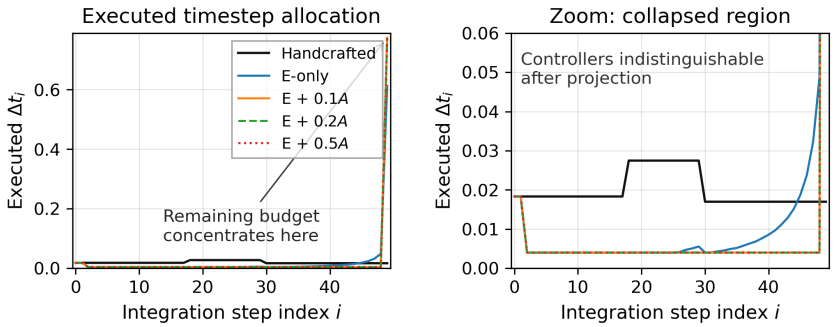


Figure 3: **Projection-induced schedule collapse.** Different controller proposals yield distinct raw timestep allocations, but after enforcing discretization constraints, they collapse to nearly identical executed schedules. Most steps are saturated at the minimum timestep $\Delta t_{\min}$, while the remaining budget concentrates in a small number of large steps.

samples confirms that both timestep sequences and generated outputs coincide across these controllers.

Saturation reduces controller distinguishability. On CIFAR-10 with Rectified Flow, curvature-aware controllers have 94% of timestep intervals clamped at $\Delta t_{\min}$ across all five seeds (see Appendix D.2). Consequently, three distinct settings $\Lambda_A \in \{0.1, 0.2, 0.5\}$ collapse to the same executable schedule and, under deterministic sampling, produce byte-identical sets of 1,000 images. This provides direct evidence that saturation can eliminate controller distinctions, consistent with the active-set non-injectivity and loss of local degrees of freedom characterized in Section 4.

Additional experiments on AFHQ-Cat at 256×256 show that phase-aware trajectory scheduling can also improve over handcrafted and uniform schedules at higher resolution (Appendix D.3).

6 Conclusion

Discussion Although trajectory-level signals exhibit rich and consistent geometric and topological structure across models and datasets, they do not consistently improve sampling under fixed discretization budgets. We identify an important mechanism underlying this behavior as the *projection bottleneck*: discretization constraints can induce a many-to-one mapping that collapses distinct controller proposals into nearly identical executable schedules, thereby suppressing the correspondence between trajectory sensitivity and timestep allocation. A central insight is that extracting trajectory-dependent information and preserving its influence during schedule construction are separate challenges, making schedule construction an important factor in adaptive sampling under fixed discretization budgets. The proposed global allocation formulation addresses this limitation by allocating the sampling budget jointly across the trajectory, thereby avoiding repeated step-wise projection and reducing projection-induced collapse.

Conclusion. Our empirical and theoretical analyses identify projection-induced information loss as an important failure mode of bounded, projection-based step-wise adaptive scheduling under fixed computational budgets. Active-set saturation can suppress distinctions between controller proposals, limiting how trajectory-dependent information influences the executed schedule. Global allocation provides an alternative to repeated local projection and recovers competitive sampling quality, highlighting schedule construction as an important design dimension for adaptive sampling. Future work should examine whether these findings extend to larger generative models, including conditional and more general stochastic or learned schedulers. It would also be valuable to study alternative constraint-handling mechanisms that preserve controller distinctions while satisfying practical sampling requirements. Finally, jointly learning trajectory-aware signals and schedule allocation may provide a path toward adaptive samplers that better align trajectory information with computational allocation.

REFERENCES

Shadab Ahamed, Eshed Gal, Simon Ghyselincks, Md Shahriar Rahim Siddiqui, Moshe Eliasof, and Eldad Haber. Preconditioned score and flow matching. *arXiv preprint arXiv:2603.02337*, 2026.

Noguer I Alonso et al. Three perspectives on stochastic differential equations: Kinetic, numerical, and geometric viewpoints. *Three Perspectives on Stochastic Differential Equations: Kinetic, Numerical, and Geometric Viewpoints (January 01, 2026)*, 2026.

