Bypassing Neural Evaluations for Fast Audio Editing via Adaptive Trajectory Extrapolation

Xiaoqian Liu[1*], Zhengkun Ge[1*], Jianjin Wang[1], Haoran Zhang[1], Yuan Ge[1], Kaiyan Chang[1], Chen Xu[2], Tong Xiao[1,3], Zhengtao Yu[4], Linfeng Zhang[5†], Jingbo Zhu[1,3†]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[3]NiuTrans Research, Shenyang, China [4]Kunming University of Science and Technology
[5]Shanghai Jiao Tong University , Shanghai, China
liuxiaoqian0319@outlook.com
zhanglinfeng@sjtu.edu.cn, zhujingbo@mail.neu.edu.cn

Abstract

Recent advancements in audio diffusion models have significantly improved text-to-audio editing via inversion techniques. However, these models typically rely on dense, fixed-step sampling trajectories to maintain structural integrity during inversion and generation, leading to prohibitive computational costs. We propose *AdaTE*, a model-agnostic *Ada*ptive *T*rajectory *E*xtrapolation framework that accelerates the inversion-based editing process by dynamically evaluating only the most critical generative phases. Specifically, we introduce a hierarchical probing mechanism that monitors curvature acceleration and information gain to detect pivotal transitions within the latent flow. This allows the model to selectively skip redundant segments via linear extrapolation while preserving dense neural evaluations for complex semantic changes. Extensive experiments across AudioLDM2, Auffusion, and Tango2 demonstrate that AdaTE achieves up to a 3.9× speedup with negligible loss in fidelity. AdaTE significantly shifts the Pareto frontier, providing an efficient solution for high-fidelity audio synthesis and editing.

1 Introduction

In recent years, text-to-audio (T2A) (Liu et al., 2023a; Kong et al., 2021) generation has achieved remarkable milestones, driven by the emergence of powerful generative frameworks such as latent diffusion models. Beyond simple generation, audio editing (Wang et al., 2023) has surfaced as a critical task that enables users to perform fine-grained modifications including the addition of sound events, the removal of noise, or the replacement of acoustic textures while preserving the global structure of the original signal (Xu et al., 2024). Most audio editing methods rely on inversion techniques (Mokady et al., 2023; Jia et al., 2025b) to map the source audio into a latent noise space before subsequently re-rendering it toward a target prompt. This inversion-based paradigm ensures high-fidelity content preservation and precise semantic alignment, making it the de facto standard for professional audio manipulation (Zhang et al., 2024; Li et al., 2024; Meng et al., 2022).

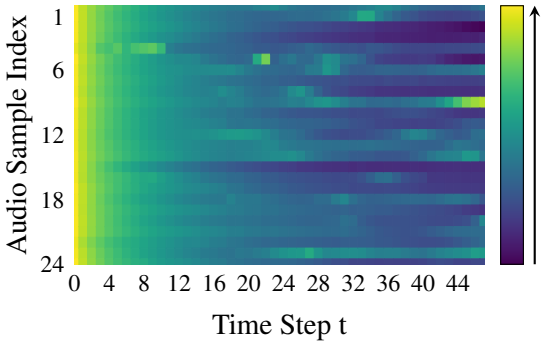


Figure 1: **Heatmap of Information Gain ($\mathcal{I}_t$) across 20 random audio samples.** During the inversion process, it shows a sharp concentration of informational density only in early time steps.

Despite their impressive synthesis quality, the practical utility of these models is severely constrained by their excessive computational overhead (Wallace et al., 2023; Kawar et al., 2023). To maintain the structural integrity of the audio and avoid artifacts during the inversion-to-generation cycle, these frameworks typically require dense, fixed-step sampling with 50 or more steps (Manor and Michaeli, 2024), leading to high inference latency (Song et al., 2021). While several acceleration methods such as DPM-Solver (Lu et al., 2022) has been proposed to reduce the number of function evaluations (NFE) (Salimans and Ho, 2022; Luo et al., 2023; Lu et al., 2025), they often suffer from a significant drop in fidelity or structural drift when the step count is drastically reduced, failing to capture the intricate nuances of complex audio

* Equal contribution.
† Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 16633–16647
July 2-7, 2026

signals (Huang et al., 2022; Ye et al., 2023).

We observe that the limitation of existing methods stems from the inherent non-uniformity and local geometric properties of the generative trajectory. As illustrated in Figure 1, our analysis of information gain ($\mathcal{I}_t$) reveals that generative cues are highly concentrated in specific stages, suggesting that a uniform computational budget is inherently suboptimal. Also, we observe that the PCA-projected trajectory in Figure 2 exhibits a pattern of local smoothness. This behavior provides an intuitive illustration of locally stable segments along the trajectory which serves as a useful empirical motivation for our extrapolation strategy.

These insights suggest that an optimal acceleration strategy should not only be information-aware to detect critical transitions, but also leverage the local linearity of the latent manifold for accurate state estimation. We propose AdaTE, a model-agnostic adaptive trajectory extrapolation framework that accelerates inference by prioritizing informative evaluation phases. Specifically, a hierarchical probing mechanism concurrently monitors trajectory curvature and information gain to distinguish between pivotal semantic transitions and redundant segments. AdaTE maintains dense neural evaluations in high-curvature regions to ensure fidelity, while skipping stable intervals via linear extrapolation upon detecting local smoothness. This plug-and-play strategy allows for seamless integration into existing diffusion-based backbones, significantly shifting the Pareto frontier of audio editing with minimal computational overhead.

Our contributions are summarized as follows:

- **Theoretical Foundation:** We characterize the latent trajectory of audio generative models as a manifold of non-uniform informational density, providing a geometric foundation for adaptive inference beyond traditional fixed-step sampling paradigms.
- **Adaptive Framework:** We propose AdaTE, a model-agnostic framework that employs hierarchical probing to dynamically balance computational budget between critical transitions and locally linear segments.
- **Empirical Validation:** We achieve up to $3.9\times$ speedup across AudioLDM2, Auffusion, and Tango2 without retraining, thereby shifting the Pareto frontier to establish a robust method for high-fidelity audio editing.

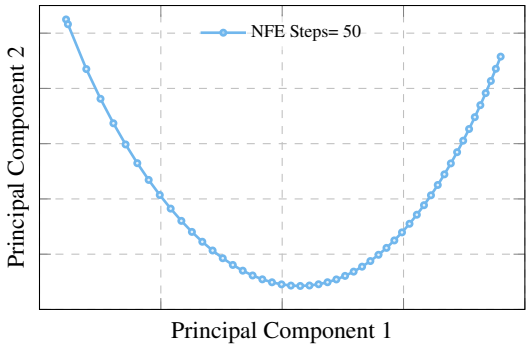


Figure 2: **Two-dimensional principal component analysis of a representative latent trajectory.** The locally smooth pattern in the projected space provides an intuitive visualization of short stable segments. This figure is illustrative only; AdaTE makes its bypassing decisions in the original high-dimensional space.

