WaveEx: Accelerating Flow Matching-based Speech Generation via Wavelet-guided Extrapolation

Xiaoqian Liu[1,2*], Xiyan Gui[2,3*], Zhengkun Ge[1], Yuan Ge[1], Chang Zou[2], Jiacheng Liu[2], Zhikang Niu[2], Qixi Zheng[2], Chen Xu[4], Xie Chen[2], Tong Xiao[1,5], Jingbo Zhu[1,5†], Linfeng Zhang[2†]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Shanghai Jiao Tong University , Shanghai, China
[3]Huazhong University of Science and Technology, Wuhan, China
[4]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[5]NiuTrans Research, Shenyang, China
liuxiaoqian0319@outlook.com, zhujingbo@mail.neu.edu.cn, zhanglinfeng@sjtu.edu.cn

Abstract

Flow matching-based generative models offer a principled approach to modeling continuous-time dynamics in speech generation. However, inference is often computationally expensive due to repeated neural network evaluations required by ODE solvers. We propose *WaveEx*, a training-free and plug-in acceleration framework which replaces portions of ODE integration with *wave*let-guided *ex*trapolation. By leveraging the multi-scale structure of latent trajectories, WaveEx predicts future states directly in the frequency domain without additional model evaluations or architectural changes. WaveEx consistently accelerates inference across diverse speech generation tasks. The gains are especially pronounced in tasks like speech synthesis (up to 5.73× speedup) and music generation (2.75×), where flow matching plays a central role in alignment modeling and dense ODE integration. Even in tasks with simpler input-output mappings such as speech enhancement (4.55×) and voice conversion (2.75×), WaveEx still achieves notable acceleration, demonstrating the robustness and generalizability of the approach. These results highlight wavelet-guided extrapolation as a lightweight and broadly applicable alternative to full ODE solving for flow matching-based speech generation.

Introduction

Speech generation enables machines to produce human-like audio for applications like voice assistants and music synthesis, requiring both accurate linguistic content and natural waveforms. To meet these demands, continuous-time neural models formulate generation as solving an ordinary differential equation (ODE), allowing flexible modeling of variable-length and continuous signals. In particular, flow matching (FM) models avoid likelihood-based objectives and directly learn continuous dynamics, demonstrating success across diverse tasks, including text-to-speech (TTS) (Le et al. 2023; Guo et al. 2024; Mehta et al. 2024), music generation (Ning et al. 2025), voice conversion (VC) (Du et al. 2024; Liu 2024), and speech enhancement (SE) (Lee et al. 2025).

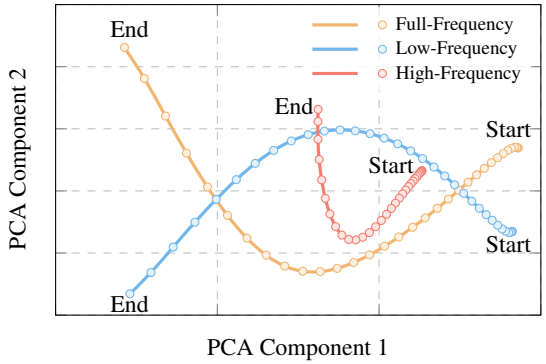


Figure 1: Principal Component Analysis (PCA) applied to latent trajectories. The plot shows projections onto the top two components. Full-frequency (orange) shows smooth global evolution; low-frequency (blue) captures coherent trends; high-frequency (red) remains spatially localized.

Despite their success, FM-based models incur high inference cost (Zhu et al. 2025), as each output frame requires solving an ODE with multiple full-model evaluations. Recent acceleration efforts (Zhuang et al. 2021; Salimans and Ho 2022; Dupont, Doucet, and Teh 2019; Zheng et al. 2025) require extra training or lack generalizability. In contrast, extrapolation-based skipping strategies (Liu et al. 2025) used in computer vision offer a promising alternative by reducing model evaluations during inference without modifying the model. Their applicability to speech with complex time-frequency dynamics remains yet underexplored.

Our work is motivated by two observations in Figure 1:

(1) Global Smoothness: Latent trajectories exhibit global smoothness and continuity. This suggests predictable, continuous latent dynamics, which strongly supports the use of extrapolation-based acceleration.

(2) Frequency-dependent Structure: Different bands display distinct latent behaviors: low-frequency components

*Authors contributed equally as EPIC research assistants.
†Corresponding authors.

traverse long and coherent paths, capturing global trends, whereas high frequencies stay within localized regions. This structure implies that modeling the entire trajectory is suboptimal as it may reduce extrapolation accuracy.

These insights motivate *WaveEx*, a wavelet-guided extrapolating strategy, where low- and high-frequency bands are extrapolated separately, respecting their inherent dynamics.