Georgios Arvanitidis, Miguel González-Duque, Alison Pouplin, Dimitris Kalatzis, and Søren Hauberg. Pulling back information geometry. *arXiv preprint arXiv:2106.05367*, 2021.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Gunnar Carlsson and Vin De Silva. Zigzag persistence. *Foundations of computational mathematics*, 10(4):367–405, 2010.

Ping Chen, Xiang Liu, Xingpeng Zhang, Fei Shen, Xun Gong, Zhaoxiang Liu, Zezhou Chen, Huan Hu, Kai Wang, and Shiguo Lian. Chain-of-trajectories: Unlocking the intrinsic generative optimality of diffusion models via graph-theoretic planning. *arXiv preprint arXiv:2603.14704*, 2026.

Yuzhou Chen and Yulia Gel. Topological zigzag spaghetti for diffusion-based generation and prediction on graphs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8185–8194. IEEE, 2020.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

Tamal Krishna Dey and Yusu Wang. *Computational topology for data analysis*. Cambridge University Press, 2022.

Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, 28(4):511–533, 2002.

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

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Rafał Karczewski, Markus Heinonen, Alison Pouplin, Søren Hauberg, and Vikas Garg. Spacetime geometry of denoising in diffusion models. *arXiv preprint arXiv:2505.17517*, 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.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

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.

Alexander Lobashev, Dmitry Guskov, Maria Larchenko, and Mikhail Tamm. Hessian geometry of latent space in generative models. *arXiv preprint arXiv:2506.10632*, 2025.

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.

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.

mnmoustafa and Mohammed Ali. Tiny imagenet. `https://www.kaggle.com/competitions/tiny-imagenet`, 2017. Kaggle.

Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of riemannian geometry. *Advances in Neural Information Processing Systems*, 36:24129–24142, 2023.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11410–11420, 2022.

Michele Pavon, Giulio Trigila, and Esteban G Tabak. The data-driven schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74(7):1545–1573, 2021.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

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*, 2020b.

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022.

Bo Yin, Xiaobin Hu, Xingyu Zhou, Peng-Tao Jiang, Yue Liao, Junwei Zhu, Jiangning Zhang, Ying Tai, Chengjie Wang, and Shuicheng Yan. Fera: Frequency-energy constrained routing for effective diffusion adaptation fine-tuning. *arXiv preprint arXiv:2511.17979*, 2025.

Zhenyu Zhou, Defang Chen, Can Wang, Chun Chen, and Siwei Lyu. Simple and fast distillation of diffusion models. *Advances in Neural Information Processing Systems*, 37:40831–40860, 2024.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this paper. Specifically, they were used to improve language and clarity, condense selected paragraphs, and summarize experimental settings based on information provided by the authors. AI tools were also used to improve the presentation and wording of theoretical proofs, without generating the underlying mathematical arguments or claims. In addition, AI tools assisted with understanding and organizing existing code.

Generative AI tools were not used for the ideation of research, the execution of experiments, the generation or modification of datasets or experimental results or the generation of mathematical claims. All experimental results, tables, and figures were produced and verified by the authors. The authors reviewed and edited all AI-assisted content and assume full responsibility for the final manuscript and its scientific claims.

A Related Work

Diffusion and Continuous-Time Generative Models. Diffusion models and their continuous-time formulations have emerged as the dominant paradigm in the field of generative modeling, encompassing score-based models and probability flow ordinary differential equations (ODEs) Song et al. (2020b), and rectified flows Liu et al. (2022); Lipman et al. (2022). These methods can be viewed under a unified framework of continuous-time latent variable models, where sampling corresponds to solving a stochastic or deterministic differential equation that transports a simple prior distribution to the data distribution. Extensions such as Schrödinger bridges De Bortoli et al. (2021); Pavon et al. (2021) and Poisson flow generative models Xu et al. (2022) further generalize this perspective by modifying the underlying coupling mechanisms or stochastic processes. Although the aforementioned studies primarily focus on improving training objectives or enhancing sampling efficiency, they typically continue to rely on fixed or heuristically designed time discretization schemes.