2 Preliminaries

2.1 Deterministic Audio Diffusion Dynamics

Diffusion-based models define a forward process that transforms an audio latent state z_t from data ($t = 0$) to Gaussian noise ($t = T$). Audio editing typically leverages the Probability Flow Ordinary Differential Equation (ODE) to ensure deterministic and reversible manipulation. The evolution of the latent trajectory $\{z_t\}_{t=0}^{T}$ is governed by:

$$\frac{dz_t}{dt} = v(z_t, t) \tag{1}$$

where $v(z_t, t)$ denotes the ODE velocity field governing the deterministic sampling dynamics. In practice, this field is instantiated from the backbone output $\epsilon_\theta(z_t, t)$, whose parameterization may vary across models (e.g., noise or velocity prediction). Throughout the remainder of the paper, we use v_t as a shorthand for the model output, i.e., $v_t := \epsilon_\theta(z_t, t)$, and reserve $v(z_t, t)$ exclusively for the ODE velocity field in Eq. (1) and Eq. (2).

2.2 Deterministic Inversion and NFE

Deterministic inversion maps the signal into the latent space to facilitate granular editing while preserving its acoustic identity. A standard first-order solver updates the state as:

$$z_{t+\Delta t} = z_t + v(z_t, t) \cdot \Delta t \tag{2}$$

The computational cost is quantified by the Number of Function Evaluations (NFE), representing the total forward passes through ϵ_θ. NFE is intrinsically coupled with the number of steps

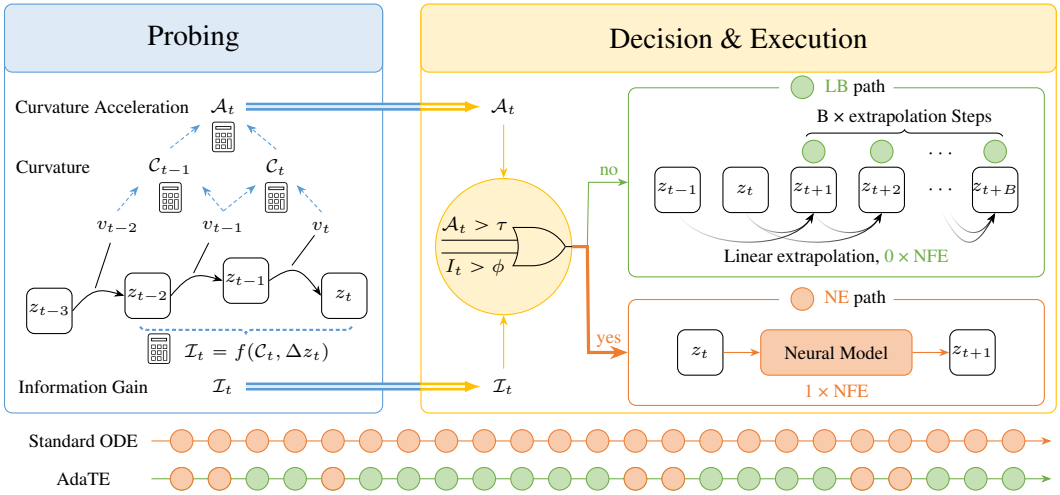


Figure 3: **Overview of the AdaTE framework.** The system consists of three primary stages: (1) **Probing**: A hierarchical mechanism concurrently calculates curvature acceleration ($\mathcal{A}_t$) and information gain ($\mathcal{I}_t$) from the latent trajectory. (2) **Decision & Execution**: Based on thresholding logic, the framework selects between the Neural Evaluation (NE) path for high-complexity transitions or the Linear Bypassing (LB) path for stable regions. The bottom timeline illustrates how AdaTE achieves a significantly lower NFE compared to standard ODE solvers by adaptively skipping redundant steps.

N (i.e., NFE $=$ N) in conventional fixed-step schemes. While maintaining audio integrity requires a dense discretization to minimize phase and structural distortions, the resulting computational overhead incurs prohibitive latency. Our objective is to decouple NFE from N by bypassing neural evaluations in stable trajectory segments, enabling high-resolution processing with significantly fewer model calls.

3 Methodology

As illustrated in Figure 3, AdaTE operates by dynamically switching between precise neural evaluations and efficient linear approximations based on the intrinsic geometric properties of the latent trajectory. AdaTE framework facilitates this acceleration via a modular three-stage pipeline: (i) **Information-Aware Probing** for real-time dynamics estimation, (ii) **Adaptive Bypassing Decision** for stability evaluation, and (iii) **Budgeted Execution** for optimized state propagation.

3.1 Information-Aware Probing

The Probing module monitors the evolution of latent states $\{z_t\}_{t=0}^{T}$. It computes the curvature C_t, its acceleration A_t, and the information gain I_t from consecutive model outputs, providing deterministic signals for the subsequent decision-making stage.

Directional Curvature ($\mathcal{C}_t$) We define the curvature C_t as the temporal variation rate of the model output v_t, where $v_t := \epsilon_\theta(z_t, t)$ denotes the backbone output at time t. Note that v_t is used here as a shorthand for the model output and is not assumed to be identical to the ODE velocity field $v(z_t, t)$ in Eq. (1):

$$\mathcal{C}_t = \frac{\|v_t - v_{t-\Delta t}\|_2}{\Delta t} \tag{3}$$

where a vanishing C_t indicates that consecutive model outputs vary smoothly, suggesting a locally stable and predictable trajectory segment.

Curvature Acceleration ($\mathcal{A}_t$) To detect the onset of structural transitions, we monitor the curvature acceleration A_t, which captures the second-order variation of the model-output trajectory. This metric represents the temporal gradient of the curvature and is formulated as:

$$\mathcal{A}_t = \frac{\mathcal{C}_t - \mathcal{C}_{t-\Delta t}}{\Delta t} \tag{4}$$

By evaluating the rate of change in curvature, $\mathcal{A}_t$ serves as a sensitive indicator of manifold transitions. High values of $\mathcal{A}_t$ effectively identify mutation phases where the latent flow undergoes abrupt structural updates.

Metric of Information Gain ($\mathcal{I}_t$) Information gain $\mathcal{I}_t$ is formulated as a dimensionless scalar that dictates the necessity of a neural evaluation at step t. To quantify the evolution of the generative path, we define the latent displacement operator $\Delta z_t = \|z_t - z_{t-\Delta t}\|_2$ as the ℓ_2-norm of the state transition between successive probing steps. $\mathcal{I}_t$ integrates the normalized output variation with the relative evolution of the latent state:

$$\mathcal{I}_t = \underbrace{\alpha \cdot \mathcal{C}t}_{\text{Output Variation}} + \underbrace{(1-\alpha) \cdot \frac{\Delta z_t}{\Delta t}}_{\text{State Evolution}} \quad (5)$$

where $\alpha \in [0, 1]$ is a balancing hyperparameter that modulates the sensitivity between prediction volatility and geometric displacement. By consolidating these dual perspectives, $\mathcal{I}_t$ provides a robust proxy for the total informational density of the current sampling step.

3.2 Adaptive Bypassing Decision

In the decision-making stage, the system evaluates the stability metrics against predefined thresholds to gatekeep the neural network invocations. This mechanism ensures that computational resources are concentrated on the most informative segments of the latent trajectory.