ODE solvers require multiple forward passes per step, with per-step complexity $\mathcal{O}(L \cdot d^2)$ for a Transformer of L layers and hidden size d, often repeated dozens or hundreds of times. In contrast, WaveEx relies on lightweight numerical operations with $\mathcal{O}(T \log T)$ complexity for sequence length T, entirely independent of model size. This substantial gap in computational complexity explains the high cost of ODE-based inference and motivates our effort to replace selected model calls with efficient numerical surrogates. WaveEx replaces selected model evaluations with a three-stage numerical procedure:

- **Wavelet decomposition**: Separates the latent feature into coarse low- and fine high-frequency components.
- **Trajectory extrapolation**: Independently extrapolates both low- and high-frequency signals using Taylor expansion, taking advantage of their predictable structures.
- **Wavelet reconstruction**: Recombines the extrapolated components to produce the predicted latent state.

We evaluate WaveEx on four speech generation tasks using five public models: F5-TTS (Chen et al. 2025) and E2-TTS (Eskimez et al. 2024) for TTS, DiffRhythm (Ning et al. 2025) for music generation, Seed-VC (Liu 2024) for voice conversion, and FlowSE (Lee et al. 2025) for speech enhancement. WaveEx consistently accelerates inference while maintaining high quality, demonstrating its generality and effectiveness. Our main contributions can be summarized as follows:

- We theoretically characterize the structure of FM-based latent trajectories and identify the spectral compressibility of Mel-spectrograms, laying a principled groundwork for efficient acceleration in speech generation.
- We propose WaveEx, a training-free, model-agnostic extrapolation framework that substitutes a portion of the ODE model evaluations with wavelet-guided predictions, reducing expensive neural network calls.
- We validate that WaveEx achieves up to 5.73× inference acceleration while maintaining comparable synthesis quality, demonstrating its effectiveness across multiple speech generation models and tasks.
- We show that frequency-aware extrapolation enabled by wavelet decomposition yields more stable and controllable trajectory prediction than naive skipping.

Related Work

Flow Matching-based Speech Generation

Flow matching has become a prominent method for speech generation by modeling data as continuous-time trajectories (Lee et al. 2024; Eskimez et al. 2024; Le et al. 2023). FM-based models provide fine-grained control and high-quality synthesis for various tasks, including text-to-speech (TTS) (Shen et al. 2024, 2018; Eskimez et al. 2024), voice conversion (VC) (Du et al. 2024; Liu 2024), and speech enhancement (SE) (Lee et al. 2025). The generation process involves solving an ODE, typically via numerical integration that requires multiple evaluations of a large neural network. This inference process incurs substantial computational cost.

Acceleration for Speech Generative Models

Existing acceleration efforts can be broadly categorized into two types. First, model compression and architectural simplifications (Zhu et al. 2025) reduce the per-step cost by designing lighter models, but often require retraining and may trade off performance. Second, adaptive integration and progressive distillation (Salimans and Ho 2022; Zhuang et al. 2021) reduce the number of ODE steps by learning a shortcut trajectory. Some recent methods, such as EPSS (Zheng et al. 2025), further attempt to prune redundant steps by inspecting the sampling trajectory and selecting a subset of time steps. While effective, these approaches often rely on empirical analysis, retraining, access to gradient information, or architecture-specific modifications, limiting their generality across different speech generation tasks.

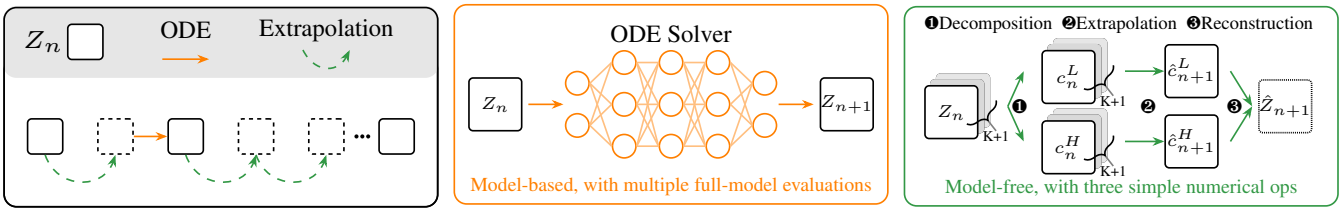
Numerical Extrapolation in Generative Modeling

Recent studies in computer vision domain have revealed that diffusion-based model features across adjacent timesteps exhibit high similarity or smooth predictability. Accordingly, various feature caching techniques have been explored, including reusing cached features from previous steps (Selvaraju et al. 2024; Zou et al. 2025, 2024; Chen et al. 2024) and extrapolating future states based on historical token-wise feature trajectories (Liu et al. 2025). These approaches aim to predict future hidden states within model calls, thereby skipping redundant computations. However, they have mainly focused on computer vision tasks and rarely consider the structure or dynamics of speech signals. Furthermore, such methods typically operate directly in the latent space, which limits their effectiveness on temporally sensitive modalities like speech.

Wavelet Methods in Speech Processing

Wavelet transforms (Daubechies 1988; Cohen, Daubechies, and Feauveau 1992; Daubechies 1992) have a long history in speech analysis, where they are used for denoising, compression, and multi-resolution feature extraction. Their ability to separate signals into multi-scale components makes them particularly suitable for modeling the time-frequency nature of speech. However, wavelet methods have not been integrated into the inference process of generative models, and their potential for trajectory extrapolation in FM-based generation remains largely unexplored.