Adaptive Sampling and Time Discretization. An increasing number of studies are dedicated to exploring how to reduce the number of function evaluations (NFE) required for diffusion sampling. Existing approaches include higher-order solvers Lu et al. (2022); Karras et al. (2022), learned samplers Salimans & Ho (2022); Meng et al. (2023), and heuristic step-size adaptation strategies based on local error estimates or score magnitudes. More recently, research has also explored the use of distilled diffusion samplers to lower inference costs Zhou et al. (2024). Despite their empirical success, these methods rely heavily on local numerical signals (e.g., truncation errors, score norms) and lack a principled, global characterization of the structure of the sampling trajectory. Specifically, they do not explicitly model the evolution of information along the entire generation path.

Geometric Perspectives on Diffusion Models. Recent studies have proposed various geometric interpretations of diffusion processes. The framework of Riemannian geometry explores diffusion phenomena on curved manifolds De Bortoli et al. (2022); Huang et al. (2022), concurrently, researchers have utilized pullback metrics and latent space geometry to analyze the geometric structures induced by generative mappings Arvanitidis et al. (2021); Park et al. (2023). Within the field of generative modeling, higher-order geometric structures—including the Hessian geometry of latent manifolds—have also recently become subjects of exploration Lobashev et al. (2025).

However, researchers have observed that the pullback geometry in diffusion models sometimes exhibits degeneracy: geodesics in the latent space, when decoded into the data space, often manifest as nearly straight trajectories, thereby limiting the models' capacity to capture intrinsic semantic structures.

Recently, researchers introduced an information-geometric perspective to address this issue by defining the Fisher-Rao metric on the denoising distribution Karczewski et al. (2025). By lifting latent representations onto a spacetime manifold $z = (x_t, t)$, these methods reveal non-trivial geometric features of trajectories and enable the analysis of geodesics and trajectory energies within the diffusion process. Although these perspectives provide a geometrically grounded characterization of generative dynamics, they largely remain at a descriptive level and have not yet directly addressed practical challenges such as adaptive discretization or time-step allocation.

Topology and Structure in Generative Models. Topological data analysis (TDA), including persistent homology Edelsbrunner et al. (2002); Carlsson (2009); Dey & Wang (2022), has been applied to understand the structure of learned representations and data manifolds. Beyond standard persistent homology, zigzag persistence enables the tracking of topological features across non-monotonic filtration sequences and evolving structures Carlsson & De Silva (2010). Recent research has explored the application of topology within the context of deep learning, spanning domains such as representation learning, robustness, and uncertainty quantification. In the context of generative models, topology has been utilized to characterize latent spaces or intermediate features; however, such applications are typically confined to static settings.

However, the topology of *trajectories* in generative processes remains largely unexplored. Specifically, existing methods fail to capture the dynamic evolution of topological features along the generative path over time, nor do they reveal the relationship between this evolution and sampling stability or efficiency.

B PROJECTION STATISTICS

To quantify the projection bottleneck, we introduce diagnostic metrics that characterize the behavior of the executed timestep schedule.

Saturation ratio. We measure the fraction of timesteps that are clamped to the minimum step size:

$$r_{\min} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{1} \left[\Delta t_i^{\text{final}} = \Delta t_{\min} \right]. \tag{27}$$

A large $r_{\min}$ indicates that the schedule operates in a saturated regime where many degrees of freedom are constrained.

Deviation from base schedule. We quantify how much the executed schedule deviates from a reference allocation (e.g., uniform or phase-based):

$$D_{\text{base}} = \frac{1}{N} \sum_{i=0}^{N-1} \left| \Delta t_i^{\text{final}} - \Delta t_i^{\text{base}} \right|. \tag{28}$$

Schedule variation. We measure local variability of the executed schedule:

$$R_{\text{sched}} = \frac{1}{N} \sum_{i=1}^{N-1} \left| \Delta t_i^{\text{final}} - \Delta t_{i-1}^{\text{final}} \right|. \tag{29}$$

Empirically, we observe that step-wise adaptive controllers exhibit high saturation ratios (often $r_{\min} > 0.7$), indicating that most timesteps collapse to the minimum bound. In contrast, projection-free global allocation maintains low saturation and smoother schedules.

These statistics confirm that the projection operator operates in a highly constrained regime, where most coordinates are inactive and variations in raw proposals are suppressed. This behavior is consistent with the active-set analysis in Section 4: as more coordinates become saturated, fewer local proposal directions can influence the executed schedule.