Triggering Mechanism For each step t, a neural evaluation (NE) is mandatory if either the cumulative information gain or the structural mutation exceeds the safety bounds:

$$\text{Trigger NE if: } \mathcal{I}_t > \phi \quad \text{or} \quad \mathcal{A}_t > \tau \quad (6)$$

where ϕ denotes the information tolerance and τ represents the mutation threshold. If neither condition is met, the system enters the linear bypass (LB) mode, identifying the current segment as a stable phase where the flow is predictable.

> **Stability Intuition:** The local extrapolation error is governed by higher-order variation of the trajectory. Since A_t serves as a discrete proxy for such variation, thresholding A_t helps identify segments where short-range extrapolation is likely to remain stable. In addition, periodic probing resets the approximation, which empirically limits error accumulation over long trajectories.

3.3 Budgeted Execution

The execution stage translates the stability diagnosis into computational compression by dynamically modulating the skipping interval.

Adaptive Budget Calculation Upon entering bypass mode, the system calculates a budget B, which defines the number of subsequent steps to be extrapolated without invoking ϵ_θ. To ensure the skipping window is inversely proportional to the informational density, B is formulated as follows:

$$B = \min\left(\left\lfloor \frac{\gamma \cdot \phi}{\mathcal{I}_t} \right\rfloor, B_{\max}\right) \quad (7)$$

where γ is a scaling factor and the term $B_{\max}$ serves as a safety constraint, ensuring trajectory stability by preventing excessive deviation.

Trajectory Extrapolation During the skipping window defined by B, the system bypasses all neural evaluations and propagates the latent state using a diffusion scheduler $\mathcal{S}$:

$$z_{t+k\Delta t} = \mathcal{S}\big(z_{t+(k-1)\Delta t}, t + (k-1)\Delta t, \hat{v}_t\big) \quad (8)$$

where $k \in \{1, \ldots, B\}$, and $\hat{v}_t$ is obtained via first-order extrapolation from the two most recent model outputs. This linear extrapolation allows the framework to traverse stable segments of the latent manifold with negligible computational overhead. By substituting costly function evaluations with first-order approximations in redundant regions, AdaTE effectively decouples the NFE from the total discretization steps N. The resulting efficiency gain is visualized in the bottom timeline of Figure 3, where green nodes represent the accelerated LB path.

4 Experiments

4.1 Backbones and Task Construction

Backbones and Dataset Extensive experiments are conducted across three representative backbones: **AudioLDM2** (Liu et al., 2024a), which utilizes a language modeling approach to enhance the temporal consistency of latent diffusion; **Auffusion** (Xue et al., 2024), a T2I-adapted framework that leverages pre-trained visual priors and cross-modal attention for robust audio-visual alignment; and **Tango2** (Majumder et al., 2024), a model based on diffusion policy optimization that excels in complex text-to-audio mapping. These backbones cover a spectrum of architectures, ensuring a comprehensive evaluation.

Method	NFE	Speedup	Adding				Removing				Replacing			
			CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑
Backbone: AudioLDM2														
DDIM Inv.	50	1.0×	39.63	2.186	3.385	3.683	43.94	2.027	**2.853**	**3.683**	41.79	2.357	**2.847**	**3.064**
DDIM Inv.	15	3.3×	40.75	**2.095**	3.784	3.659	43.71	**1.859**	3.270	3.305	**42.41**	**2.247**	3.105	2.850
+ DPM	15	3.3×	35.02	2.396	7.442	2.742	37.94	2.381	6.670	2.497	36.74	2.554	6.689	2.266
+ AdaTE	12.8	3.9×	**40.88**	2.214	**3.104**	**3.704**	**44.65**	2.109	3.308	3.466	41.50	2.486	3.060	2.978
Backbone: Auffusion														
AudioEditor	25	1.0×	**52.07**	2.496	2.572	4.932	**49.88**	2.389	2.655	**4.920**	53.43	2.578	**2.094**	**4.776**
AudioEditor	15	1.7×	49.84	**2.315**	3.204	4.357	47.58	2.290	3.107	4.236	50.88	2.426	2.843	4.177
+ DPM	15	1.7×	49.17	2.408	3.015	4.576	47.88	2.374	3.335	4.547	51.51	**2.513**	2.904	4.417
+ AdaTE	13.6	1.8×	51.87	2.405	**2.571**	**5.140**	49.08	**2.240**	**2.650**	4.626	**53.45**	2.538	2.315	4.701
Backbone: Tango2														
AudioMorphix	50	1.0×	**41.71**	**1.315**	**2.777**	**4.018**	**43.17**	**2.436**	**8.137**	**3.050**	**39.95**	**2.546**	**7.621**	**3.171**
AudioMorphix	15	3.3×	41.22	1.520	5.359	3.821	34.92	3.586	17.59	2.074	32.72	3.394	17.84	2.048
+ DPM	15	3.3×	41.20	1.518	5.360	3.818	35.12	3.596	17.60	2.084	32.85	3.390	17.84	2.051
+ AdaTE	15.7	3.2×	40.29	1.492	3.110	3.625	37.11	2.861	9.367	2.907	35.70	2.673	8.798	2.919

Table 1: **Main Results on Audio Editing Efficiency and Quality.** We evaluate AdaTE across three backbones (AudioLDM2, Auffusion, and Tango2) on three tasks (Adding, Removing, and Replacing). **NFE** represents the average number of neural evaluations across the test set, calculated as $\mathbb{E}[\text{NFE}] = \frac{1}{n}\sum_{i=1}^{n} \text{NFE}_i$, where NFE_i is the actual evaluation steps for the i-th sample. **Speedup** is measured relative to the standard setting of the original backbone. **Bold** and underlined values indicate the best and second-best performance within each backbone.

Task Construction We primarily evaluate audio editing on **AudioSet**[1], and further assess generalization to music editing on **BabySlakh**[2]. We construct three core editing tasks with well-defined source-target pairs by manipulating sound event segments within background audio: **(1) Adding**: A sound event is inserted into background audio; the pre- and post-insertion audios are the *Source* and *Target*, respectively. **(2) Removing:** The inverse of Adding. **(3) Replacing**: Two distinct sound events are inserted into the same background position, forming the *Source-Target* pair. Metric definitions are provided in Appendix A.1.

Baselines We benchmark AdaTE against several paradigms: **(1) Deterministic Solvers:** We employ standard DDIM Inversion (Song et al., 2021), AudioMorphix (Liang et al., 2025), and AudioEditor (Jia et al., 2025b) at varying NFEs as the primary baselines for quality-speed anchors. **(2) Advanced Solvers:** We additionally include DPM-Solver (Lu et al., 2022) as a representative high-order solver, and further extend comparisons to advanced samplers and adaptive ODE solvers such as RK45 (P. Bogacki, 1989), DPM++ (Lu et al., 2025), and UniPC (Zhao et al., 2023).