Our work is the first to exploit wavelet decomposition to guide extrapolation in the latent dynamics of speech generation. In contrast to prior works, WaveEx provides a training-free, model-agnostic acceleration framework. It leverages both wavelet-based signal decomposition and numerical extrapolation, offering a principled and lightweight solution to the inference bottleneck in FM-based speech generation.



(a) Overall Inference Pipeline

(b) ODE Step, Slow and Complex

(c) Extrapolation Step, Fast and Simple

Figure 2: Illustration of the WaveEx framework. **(a):** Inference pipeline showing latent states $\mathbf{z}_n$, where orange solid arrows represent standard ODE-based updates, and green dashed arrows indicate WaveEx extrapolation steps that bypass full model evaluation. **(b):** Standard ODE solvers require multiple expensive model calls. **(c):** WaveEx predicts the next state via: (1) wavelet decomposition of recent states, (2) band-wise extrapolation, and (3) reconstruction via inverse transform.

Method

Preliminary: Flow Matching-based Generation

Let $\mathbf{z}(t) \in \mathbb{R}^D$ be the latent state at time t, and f be a neural network. The generative process is governed by the following ODE function:

$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t), \quad \mathbf{z}(0) = \mathbf{z}_0. \tag{1}$$

During inference, numerical solvers compute the latent trajectory $\mathbf{z}(t)$ by iteratively updating $\mathbf{z}_{n+1}$ from $\mathbf{z}_n$, typically requiring multiple neural network evaluations per step. To ensure stability and accuracy, solvers adopt small step sizes and densely sample the trajectory, causing numerous function evaluations (NFE) that dominate inference time.

WaveEx: Wavelet-Guided Extrapolation

As shown in Figure 2(a), WaveEx accelerates FM-based generation by alternating standard solver steps with extrapolation, forming a fast yet accurate hybrid trajectory. Figure 2(b) shows a standard ODE step, which involves full forward passes through the neural network. In contrast, Figure 2(c) illustrates an extrapolation step, which predicts future latent states based solely on previously observed states, without invoking the model. We now detail the design and implementation of the extrapolation step.

Extrapolation Procedure

Step 1: Wavelet Decomposition To perform wavelet extrapolation, we first collect a sliding window of the past $K+1$ latent states at each inference step:

$$\mathcal{H}_n = \{\mathbf{z}_{n-K}, \ldots, \mathbf{z}_n\} \in \mathbb{R}^{(K+1)\times D}. \tag{2}$$

This latent history cache provides the temporal context necessary for multi-scale frequency analysis and derivative estimation. We then apply a discrete wavelet transform (DWT) along the temporal axis to decompose the latent sequence into coarse and fine components:

$$\mathcal{H}_n \xrightarrow{\text{DWT}} \left\{\mathbf{c}^L_{n-K:n}, \mathbf{c}^H_{n-K:n}\right\}, \tag{3}$$

where $\mathbf{c}^L, \mathbf{c}^H \in \mathbb{R}^{(K+1)\times D}$ represent the low-frequency and high-frequency coefficient sequences, respectively.

The low-frequency component $\mathbf{c}^L$ captures the smooth, global evolution of the trajectory, while the high-frequency component $\mathbf{c}^H$ retains local modulations and fine-grained dynamics. This decomposition allows us to model the structured variation in latent space across multiple scales, aligning with the hierarchical nature of speech signals.

Step 2: Multi-scale Extrapolation We perform Taylor extrapolation independently on the low-frequency and high-frequency components obtained from the decomposition step. For each component $\mathbf{c}^{(\cdot)} \in \{\mathbf{c}^L, \mathbf{c}^H\}$, we estimate temporal derivatives up to order p via backward finite differences computed from the past $K+1$ frames:

$$\left(\mathbf{c}_n^{(\cdot)}\right)^{(j)} = \frac{1}{h^j} \sum_{i=0}^{j} w_i^{(j)}\, \mathbf{c}_{n-i}^{(\cdot)}, \quad j = 1, \ldots, p, \tag{4}$$

where $w_i^{(j)}$ are finite difference weights, and h is the time step size. We then construct the extrapolated coefficients at the next time step $n+1$ via a truncated Taylor series:

$$\hat{\mathbf{c}}_{n+1}^{(\cdot)} = \sum_{j=0}^{p} \frac{h^j}{j!} \left(\mathbf{c}_n^{(\cdot)}\right)^{(j)}. \tag{5}$$

By extrapolating $\mathbf{c}^L$ and $\mathbf{c}^H$ separately, the method can emphasize smooth progression in the low-frequency branch while selectively attenuating noise or instability in the high branch. This flexibility enhances robustness when predicting future latent states across different temporal scales.