C EXPERIMENTAL DETAILS

Implementation Details. All experiments are conducted on a server equipped with 4× NVIDIA RTX A5000 GPUs (24GB memory each), using CUDA 12.2 and driver version 535.230.02. We evaluate timestep scheduling under a strictly controlled inference setting, where model parameters are fixed and only the timestep schedule is varied. All methods are compared under matched NFE.

Models and Training. For Rectified Flow (RF), we use pretrained checkpoints from the official implementation Liu et al. (2022).[1] For diffusion models, we use DDPM and DDIM implementations based on the open-source repository Ho et al. (2020).[2] Models are trained on CIFAR-10 Krizhevsky (2009), MNIST LeCun et al. (2010), and Tiny ImageNet mnmoustafa & Ali (2017) with standard U-Net architectures. For CIFAR-10 and Tiny ImageNet, we use image resolutions of 32×32 and 64×64, respectively, with base channel dimension 128 and multiplier $(1, 2, 4)$. For MNIST, we use grayscale inputs with base dimension 64. Conditional models use class embeddings corresponding to dataset labels (10 classes for CIFAR-10/MNIST and 200 for Tiny ImageNet). For higher resolution evaluation, we additionally evaluate a pretrained Rectified Flow checkpoint on AFHQ-Cat Choi et al. (2020) at 256×256.

Trajectory Signal Extraction. Hidden representations are extracted from intermediate network layers (e.g., U-Net bottleneck features). At each timestep, features are treated as point clouds, from which multi-scale trajectory sensitivity signals are computed, including local geometric measures (speed and curvature) and global topological summaries (zigzag persistence). All signals are normalized and optionally smoothed via exponential moving averages.

[1] https://github.com/gnobitab/RectifiedFlow

[2] https://github.com/lucidrains/denoising-diffusion-pytorch

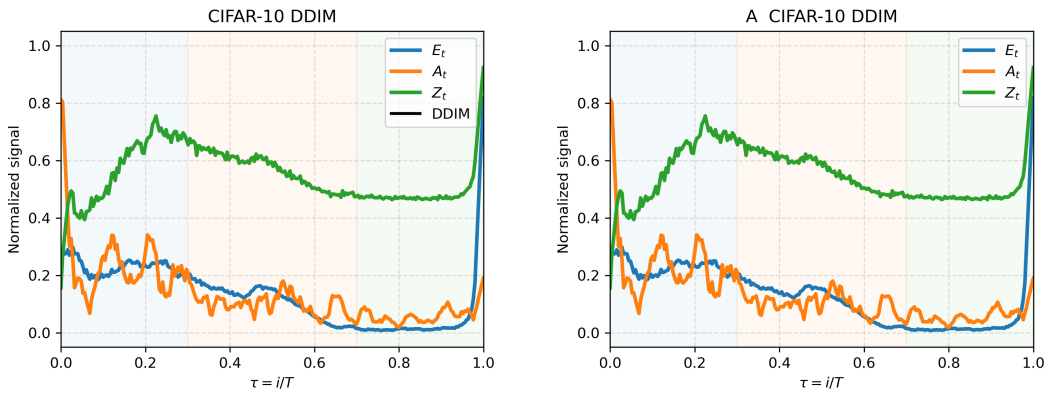


Figure 4: Trajectory sensitivity on CIFAR-10 using DDIM sampling. Both geometric quantities (E_t, A_t) and topological summaries (Z_t) exhibit stable temporal organization along the sampling trajectory.