[1]https://research.google.com/audioset/download_strong.html
[2]https://zenodo.org/records/4603870

4.2 Main Results

4.2.1 Overall Performance and Efficiency

As summarized in Table 1, AdaTE achieves a superior Pareto front between computational efficiency and generative quality across all evaluation benchmarks. Specifically, on the AudioLDM2 backbone, AdaTE reduces the NFE to 12.8 (a 3.9× speedup) while consistently outperforming the competitive DPM-solver. In many instances, such as the Adding task for AudioLDM2, AdaTE even exceeds the fidelity (FAD: 3.104 vs. 3.385) and inception scores of the 50-step DDIM Inversion. This suggests that by dynamically skipping redundant trajectory segments, AdaTE effectively prevents the accumulation of numerical errors inherent in dense, fixed-step sampling.

4.2.2 Robustness Against Trajectory Collapse

A critical finding is the resilience of AdaTE in scenarios where standard fast-sampling methods fail. This is most evident on the Tango2 backbone. When the NFE is reduced to 15, the baseline AudioMorphix suffers a catastrophic performance drop in Removing and Replacing tasks (e.g., FAD soaring to 17.59). Under nearly identical NFE constraints, AdaTE maintains structural integrity with an FAD of 9.367 and 8.798, respectively. This performance gap validates our

Method	NFE	Speedup	CS↑	KL↓	FAD↓	IS↑
DDIM	50	1.0×	41.79	2.190	**3.028**	**3.477**
DDIM	15	3.3×	42.29	**2.067**	3.386	3.271
+ RK45	14	3.6×	9.92	3.754	28.735	1.140
+ DPM++	15	3.3×	19.88	2.764	8.897	2.647
+ UniPC	15	3.3×	40.06	2.296	4.637	2.967
+ AdaTE	13	3.9×	**42.34**	2.270	3.157	3.383

Table 2: **Comparison with Advanced ODE Solvers on AudioLDM2.** We compare AdaTE with advanced solvers including RK45, DPM++ and UniPC under identical settings.

hierarchical probing mechanism: by identifying high-curvature regions that require neural evaluation, AdaTE preserves essential manifold transitions that fixed-step solvers overlook.

4.2.3 High-Fidelity Content Preservation

Beyond global distribution metrics like FAD and IS, AdaTE excels in maintaining semantic alignment and content preservation, as evidenced by the competitive CS and KL scores. On Auffusion, AdaTE achieves the highest CS and IS scores in the Adding and Removing tasks among 15-step alternatives. This indicates that our information-aware strategy ensures that the essential semantic features of the audio are computed with high precision, while only geometrically stable and thus predictable segments are bypassed.

4.2.4 Comparison with Advanced Solvers

We extend our comparison to include stronger high-order and adaptive sampling methods, including DPM-Solver++, UniPC, and RK45. We evaluate these baselines under the same AudioLDM2 backbone and identical NFE settings for a fair comparison. As shown in Table 2, we report the average performance over the three editing tasks, while the full results are provided in Appendix B.1. While these higher-order and adaptive solvers can improve sampling efficiency in moderate settings, their performance tends to degrade when the number of function evaluations is heavily reduced. AdaTE maintains a better trade-off between efficiency and fidelity. This suggests that improving numerical solvers alone may be insufficient for robust audio editing, and that adaptive trajectory selection provides a more stable mechanism under aggressive acceleration.

Method	NFE	Speed	CS↑	KL↓	FAD↓	IS↑
AudioLDM2 (Music Editing)						
DDIM	50	1.0×	43.65	0.708	2.582	**1.397**
DDIM	15	3.3×	43.94	**0.696**	**2.369**	1.349
+ AdaTE	10.3	4.9×	**44.01**	0.703	2.596	1.392
Auffusion (Music Editing)						
AudioEditor	25	1.0×	40.52	1.209	5.458	**1.700**
AudioEditor	15	1.7×	40.00	1.160	5.614	1.610
+ AdaTE	13.2	1.9×	**42.18**	**1.156**	**4.708**	1.632

Table 3: **Main Results on Music Editing.** We evaluate AdaTE across different backbones on the BabySlakh dataset. Results are reported as the average performance across three editing tasks. **Bold** and underlined values indicate the best and second-best performance within each backbone.

4.2.5 Subjective Evaluation

We evaluated perceptual fidelity using a Mean Opinion Score (MOS) test involving 25 participants on the AudioLDM2 and Auffusion backbones. Participants rated audio samples generated by different methods, and the reported MOS values are averaged over all valid ratings. Overall, AdaTE consistently outperforms the low-NFE baseline methods and achieves perceptual quality comparable to the original high-NFE methods. Detailed per-task subjective results are provided in Appendix B.2.

4.2.6 Inference Efficiency and Latency

To verify that our NFE reduction translates into actual hardware-level speedup, we measure the Real-Time Factor (RTF) for each backbone under the hardware setup available in our experiments. The reported RTF values are intended for within-backbone comparison between AdaTE and its corresponding baseline under the same hardware setting. The results indicate that AdaTE significantly reduces end-to-end latency; for instance, it compresses the RTF of AudioLDM2 from 1.167 to 0.324. Importantly, this efficiency gain is consistent across diverse architectures. Detailed RTF comparisons are provided in Appendix B.3.

4.2.7 Music Editing

We further evaluate AdaTE on the BabySlakh dataset, a synthetic music dataset characterized by structured multi-track compositions and strong harmonic dependencies. Compared to AudioSet, it exhibits more complex temporal and spectral structures, posing a greater challenge for high-fidelity editing. As shown in Table 3 (which re-

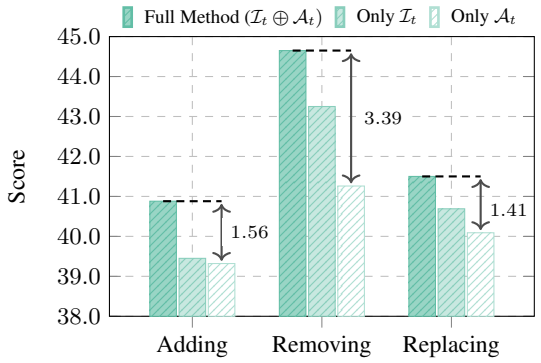


Figure 4: **Ablation study of the dual-metric decision module on AudioLDM2.** We report the CLAP scores with all variants evaluated under a consistent $\sim 3.9\times$ speedup. The results show that the synergy between $\mathcal{I}_t$ and $\mathcal{A}_t$ is essential for maintaining semantic alignment.

ports the average performance across all tasks; see Appendix B.4 for the full per-task breakdown), AdaTE consistently achieves better or comparable performance to full-NFE baselines while providing significant speedups across diverse musical structures. On AudioLDM2, it maintains a balanced performance profile under low-NFE settings, while on Auffusion it consistently improves most metrics over the AudioEditor baseline. These results demonstrate that AdaTE generalizes well to music scenarios and remains robust even under aggressive acceleration.

5 Ablation Studies and Analysis

We conduct a series of progressive ablation experiments to validate the design choices of AdaTE. We analyze (1) the necessity of decision metrics, (2) the formulation of information gain, (3) hyperparameter sensitivity, (4) the statistical foundation of dynamic budget allocation, and (5) case study.