Step 3: Wavelet Reconstruction The next latent state $\hat{\mathbf{z}}_{n+1}$ is reconstructed by applying IDWT, the inverse discrete wavelet transform, to the separately extrapolated low-frequency and high-frequency coefficients:

$$\hat{\mathbf{z}}_{n+1} = \text{IDWT}\left(\hat{\mathbf{c}}^L_{n+1}, \hat{\mathbf{c}}^H_{n+1}\right). \tag{6}$$

The IDWT combines the multi-scale frequency components into a unified latent vector, restoring the original temporal and spectral structure of the signal. This reconstruction preserves essential details captured at different frequency bands while maintaining overall signal continuity.

The reconstructed latent state $\hat{\mathbf{z}}_{n+1}$ serves as a model-free prediction for the next step, effectively bypassing a costly neural network evaluation. This predicted latent is then used either as input for subsequent ODE solver steps or as the final representation for speech generation modules.

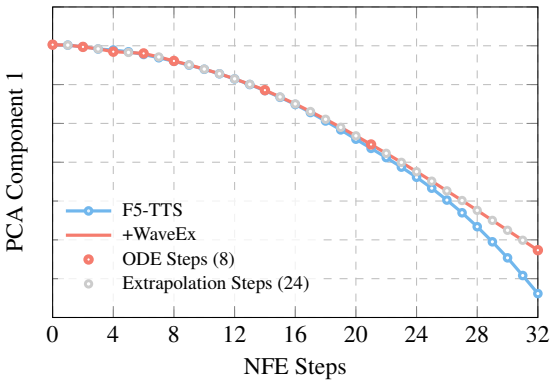


Figure 3: Principal Component Analysis (PCA) applied to latent trajectories to visualize sampling distributions under the +WaveEx strategy. The blue curve shows the full 32-NFE trajectory of F5-TTS. The red curve represents +WaveEx, where red dots denote ODE solver steps and gray dots denote extrapolation steps. Unlike naive step reduction, +WaveEx skips selected ODE solving via extrapolation.

Extrapolation Scheduling In speech generation, we observe that latent representations evolve rapidly in early steps due to phonetic transitions and speaker-specific information, whereas in later stages, the trajectory becomes smoother and more predictable. To balance modeling fidelity and inference efficiency, we adopt a two-phase scheduling strategy:

- **Early Phase** ($t \in [0, 0.25]$): We interleave extrapolation with neural ODE evaluations. This hybrid scheme ensures robust handling of high-curvature regions where extrapolation alone may incur large errors.
- **Late Phase** ($t > 0.25$): As the trajectory enters a smoother regime, we reduce the number of model evaluations by performing only a few ODE steps per window, using extrapolation to infer the remaining steps. This exploits the increased regularity to accelerate generation without significant loss in quality.

Figure 3 illustrates how this two-phase schedule applied to a WaveEx-accelerated model with 32 NFE steps. This scheduling mechanism is lightweight and model-agnostic, and aligns with the dynamic nature of speech synthesis: it adaptively allocates computational resources based on the expected complexity of the latent dynamics, offering a principled trade-off between accuracy and speed.

Experiments

Experimental Setup

Tasks, Models and Evaluation Metrics We evaluate our method across four diverse speech generation tasks. All models are used with original configurations and pre-trained checkpoints to ensure fair and reproducible comparisons. We use task-specific evaluation metrics and standardized test sets to ensure meaningful and consistent comparisons.

- **Text-to-Speech**: Due to their different model backbone, we choose F5-TTS (Chen et al. 2025) and E2-TTS (Eskimez et al. 2024) for evaluation. The E2-TTS model used in our experiments corresponds to a reproduction provided by the official F5-TTS codebase. Evaluations are conducted on the LibriSpeech-PC (Meister et al. 2023) test-clean set, following the same configuration as the F5-TTS repository. We report Word Error Rate (WER), Speaker Similarity (SIM), and Mean Opinion Scores (MOS) come from a subjective test with 50 participants on a 1–10 scale to reflect generation quality.
- **Music Generation**: We evaluate on DiffRhythm (Ning et al. 2025), a diffusion-based framework for music generation. As no standard test set exists, we follow the official protocol and construct a 100-utterance evaluation set. We evaluate using SongEval (Yao et al. 2025), reporting Coherence (Coh), Clarity of Song Structure (CSS), and Naturalness of Vocal Breathing and Phrasing (NVBP).
- **Voice Conversion**: Seed-VC (Liu 2024) is a zero-shot FM-based VC model. We evaluate on the LibriTTS (Zen et al. 2019) dataset using the official evaluation scripts provided in the Seed-VC codebase, reporting Character Error Rate (CER), Speaker Embedding Cosine Similarity (SECS), and Overall Quality (OVRL) (Dubey et al. 2023) to jointly assess perceptual quality.
- **Speech Enhancement**: FlowSE (Lee et al. 2025) is an FM-based model that maps noisy to clean speech in one pass. Evaluation on DNS-Challenge 2023 (Dubey et al. 2023) uses Signal Distortion (SIG), Background Intrusiveness (BAK), and Overall Quality (OVRL) to assess clarity, noise suppression, and perceptual quality.