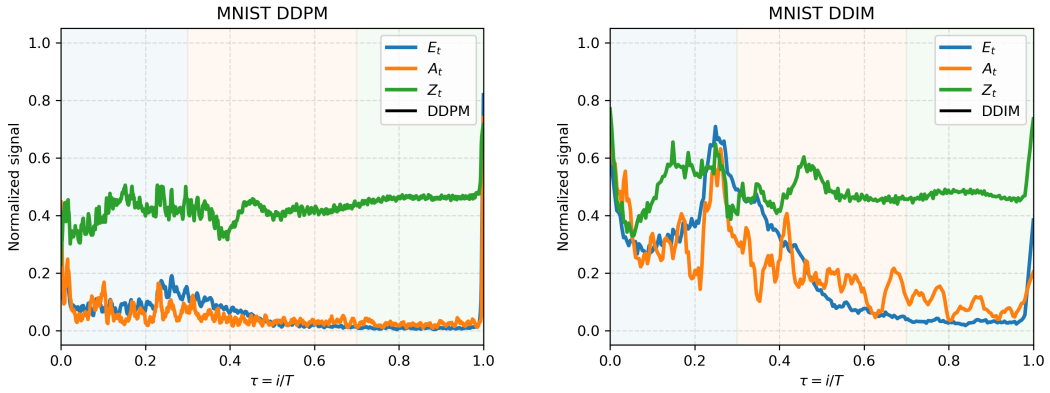


Figure 5: Trajectory sensitivity on MNIST using DDPM and DDIM sampling. The trajectory dynamics remain smooth and stable, reflecting the lower complexity of the underlying data manifold.

Scheduling and Evaluation. All timestep schedules are evaluated under strictly matched step budgets. We consider NFE levels ranging from low to high regimes, and report results averaged over multiple random seeds. Sampling quality is evaluated using Fréchet Inception Distance (FID), computed with the Clean-FID implementation of Parmar et al. (2022)[3].

Reproducibility. All hyperparameters are fixed across scheduling methods, and all evaluations use identical preprocessing and feature extraction pipelines to ensure fair comparison.

D Supplementary Results

D.1 Trajectory Sensitivity

We provide additional trajectory sensitivity visualizations in Figure 4, Figure 5, and Figure 6 across multiple datasets and generative model families, including DDPM, DDIM, and Rectified Flow on CIFAR-10, MNIST, AFHQ-Cat, and Tiny ImageNet. These results complement the main text by demonstrating that trajectory-level geometric and topological structure is highly consistent across architectures and sampling formulations.

[3] https://github.com/GaParmar/clean-fid

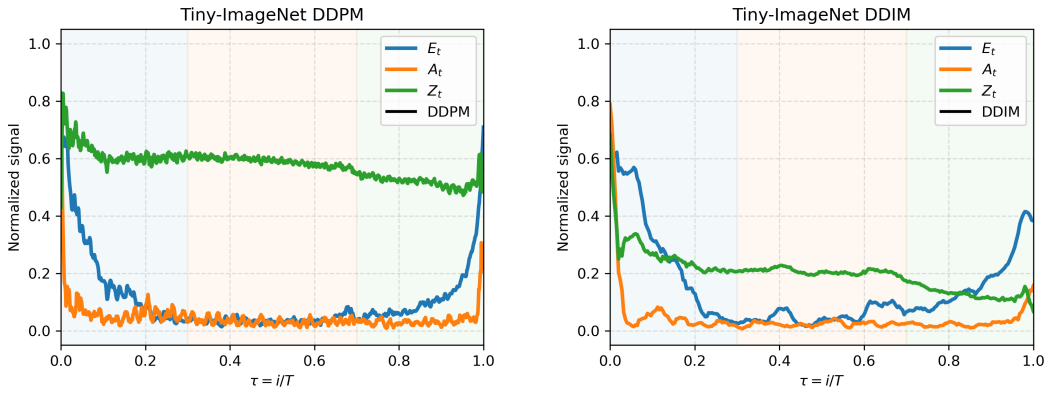


Figure 6: Trajectory sensitivity on Tiny ImageNet using DDPM and DDIM sampling. Compared to MNIST, the trajectories exhibit substantially richer geometric variation and stronger topological transition behavior.

Table 3: Projection statistics under strict matched-NFE scheduling. Higher $r_{\min}$ indicates stronger saturation at $\Delta t_{\min}$.

Method	$r_{\min}$	D_{base}	R_{sched}
NFE = 30			
Handcrafted Phase	0.400	0.0000	0.0029
E_t (step-wise)	0.473	0.0420	0.1097
$E_t + A_t$ (step-wise)	**0.900**	0.0483	0.1386
Global allocation	**0.033**	0.0027	0.0033
NFE = 50			
Handcrafted Phase	0.400	0.0000	0.0020
E_t (step-wise)	0.508	0.0260	0.0803
$E_t + A_t$ (step-wise)	**0.940**	0.0303	0.1101
Global allocation	**0.020**	0.0014	0.0022

D.2 Projection Statistics

To further quantify the projection bottleneck, we analyze statistics of the executed timestep schedules under strict matched-NFE settings. Table 3 reports three diagnostics: the saturation ratio $r_{\min}$, the deviation from the base schedule D_{base}, and the local schedule variation R_{sched}.