5.1 Dual-Metric Synergy: Information Gain $\mathcal{I}_t$ vs. Curvature Acceleration $\mathcal{A}_t$

The decision-making module of AdaTE is driven by the synergy between $\mathcal{I}_t$ which monitors semantic tolerance, and $\mathcal{A}_t$ which captures abrupt structural changes. We evaluate the semantic alignment using CS score across the editing tasks while maintaining a $\sim 3.9\times$ speedup for all variants.

As detailed in Figure 4, the Full Method consistently outperforms the single-metric variants across all tasks. Specifically, removing $\mathcal{A}_t$ (*Only* $\mathcal{I}_t$) causes CLAP scores to drop, notably in Removing tasks (44.65 to 43.25), suggesting that second-order trajectory shifts are vital for precise segment excision. The degradation is more severe without $\mathcal{I}_t$ (*Only* $\mathcal{A}_t$), where the average CS score falls to 40.22, and the Removing task collapses by 3.39 points. These results empirically justify our dual-metric design: $\mathcal{A}_t$ ensures structural precision, while $\mathcal{I}_t$ maintains semantic sensitivity during complex audio editing.

5.2 Formulating Information Gain $\mathcal{I}_t$: Time-Relative vs. State-Recursive

To investigate the scientific rationality of information gain metric, we formalize and compare two contrastive strategies. **Time-Relative $\mathcal{I}_t^{(1)}$:** defined as $\mathcal{I}_t^{(1)} \propto \Delta z_t / \Delta t$, which monitors the evolution rate relative to the fixed temporal grid, viewing information density as the velocity of latent drift; **State-Recursive $\mathcal{I}_t^{(2)}$:** defined as $\mathcal{I}_t^{(2)} \propto \Delta z_t / \Delta z_{t-\Delta t}$, which measures the relative innovation by anchoring the current displacement to the preceding state increment.

As illustrated in Figure 5, the time-relative $\mathcal{I}_t^{(1)}$ consistently outperforms the recursive $\mathcal{I}_t^{(2)}$ in audio fidelity. For instance, in the Adding task, $\mathcal{I}_t^{(1)}$ achieves a significantly higher CLAP score (40.88 vs. 38.14) and a lower FAD (3.104 vs. 3.812). The performance gap suggests that recursive normalization $\mathcal{I}_t^{(2)}$ is highly sensitive to local numerical fluctuations. When the trajectory enters a near-stationary region where $\Delta z_{t-\Delta t} \to 0$, the metric may encounter instability, leading to inaccurate bypassing decisions. In contrast, time-normalized $\mathcal{I}_t^{(1)}$ provides a globally consistent and stable measure of informational density, ensuring that linear extrapolations are grounded in the intrinsic physical flow of the diffusion process. Thus, we adopt $\mathcal{I}_t^{(1)}$ as the default formulation for AdaTE.

5.3 Efficiency-Fidelity Trade-off: Sensitivity of Threshold ϕ

The information tolerance threshold ϕ governs the aggressiveness of the acceleration. To determine the optimal operating point, we conduct a sensitivity analysis by varying ϕ from 1 to 10 as shown in Table 4. Audio quality remains remarkably stable within $\phi \in [1, 5]$; notably, for the Adding task, $\phi = 5$ slightly improves the FAD score (3.104) over the conservative $\phi = 1$ (3.339). While increasing the threshold consistently reduces computational cost, an excessively aggressive policy ($\phi = 10$) may overlook critical semantic transi-

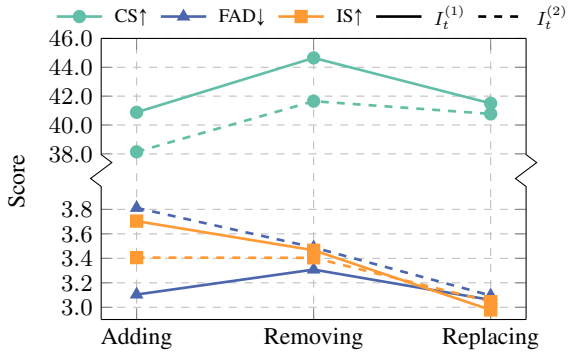


Figure 5: **Ablation Study on Information Gain Formulation.** We compare our proposed formulation $\mathcal{I}_t^{(1)}$, which computes information density relative to the time step Δt, against a recursive variant $\mathcal{I}_t^{(2)}$ that evaluates innovations relative to the preceding state $\Delta z_{t-\Delta t}$.

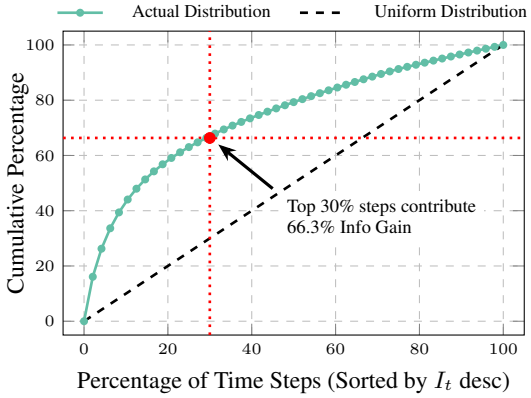


Figure 6: **Cumulative Distribution of Information Gain $\mathcal{I}_t$.** The sorted distribution reveals that information is heavily concentrated in a minority of critical steps, justifying an adaptive bypassing strategy.

Task	ϕ	Speedup	CS ↑	FAD ↓	IS ↑
Adding	1	1.6×	40.36	3.339	**3.710**
	5	4.0×	**40.88**	**3.104**	3.704
	10	6.6×	39.81	4.095	3.381
Removing	1	1.5×	**44.71**	**2.899**	**3.588**
	5	3.9×	44.65	3.308	3.466
	10	6.6×	43.46	3.214	3.378
Replacing	1	1.5×	**42.14**	**2.802**	**3.003**
	5	3.8×	41.50	3.060	2.978
	10	6.6×	41.09	3.285	2.862

Table 4: **Ablation Study on Information Tolerance ϕ.** As the primary controller of the execution budget, a larger ϕ aggressively compresses computation while a smaller ϕ prioritizes fidelity. Results on AudioLDM2 show that $\phi = 5$ provides up to $4.0\times$ speedup with negligible degradation, achieving an optimal balance.

tions, leading to a noticeable degradation in CS and IS metrics. We identify $\phi = 5$ as the optimal "sweet spot" for balancing compression and structural integrity, adopting it as default configuration.

5.4 Justifying Dynamic Budget via Information Density

We provide a statistical foundation for our dynamic budget allocation policy by analyzing the distribution of information gain $\mathcal{I}_t$ across the generative trajectory. As illustrated in Figure 6, the information density is highly non-uniform and exhibits a pronounced long-tail characteristic, where the top 30% of time steps account for 66.3% of the total information gain. This uneven distribution justifies our strategy of assigning a dynamic budget B that is inversely correlated with information gain $\mathcal{I}_t$. By allocating more neural evaluations to high-information phases and aggressively bypassing redundant steps in the stable tail, AdaTE achieves a much more efficient computational distribution compared to static uniform skipping.