WaveEx Parameters Unless otherwise specified, we use the Symlet-6 (sym6) wavelet with single-level decomposition for its near-symmetry and good time-frequency localization, which help preserve the structure of both low- and high-frequency components during extrapolation. First-order Taylor extrapolation is applied independently to both the low-frequency and high-frequency coefficients obtained from the decomposition. Following a two-phase scheduling strategy, we perform ODE evaluations at steps $[0, 2, 4, 6, 8]$ during the early phase. In the late phase, ODE steps are applied at task-specific positions: step 14 for TTS, step 20 for VC, step 24 for SE, and step $[12, 19, 31]$ for music generation (these configurations also serve as the default for Analysis). All remaining steps are estimated via extrapolation.

Main Results

1. Text-to-Speech. We evaluate WaveEx on F5-TTS (DiT backbone) and E2-TTS (UNet backbone, reimplemented via F5-TTS repository). For F5-TTS, WaveEx reduces NFE from 32 to 6, achieving a 4.46× speedup in RTF with only a slight WER increase (2.132 → 2.175) and preserved SIM (0.672 → 0.668). Compared to EPSS, WaveEx yields comparable acceleration while delivering better WER and MOS. On E2-TTS, the vanilla 6-step decoding leads to severe quality degradation (WER: 20.141, SIM: 0.396), while both WaveEx and EPSS restore performance. Notably, WaveEx achieves the best SIM (0.703) and a favorable trade-off between speed (5.73×) and generation quality (WER: 3.323, MOS: 6.029), outperforming EPSS in SIM and MOS.

Task	Model	NFE	WER (%) ↓	SIM ↑	MOS ↑	RTF ↓	Speedup ↑
1. Text-to-Speech	F5-TTS	32	**2.132**	**0.672**	**7.322**	0.116	1.00×
	F5-TTS	6	4.323	0.639	4.321	0.028	3.87×
	+ EPSS	6	2.227	0.662	6.341	0.022	5.27×
	+ WaveEx (6+26)†	32	<u>2.175</u>	<u>0.668</u>	<u>6.868</u>	0.026	4.46×
	E2-TTS	32	**3.018**	**0.703**	**6.742**	0.321	1.00×
	E2-TTS	6	20.141	0.396	2.660	0.053	6.06×
	+ EPSS	6	3.834	0.643	5.843	0.046	6.98×
	+ WaveEx (6+26)†	32	<u>3.323</u>	<u>0.702</u>	<u>6.029</u>	0.056	5.73×
Task	**Model**	**NFE**	**Coh ↑**	**CSS ↑**	**NVBP ↑**	**RTF ↓**	**Speedup ↑**
2. Music Generation	DiffRhythm	32	**3.805**	<u>3.598</u>	<u>3.414</u>	0.044	1.00×
	DiffRhythm	8	3.316	3.069	3.065	0.014	3.14×
	+ WaveEx (8+24)†	32	<u>3.804</u>	**3.613**	**3.431**	0.016	2.75×
Task	**Model**	**NFE**	**CER ↓**	**SECS ↑**	**OVRL ↑**	**RTF ↓**	**Speedup ↑**
3. Voice Conversion	Seed-VC	25	**2.401**	**0.859**	<u>2.990</u>	0.187	1.00×
	Seed-VC	6	2.641	0.847	2.963	0.062	3.02×
	+ WaveEx (6+19)†	25	<u>2.338</u>	**0.859**	**3.040**	0.068	2.75×
Task	**Model**	**NFE**	**SIG ↑**	**BAK ↑**	**OVRL ↑**	**RTF ↓**	**Speedup ↑**
4. Speech Enhancement	FlowSE	32	**3.295**	**3.729**	**2.882**	0.250	1.00×
	FlowSE	6	3.247	3.582	2.774	0.049	5.10×
	+ WaveEx (6+26)†	32	<u>3.283</u>	<u>3.720</u>	<u>2.868</u>	0.055	4.55×

- † We denote hybrid settings of WaveEx as $(a + b)$, where a is the number of ODE solver steps and b is the number of extrapolation steps.

Table 1: Performance comparison across multiple speech generation tasks and models. WaveEx consistently reduces decoding cost while maintaining or improving generation quality. The real-time factor (RTF) is evaluated on a single NVIDIA A100 GPU. Best results are highlighted in **bold**, and second-best results are <u>underlined</u>.

2. Music Generation. We evaluate on DiffRhythm for full-length song generation. Directly reducing NFE from 32 to 8 leads to notable degradation in musical quality across all metrics (e.g., Coh: 3.805 → 3.316). In contrast, WaveEx successfully recovers performance: it preserves the original Coh (3.804 vs. 3.805), and even improves CSS and NVBP (3.598 → 3.613 and 3.414 → 3.431), while achieving a $2.75\times$ speedup in RTF. These results demonstrate that WaveEx enables fast decoding in music generation without compromising coherence or structural integrity.

3. Voice Conversion. For voice conversion task, we apply WaveEx to Seed-VC. Reducing NFE from 25 to 6 yields faster decoding with slight quality degradation. WaveEx further improves upon this low-NFE baseline, recovering intelligibility (CER: 2.338) and enhancing overall quality (OVRL: 3.040 vs. 2.990), while maintaining a $2.75\times$ decoding speedup. These results validate the effectiveness of WaveEx in accelerating VC without sacrificing speaker similarity or naturalness.