Several consistent patterns emerge. First, step-wise adaptive controllers exhibit severe saturation, with more than 90% of timesteps clamped at $\Delta t_{\min}$ for curvature-aware variants. This indicates that the projection operates in a highly constrained regime where most coordinates become inactive.

Second, the executed schedules produced by step-wise control deviate substantially from the intended allocation, as reflected by large values of D_{base} and R_{sched}. Although trajectory sensitivity signals generate diverse raw proposals, the projection suppresses these variations and concentrates the remaining budget into a small number of large steps.

In contrast, the proposed global allocation formulation maintains extremely low saturation ratios and smooth schedule variation across all budgets. Because the allocation distributes the sampling budget jointly without repeated step-wise projection, the resulting schedules avoid the severe saturation observed in step-wise controllers.

These observations are consistent with the active-set non-injectivity and local loss of controller degrees of freedom characterized in Section 4, where active constraints can increasingly suppress controller distinguishability as saturation increases.

Table 4: AFHQ-Cat (256×256) results. FID ↓, reported as mean±std over three seeds (500 samples/seed).

Method	**NFE=30**	**NFE=50**
Uniform	35.63±0.30	30.27±1.23
Handcrafted	34.90±0.17	29.06±0.86
Phase-aware topology	**27.22±3.13**	**25.41±1.07**

Table 5: Geometry-based controller ablation on CIFAR-10 under strict matched-NFE scheduling. Directly using trajectory sensitivity signals for step-wise adaptive control leads to severe performance degradation.

Method	**NFE=30**	**NFE=50**
Handcrafted Phase	**34.10** $\pm$ 0.37	**32.92** $\pm$ 0.35
E_t (speed only)	82.74 $\pm$ 0.72	76.03 $\pm$ 0.64
$E_t + 0.1A_t$	125.24 $\pm$ 1.32	133.64 $\pm$ 1.57
$E_t + 0.2A_t$	125.94 $\pm$ 1.43	133.64 $\pm$ 1.57
$E_t + 0.5A_t$	125.94 $\pm$ 1.43	133.64 $\pm$ 1.57

D.3 Higher-Resolution Evaluation on AFHQ-Cat

We additionally evaluate trajectory-aware scheduling on AFHQ-Cat at 256×256 resolution using Rectified Flow with Euler sampling. We consider a phase-aware topology schedule derived from the trajectory acceleration signal, which is distinct from the global allocation method in Section 3.4. All methods use the same checkpoint and initial noise, with 500 generated samples per seed over three seeds. As shown in Table 4, the phase-aware schedule improves FID over both handcrafted and uniform schedules at NFE=30 and 50.

D.4 Failure of Geometry-Based Adaptive Control

We further isolate the effect of geometry-based adaptive control using hidden trajectory sensitivity signals. Table 5 evaluates controllers constructed from hidden speed (E_t) and curvature (A_t) under strict matched-NFE settings on CIFAR-10.

Contrary to expectation, directly using trajectory sensitivity for step-wise adaptive control does not improve sampling quality and instead leads to catastrophic degradation. Using hidden speed alone increases FID from 34.10 to 82.74 at NFE=30 and from 32.92 to 76.03 at NFE=50. Incorporating curvature further amplifies this instability, with FID exceeding 125 at NFE=30 and 133 at NFE=50.

Notably, performance saturates across curvature-aware variants. At NFE=50, all controllers with $\lambda_A \in \{0.1, 0.2, 0.5\}$ produce identical FID values despite using different trajectory signals. Direct inspection confirms that these controllers generate identical executed schedules and identical samples across seeds.

These results reveal a severe failure mode of projection-based step-wise adaptive control. Although trajectory sensitivity signals are highly structured, their induced proposal differences can be eliminated after projection onto the feasible set. In saturated regimes, distinct controllers can therefore collapse to the same effective discretization, preventing trajectory-aware distinctions from influencing the executed schedule.