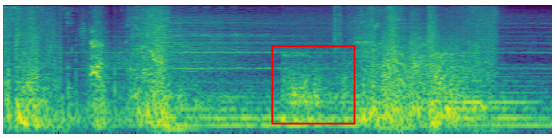
5.5 Case Study: Mel-spectrogram Visualizing

Finally, we present a case study of an audio editing task to provide a granular understanding of how AdaTE operates on real-world samples. Figure 7 visualizes the original spectrogram, the 50-step full-inversion baseline, and the edited result via AdaTE. A comparison between (a) and (b) reveals that the inversion process precisely isolates and modifies the target acoustic regions. Crucially, the spectrogram produced by AdaTE (c) remains nearly indistinguishable from the non-accelerated reference (b), preserving fine-grained harmonic structures and transient details despite a substantial reduction in neural evaluations. This qualitative evidence demonstrates that our information-aware policy effectively guards the perceptual quality of the audio while skipping redundant computations in stable regions.

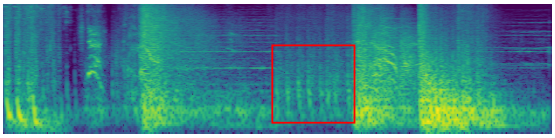
6 Related Work

6.1 Trajectory-based Audio Editing

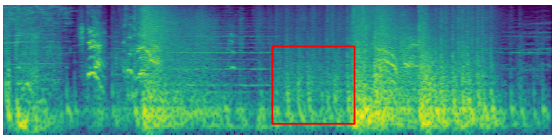
Diffusion models and Flow Matching dominate audio editing by mapping trajectories between noise and data distributions (Liu et al., 2023a; Guo et al., 2024; Majumder et al., 2024), ensuring semantic consistency in tasks like inpainting and style transfer (Xu et al., 2024; Xue et al., 2024). However, editing requires solving Differential Equations via iterative sampling. This in-



(a) Original source audio



(b) DDIM Inversion



(c) DDIM Inversion + AdaTE

Figure 7: **Mel-spectrograms Comparison.** (a) Original audio; (b) Full DDIM-inversion baseline (50 steps); (c) Accelerated result via AdaTE. While (b) highlights the edited regions, (c) is nearly indistinguishable from (b), demonstrating that our adaptive strategy preserves fine-grained harmonic structures and transient details despite significant computational speedup.

volves multiple NFEs of networks, hindering low-latency interactive editing and edge deployment.

Inversion-based methods map source audio to latents to maintain structural integrity. Although common, DDIM inversion (Song et al., 2021) suffers from reconstruction-distorting accumulation errors. Recent works like Null-text (Mokady et al., 2023), Negative-prompt (Miyake et al., 2025), or fixed-point iteration (Mehta et al., 2024) improve editability-fidelity trade-offs. Although FM allows exact inversion via backward integration (Le et al., 2023; Vyas et al., 2023), it still requires dense ODE discretization to minimize error, remaining computationally expensive for real-time use.

6.2 Numerical Solvers and Step-size Control

The inference of generative trajectories is typically treated as solving an Initial Value Problem (IVP). Standard solvers such as DPM-Solver (Lu et al., 2022), DEIS (Zhang and Chen, 2023), and UniPC (Zhao et al., 2023) utilize higher-order polynomial approximations to achieve faster convergence to reduce the number of sampling steps. And adaptive step-size solvers like the Bogacki-Shampine (P. Bogacki, 1989) and Runge-Kutta-Fehlberg (Fehlberg, 1969) methods have been integrated into diffusion frameworks (Song et al., 2021; Jolicoeur-Martineau et al., 2021) to adjust discretization intervals based on local truncation error estimates. Unfortunately, these general-purpose numerical solvers are often task-agnostic as they prioritize global numerical precision over the specific semantic stability required in audio editing. In many editing scenarios, large segments of the latent trajectory exhibit low informational density. Traditional solvers lack a mechanism to sense these regions from a semantic perspective.

6.3 Accelerating via Adaptive Computation

Acceleration strategies are generally categorized into training-based and training-free paradigms. Training-based methods focus on distilling complex models into simpler ones or straightening the generation trajectories. For instance, Consistency Models (Song et al., 2023) and the variants in audio domain (Lu et al., 2024) collapse the trajectory into a few steps. Similarly, Rectified Flow (Liu et al., 2023b) and progressive distillation (Salimans and Ho, 2022) aim to create straighter ODE paths. While effective, these methods require expensive retraining on massive datasets and may suffer from a loss of generative diversity or fidelity when the distillation budget is extremely low.

Training-free approaches optimize inference by pruning redundancy or leveraging temporal coherence. Feature caching mechanisms like Fora (Selvaraju et al., 2024) and DeepCache (Ma et al., 2024) exploit the similarity within adjacent states for reusing. Furthermore, architectural-level optimizations like token pruning (Wen et al., 2025) and dynamic block skipping (Jia et al., 2025a) are also explored to reduce the per-step cost. But these methods often operate under static schedules or rely on fixed heuristic thresholds.

7 Conclusion

We introduce a model-agnostic framework that accelerates audio editing via adaptive trajectory extrapolation. By monitoring curvature acceleration and information gain, AdaTE identifies critical ODE segments for full evaluation while bypassing redundant phases through linear extrapolation. Experiments across three backbones show up to $3.9\times$ speedup without compromising quality. Statistical analysis confirms that 66.3% of generative information is concentrated in the first 30% of steps, empirically justifying our adaptive budgeting. Overall, AdaTE provides a robust, high-fidelity solution that effectively decouples computational cost from sampling density.

Limitations

Despite the significant acceleration achieved by AdaTE, this work primarily focuses on the temporal redundancy and local linearity of the latent trajectory. One potential limitation is that we do not explicitly explore the spatial redundancy within the latent representations of audio diffusion models. Since audio signals often exhibit sparse structures in the latent manifold, future research could investigate adaptive spatial pruning or sparse computation to further reduce the computational footprint. Additionally, while our framework is model-agnostic, the probing intervals currently rely on a set of predefined hyperparameters. Developing an end-to-end learnable policy to determine these critical windows could be a promising direction for future exploration. Another limitation is that our study is restricted to audio diffusion and editing pipelines, and we have not yet examined whether similar adaptive acceleration strategies remain effective in other large-scale generative settings, such as multilingual LLM-based generation (Luo et al., 2025), where the inference dynamics may differ substantially from those in audio models.

Acknowledgments

This work was supported in part by the National Science Foundation of China (Nos. 62276056 and U24A20334), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023a. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Benjamin Elizalde, Soham Deshmukh, and Huaming Wang. 2023b. Natural language supervision for general-purpose audio representations. *Preprint*, arXiv:2309.05767.

Erwin Fehlberg. 1969. *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, volume 315. National aeronautics and space administration.

Yiwei Guo, Chenpeng Du, Ziyang Ma, Xie Chen, and Kai Yu. 2024. Voiceflow: Efficient text-to-speech with rectified flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11121–11125. IEEE.

Rongjie Huang, Max W. Y. Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. 2022. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4157–4163. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Weinan Jia, Mengqi Huang, Nan Chen, Lei Zhang, and Zhendong Mao. 2025a. Dˆ 2it: Dynamic diffusion transformer for accurate image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12860–12870.