4. Speech Enhancement. We evaluate on FlowSE for denoising. Reducing NFE from 32 to 6 significantly speeds up inference (RTF: 0.250 → 0.049, $5.10\times$) at the cost of moderate quality drop. Incorporating WaveEx at the same NFE level recovers most of the lost performance (OVRL: 2.868 vs. 2.774) and closely matches the original full-step results (e.g., SIG: 3.283 vs. 3.295), while maintaining a $4.55\times$ acceleration and enabling efficient speech enhancement.

Overall Performance. Across text-to-speech, music generation, voice conversion, and speech enhancement, WaveEx consistently achieves substantial acceleration (2.75–$5.73\times$) by significantly reducing the number of function evaluations. Crucially, this speedup is attained without compromising generation quality, as evidenced by improvements or parity across task-specific metrics. These results highlight the versatility, quality preservation, and model-agnostic nature of WaveEx, demonstrating its effectiveness as a general-purpose, training-free acceleration framework for FM-based speech and audio generation.

Compared with Cache-based Methods

We further validate the effectiveness of WaveEx by comparing with recent cache-based acceleration methods including Fora (Selvaraju et al. 2024), TaylorSeer (Liu et al. 2025), and Toca (Zou et al. 2025), all integrated into F5-TTS model.

As shown in Table 2, FORA and TaylorSeer operate at the feature level, either by reusing past latent states (FORA) or by predicting future states using a first-order Taylor approximation (TaylorSeer). These methods reduce decoding cost by avoiding full ODE evaluations at every step. While achieving moderate improvements in RTF (0.067–0.076), the accuracy degrades notably as NFE is reduced.

ToCa performs token-wise caching and extrapolation, but

Model	NFE	WER (%) ↓	RTF ↓
F5-TTS (Baseline)	32+0	**2.132**	0.116
+FORA	11+21	2.382	<u>0.067</u>
+FORA	8+24	2.751	<u>0.067</u>
+TaylorSeer	11+21	2.486	0.076
+TaylorSeer	8+24	3.015	0.068
+ToCa	32+0	2.781	0.134
+WaveEx	6+26	<u>2.175</u>	**0.026**

- We denote NFE configuration as $a + b$, where a is the number of ODE solver steps and b is the number of extrapolation steps.

Table 2: Comparison with cache-based acceleration methods on librispeech-pc test-clean. All methods are integrated into the same F5-TTS model. Best results are highlighted in **bold**, and second-best results are <u>underlined</u>.

its finer granularity introduces non-negligible runtime overhead, leading to none speedup (RTF = 0.134) and suboptimal generation quality (WER = 2.781). Token-level caching may be less effective for continuous-time speech modeling.

In comparison, WaveEx achieves a WER score of 2.175 using only 6 steps, the lowest NFE among all methods, while also attaining the best RTF (0.026). Unlike other methods, WaveEx requires no model retraining and makes no assumptions about the cacheability of internal representations. Instead, it leverages the multi-scale smoothness of latent trajectories through wavelet-guided extrapolation, allowing for stable predictions with minimal inference cost.

These results suggest that while cache-based approaches can reduce decoding overhead, they are not tailored to the structured dynamics of speech latent spaces. In contrast, WaveEx offers a more principled and generalizable acceleration strategy by operating in the frequency domain.

What Order of Taylor Expansion?

Figure 4 illustrates trajectory predictions using Taylor expansions of different orders. The black solid line denotes the ground-truth trajectory generated by F5-TTS. At selected starting points (e.g., step 3, 7, 11, ...), indicated by vertical gray dashed lines, we apply wavelet decomposition to the observed segment and extrapolate the next 8 steps. Predicted trajectories are shown as red dashed lines for first-order and green dashed lines for second-order expansion, overlaid on the original trajectory for comparison.

Early-stage predictions are more challenging: extrapolated trajectories deviate noticeably from the ground truth due to residual curvature and transient dynamics, which are harder to approximate with limited history. First-order extrapolation yields more stable and accurate results, closely following the ground truth. In contrast, second-order predictions often overreact to local fluctuations due to unstable acceleration estimation. In the later stage, both methods perform similarly as the trajectory becomes smoother and nearly linear. In such cases, first-order extrapolation is sufficient to capture the trend while offering better robustness. Based on this analysis, we adopt the first-order Taylor expansion for our extrapolation.

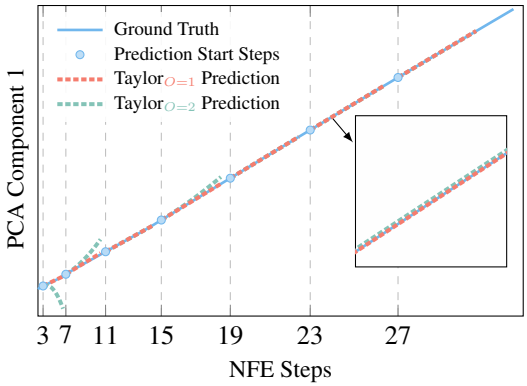


Figure 4: Trajectory prediction using first- and second-order Taylor extrapolation. The blue line is the ground truth; gray dashed lines mark starting points. Red and green dashed lines indicate first- and second-order predictions.

How to Deal with High-Frequency Components?

High-frequency components capture local variations in the latent trajectory. A natural question is whether to extrapolate, discard, or reuse them during reconstruction. To assess this, we use the music generation task, which involves complex rhythms and long-term structure, making it highly sensitive to high-frequency processing.

- **Zeroing**: Discard high-frequency content during reconstruction (i.e., $\hat{\mathbf{c}}^H_{n+1} = \mathbf{0}$).
- **Freezing**: Reuse the last observed value for reconstruction (i.e., $\hat{\mathbf{c}}^H_{n+1} = \mathbf{c}^H_n$).
- **Extrapolating (ours)**: Apply first-order Taylor extrapolation to the high-frequency branch.

We visualize the Coherence, Clarity, Musicality, and Naturalness score distributions using violin plots as shown in Figure 5. Among the high-frequency handling settings, Extrapolating shows a similar distribution like DiffRhythm (base model), with slightly heavier tails on both ends, suggesting that while it often matches DiffRhythm in quality, it also introduces more variability. Freezing shifts the distribution downward, with most scores falling between 2.0 and 2.5, reflecting its limited capacity to adaptively preserve high-frequency details. Zeroing performs the worst, with scores largely below 2.5, indicating that zeroing out high-frequency components severely harms generation.

In terms of efficiency, all three settings achieve fast inference with low RTF: Zeroing at 0.012, Freezing at 0.011, and Extrapolating at 0.016. The minor increase in RTF for Extrapolating is negligible in practice. Extrapolating consistently outperforms the others across all evaluation dimensions, demonstrating superior coherence, clarity, musicality, and naturalness. These results highlight that extrapolating high-frequency components not only preserves perceptual quality and structural fidelity, but also maintains near-equivalent inference speed.

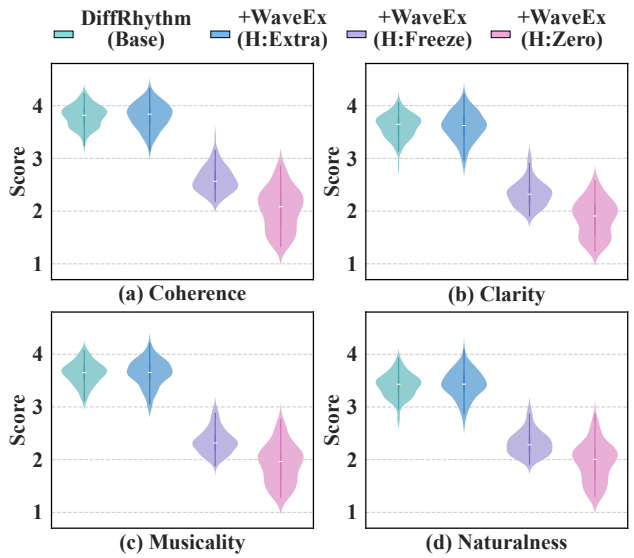


Figure 5: Evaluation across high-frequency strategies.

Schedule	Steps	WER (%) ↓	SIM ↑
Uniform	[0,6,11,16,21,26]	5.697	0.594
Quadratic	[0,1,4,9,16,25]	3.138	0.630
Two-phase	[0,2,4,6,8,14]	**2.175**	**0.664**

Table 3: Ablation on uniform, quadratic, and two-phase extrapolating schedules in F5-TTS + WaveEx.

When to Extrapolate Instead of Solving ODEs?

To investigate the impact of extrapolation scheduling, we compare three different strategies for selecting the temporal positions at which extrapolation is applied during inference. These include Uniform, Quadratic, and Two-phase schedules, as summarized in Table 3.

The Uniform schedule applies simplest extrapolation at evenly spaced steps (e.g., [0,6,11,...]), but performs the worst (WER 5.697, SIM 0.594), indicating that equal temporal spacing fails to capture critical variations that are concentrated in specific regions of the signal. The Quadratic schedule places more steps in the earlier part of the trajectory based on a quadratic progression (e.g., [0,1,4,...]), leading to improved results (WER 3.138, SIM 0.630) by focusing modeling capacity where signal variation is more intense.

Our proposed Two-phase schedule further improves performance (WER 2.175, SIM 0.664) by combining dense extrapolation in the early phase with coarser updates later. This staged design aligns with the observation that fine-grained early modeling helps establish local detail, while the latter phase benefits from broader structural continuation. The carefully designed two-phase scheme can substantially improve extrapolation effectiveness by adapting to the non-uniform complexity of speech trajectories.