Yuhang Jia, Yang Chen, Jinghua Zhao, Shiwan Zhao, Wenjia Zeng, Yong Chen, and Yong Qin. 2025b. Audioeditor: A training-free diffusion-based audio editing framework. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. 2021. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2023. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and 1 others. 2023. Voicebox: text-guided multilingual universal speech generation at scale. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 14005–14034.

Peike Patrick Li, Boyu Chen, Yao Yao, Yikai Wang, Allen Wang, and Alex Wang. 2024. Jen-1: Text-guided universal music generation with omnidirectional diffusion models. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 762–769. IEEE.

Jinhua Liang, Yuanzhe Chen, Yi Yuan, Dongya Jia, Xiaobin Zhuang, Zhuo Chen, Yuping Wang, and Yuxuan Wang. 2025. Audiomorphix: Training-free audio editing with diffusion probabilistic models. *arXiv preprint arXiv:2505.16076*.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. 2023a. Audioldm: text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 21450–21474.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. 2024a. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. 2024b. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883.

Xingchao Liu, Chengyue Gong, and Qiang Liu. 2023b. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. Dpm-solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 5775–5787.

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

Yiwen Lu, Zhen Ye, Wei Xue, Xu Tan, Qifeng Liu, and Yike Guo. 2024. Comosvc: Consistency model-based singing voice conversion. In *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 184–188. IEEE.

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

Yingfeng Luo, Ziqiang Xu, Yuxuan Ouyang, Murun Yang, Dingyang Lin, Kaiyan Chang, Tong Zheng, Bei Li, Peinan Feng, Quan Du, Tong Xiao, and Jingbo Zhu. 2025. Beyond english: Toward inclusive and scalable multilingual machine translation with llms. *CoRR*, abs/2511.07003.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2024. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. 2024. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 564–572.

Hila Manor and Tomer Michaeli. 2024. Zero-shot unsupervised and text-based audio editing using ddpm inversion. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34603–34629.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. 2024. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345. IEEE.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. 2025. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2063–2072. IEEE.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047.

L.F. Shampine P. Bogacki. 1989. A pair of runge-kutta formulas. *Applied Mathematics Letters*, 2(4):321–325.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. 2024. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *9th*

International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

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

Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, and 1 others. 2023. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*.

Bram Wallace, Akash Gokul, and Nikhil Naik. 2023. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, and Sheng Zhao. 2023. Audit: audio editing by following instructions with latent diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 71340–71357.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Token pruning in multimodal large language models: Are we solving the right problem? *arXiv preprint arXiv:2502.11501*.

Manjie Xu, Chenxing Li, Duzhen Zhang, Dan Su, Wei Liang, and Dong Yu. 2024. Prompt-guided precise audio editing with diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 55126–55143.

Jinlong Xue, Yayue Deng, Yingming Gao, and Ya Li. 2024. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zhen Ye, Wei Xue, Xu Tan, Jie Chen, Qifeng Liu, and Yike Guo. 2023. Comospeech: One-step speech and singing voice synthesis via consistency model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1831–1839.

Qinsheng Zhang and Yongxin Chen. 2023. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yixiao Zhang, Yukara Ikemiya, Gus Xia, Naoki Murata, Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. 2024. Musicmagus: Zero-shot text-to-music editing via diffusion models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 7805–7813. International Joint Conferences on Artificial Intelligence Organization. AI, Arts & Creativity.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. 2023. Unipc: a unified predictor-corrector framework for fast sampling of diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 49842–49869.

Appendix

A Experimental Setup Details

A.1 Evaluation Metrics

We employ comprehensive metrics across three dimensions: **(1) Generative Quality:** We use **FAD** (Fréchet Audio Distance) to measure the distribution overlap between generated and real audio, and **IS** (Inception Score) to assess the clarity and diversity of the samples. **(2) Editing Precision: CS** (CLAP Score)[3] (Elizalde et al., 2023a,b) quantifies the semantic alignment between edited audio and text prompts, while **KL** (Kullback-Leibler Divergence) measures information loss to assess original content preservation. The calculation of KL, FAD, and IS follows the standardized evaluation pipeline provided by the AudioLDM-Eval package[4] (Liu et al., 2024b, 2023a) to ensure reproducible results. **(3) Efficiency:** The primary metric is **NFE**, standing for the number of function evaluations, which is used to quantify inference speed and computational savings.

A.2 Implementation Details

Different backbones were evaluated on different GPUs according to the hardware available during our experiments. Specifically, AudioLDM2 and Auffusion were run on a single NVIDIA GeForce RTX 3090 GPU, while Tango2 was run on a single NVIDIA A100 (80GB) GPU. For all experiments, we maintain a consistent set of core hyperparameters to demonstrate the robustness of AdaTE. Specifically, the maximum bypassing budget is set to $B_{\max} = 10$, the balancing factor to $\alpha = 1$, and the mutation threshold to $\tau = 0$. The scaling factor γ is fixed at 0.5 to provide a safety margin for linear extrapolation. Due to the inherent variations in output scales and trajectory dynamics across different models, the information tolerance ϕ is specifically adjusted for each backbone: we set $\phi = 5$ for AudioLDM2, $\phi = 2$ for Auffusion, and $\phi = 3$ for Tango2.

[3] https://github.com/microsoft/CLAP

[4] https://github.com/haoheliu/audioldm_eval

Method	NFE	Speedup	Adding				Removing				Replacing			
			CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑
DDIM Inv.	50	1.0×	39.63	2.186	3.385	3.683	43.94	2.027	**2.853**	**3.683**	41.79	2.357	**2.847**	**3.064**
DDIM Inv.	15	3.3×	40.75	**2.095**	3.784	3.659	43.71	**1.859**	3.270	3.305	**42.41**	**2.247**	3.105	2.850
+ RK45	14	3.6×	9.659	3.485	27.188	1.167	10.28	4.272	30.551	1.135	9.812	3.506	28.466	1.119
+ DPM++	15	3.3×	20.05	2.541	8.572	2.852	21.02	2.831	8.817	2.705	18.570	2.921	9.302	2.385
+ UniPC	15	3.3×	38.25	2.269	4.958	3.177	41.71	2.184	4.498	3.112	40.220	2.434	4.454	2.613
+ AdaTE	**13**	**3.9**×	**40.88**	2.214	**3.104**	**3.704**	**44.65**	2.109	3.308	3.466	41.50	2.486	3.060	2.978

Table 5: **Full Results on Comparison with Advanced ODE Solvers.** We evaluate AdaTE against state-of-the-art ODE solvers on the AudioLDM2 backbone across three tasks (Adding, Removing, and Replacing). NFE represents the average number of function evaluations across the test set, while Speedup is measured relative to the standard DDIM (50-step) setting. **Bold** and underlined values indicate the best and second-best performance.

Backbone	Method	NFE	Add ↑	Rem ↑	Rep ↑
AudioLDM2	DDIM	50	**3.96**	3.86	**3.83**
	DDIM	15	3.35	3.42	3.38
	+ AdaTE	12.8	3.94	**3.87**	3.80
Auffusion	AudioEditor	25	**4.25**	4.18	**4.12**
	AudioEditor	15	4.12	4.05	3.98
	+ AdaTE	13.6	4.22	**4.23**	4.07

Table 6: **Perceptual Quality (MOS).** Subjective evaluations conducted by 25 participants across Adding, Removing, and Replacing tasks.

Backbone	Method	NFE	RTF ↓
AudioLDM2	DDIM	50	1.167
	DDIM	15	0.385
	+ AdaTE	12.8	0.324
Auffusion	AudioEditor	25	2.332
	AudioEditor	15	1.517
	+ AdaTE	13.6	1.324
Tango2	AudioMorphix	50	2.958
	AudioMorphix	15	0.925
	+ AdaTE	15.7	0.964

Table 7: **Inference Efficiency (RTF).** RTF is measured under the hardware available for each backbone in our experiments: NVIDIA GeForce RTX 3090 for AudioLDM2 and Auffusion, and NVIDIA A100 (80GB) for Tango2. Absolute values are mainly intended for within-backbone comparison under the same hardware setting.

B Extended Experimental Results

In this section, we provide the full breakdown of experimental data that were summarized in the main text due to space constraints.

B.1 Detailed Results for Advanced Solvers

Table 5 presents the per-task results (Adding, Removing, and Replacing) comparing AdaTE with other advanced solvers at reduced NFE settings. The data indicates that under low-NFE conditions (NFE $\approx$ 15), the performance of standard higher-order solvers varies. While UniPC provides relatively stable results, RK45 and DPM++ show a noticeable decline in content consistency (CS) and audio fidelity (FAD) compared to the original DDIM inversion. In contrast, AdaTE achieves a 3.9× speedup with 13 NFEs while maintaining metrics comparable to, and in some cases exceeding (e.g., FAD in the Adding task), the 50-step DDIM baseline. These results suggest that for accelerated audio editing, selecting an appropriate sampling trajectory is as crucial as the choice of numerical solver.

B.2 Detailed Results for Subjective Evaluation

Table 6 presents the detailed Mean Opinion Score (MOS) results across the three core editing tasks: *Adding*, *Removing*, and *Replacing*. As shown, the subjective ratings exhibit a consistent trend with the objective metrics.

B.3 Detailed Results for Hardware Efficiency

Table 7 reports the Real-Time Factor (RTF) across different backbones. According to the hardware availability during our experiments, AudioLDM2 and Auffusion were run on a single NVIDIA GeForce RTX 3090 GPU, while Tango2 was run on a single NVIDIA A100 (80GB) GPU. Consequently, absolute RTF values are primarily intended for within-backbone comparisons to demonstrate that our NFE reduction consistently translates into hardware-level speedup.

Method	NFE	Speedup	Adding				Removing				Replacing			
			CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑	CS↑	KL↓	FAD↓	IS↑
Backbone: AudioLDM2 (Music Editing)														
DDIM Inv.	50	1.0×	46.53	0.586	1.442	**1.429**	43.84	0.667	3.495	1.343	40.57	0.870	2.810	**1.418**
DDIM Inv.	15	3.3×	46.89	**0.575**	**1.230**	1.380	**44.40**	**0.670**	**3.373**	1.322	40.52	0.844	**2.505**	1.345
+ AdaTE	10.3	4.9×	**46.94**	0.586	1.446	1.418	44.34	0.682	3.474	**1.345**	**40.74**	**0.840**	2.868	1.413
Backbone: Auffusion (Music Editing)														
AudioEditor	25	1.0×	41.24	1.043	4.286	1.605	38.14	1.354	6.236	**1.624**	42.18	1.231	5.852	**1.872**
AudioEditor	15	1.7×	41.50	1.028	4.710	**1.632**	37.17	**1.263**	6.468	1.511	41.32	1.190	5.664	1.686
+ AdaTE	13.2	1.9×	**43.41**	**0.987**	**3.617**	1.591	**39.67**	1.307	**5.444**	1.600	**43.47**	**1.173**	**5.063**	1.706

Table 8: **Full Results on Music Editing.** This table provides a comprehensive per-task breakdown of generative quality and editing precision for music editing. **Bold** and underlined values indicate the best and second-best performance within each backbone.

B.4 Detailed Results on Music Editing (BabySlakh)

Table 8 provides a comprehensive breakdown of per-task metrics on the music editing benchmark. Specifically, AdaTE achieves a balanced performance profile for AudioLDM2 under low-NFE regimes, while consistently outperforming the AudioEditor baseline across most metrics when applied to Auffusion. These results underscore the robustness and versatility of AdaTE across diverse audio domains.

C Algorithm

Algorithm 1 details the execution flow of AdaTE. Each neural evaluation acts as a probing step that measures local trajectory variation through C_t, A_t, and I_t. When the estimated mutation level remains below the threshold τ, the algorithm allocates a bypass budget B according to I_t, so that smaller I_t leads to a larger skipping window. Otherwise, it falls back to standard neural evaluation. This design concentrates model calls on structurally complex regions while using short extrapolation windows in stable segments.

Algorithm 1: AdaTE: Information-Aware Adaptive Execution

Input : Source latent z_T, Scheduler S, Diffusion Model M, Information tolerance ϕ, Mutation threshold τ, Scaling factor γ, Max budget $B_{\max}$, Step size Δt, Steps $\{t_i\}_{i=1}^{N}$

Output: Edited latent z_0

Initialize $z \leftarrow z_T$, $B \leftarrow 0$, $v \leftarrow$ None, $v_{prev} \leftarrow$ None, $\mathcal{C}_{prev} \leftarrow 0$;

for $i \leftarrow 1$ **to** N **do**

  if $B > 0$ **then**

    // Bypassing Stage: Linear Extrapolation

    $\hat{v} \leftarrow 2v - v_{prev}$; // Extrapolation

    $z \leftarrow \mathrm{S}(z, t_i, \hat{v})$; // Diffusion scheduler update

    $v_{prev} \leftarrow v, \quad v \leftarrow \hat{v}$; // Maintain state

    $B \leftarrow B - 1$;

  else

    // Probing Stage: Neural Evaluation

    $v_{prev} \leftarrow v$;

    $z_{prev} \leftarrow z$;

    $v \leftarrow \mathrm{M}(z, t_i, \cdot)$; // Neural model evaluation

    $z \leftarrow \mathrm{S}(z, t_i, v)$;

    $\Delta z_t \leftarrow z - z_{prev}$;

    // Adaptive Budget Allocation

    if $v_{prev} \neq$ *None* **then**

      $\mathcal{C}_t \leftarrow \|v - v_{prev}\|_2 / \Delta t$; // Curvature

      $\mathcal{I}_t \leftarrow f(\mathcal{C}_t, \Delta z_t)$; // Info-Gain

      $\mathcal{A}_t \leftarrow (\mathcal{C}_t - \mathcal{C}_{prev}) / \Delta t$; // Acceleration

      if $\mathcal{A}_t < \tau$ **then**

        $B \leftarrow \min(B_{max}, \lfloor \gamma \cdot \phi / \mathcal{I}_t \rfloor)$;

      else

        $B \leftarrow 0$;

      end

      $\mathcal{C}_{prev} \leftarrow \mathcal{C}_t$;

    end

  end

end

return z;