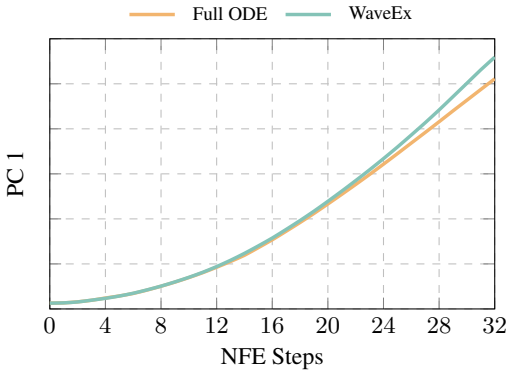


Figure 6: Latent trajectory divergence between full ODE decoding and WaveEx, quantified as the time-wise difference in projections onto the first principal component (PC1).

How Accurate is the Extrapolated Trajectory?

To assess the fidelity of WaveEx relative to the full ODE-based decoding, we compare the latent trajectories produced by both methods. As shown in Figure 6, we project the high-dimensional latent states onto their first principal component and visualize their evolution across inference steps. The orange curve represents the reference trajectory obtained by numerically solving the full ODE without any extrapolation, while the teal curve corresponds to the trajectory generated by WaveEx using scheduled wavelet-based extrapolation. The two curves exhibit a high degree of alignment throughout the entire process, indicating that WaveEx closely approximates the underlying dynamics of the ODE solver.

This result highlights the effectiveness of our method: despite significantly reducing the number of full model evaluations, WaveEx preserves the global structure and smooth evolution of the latent trajectory. The negligible discrepancy between the two curves further supports the reliability of WaveEx as a training-free, low-cost alternative for inference-time acceleration.

Conclusion

We propose WaveEx, a training-free acceleration framework for flow matching-based speech generation models. By leveraging wavelet-guided decomposition and Taylor-based extrapolation, WaveEx predicts future latent states in a frequency-aware manner, significantly reducing expensive NFE requirement during inference. We demonstrate the effectiveness of WaveEx across four representative speech generation tasks: text-to-speech, music generation, voice conversion, and speech enhancement. Experimental results show that WaveEx achieves 2.75–5.73× speedup in RTF, while maintaining generation quality across diverse metrics. Through extensive analysis, we show that WaveEx benefits from its ability to decouple and model frequency components separately, offering improved stability and robustness. WaveEx offers a lightweight, plug-and-play inference-time module, and we believe this work opens new possibilities for efficient generative modeling by exploiting structured priors in the latent trajectory space.

Acknowledgments

This work was supported in part by the Yunnan Fundamental Research Projects (No.202401BC070021), the National Science Foundation of China (Nos. 62276056 and U24A20334), the Yunnan Science and Technology Major Project (No. 202502AD080014), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009), CCF-Baidu Open Fund and Shanghai Science and Technology Program (Grant No. 25ZR1402278).

References

Chen, P.; Shen, M.; Ye, P.; Cao, J.; Tu, C.; Bouganis, C.-S.; Zhao, Y.; and Chen, T. 2024. Delta-DiT: A Training-Free Acceleration Method Tailored for Diffusion Transformers. *arXiv preprint arXiv:2406.01125*.

Chen, Y.; Niu, Z.; Ma, Z.; Deng, K.; Wang, C.; JianZhao, J.; Yu, K.; and Chen, X. 2025. F5-TTS: A Fairytaler that Fakes Fluent and Faithful Speech with Flow Matching. In *Proc. ACL*, 6255–6271. Vienna, Austria: Association for Computational Linguistics.

Cohen, A.; Daubechies, I.; and Feauveau, J.-C. 1992. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5): 485–560.

Daubechies, I. 1988. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7): 909–996.

Daubechies, I. 1992. *Ten lectures on wavelets*. SIAM.

Du, Z.; Chen, Q.; Zhang, S.; Hu, K.; Lu, H.; Yang, Y.; Hu, H.; Zheng, S.; Gu, Y.; Ma, Z.; et al. 2024. CosyVoice: A Scalable Multilingual Zero-shot Text-to-speech Synthesizer based on Supervised Semantic Tokens. *CoRR*.

Dubey, H.; Aazami, A.; Gopal, V.; Naderi, B.; Braun, S.; Cutler, R.; Gamper, H.; Golestaneh, M.; and Aichner, R. 2023. ICASSP 2023 Deep Noise Suppression Challenge. In *Proc. ICASSP*.

Dupont, E.; Doucet, A.; and Teh, Y. W. 2019. Augmented neural odes. *Advances in neural information processing systems*, 32.

Eskimez, S. E.; Wang, X.; Thakker, M.; Li, C.; Tsai, C.-H.; Xiao, Z.; Yang, H.; Zhu, Z.; Tang, M.; Tan, X.; et al. 2024. E2 TTS: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, 682–689. IEEE.

Guo, Y.; Du, C.; Ma, Z.; Chen, X.; and Yu, K. 2024. Voiceflow: Efficient text-to-speech with rectified flow matching. In *Proc. ICASSP*, 11121–11125. IEEE.

Le, M.; Vyas, A.; Shi, B.; Karrer, B.; Sari, L.; Moritz, R.; Williamson, M.; Manohar, V.; Adi, Y.; Mahadeokar, J.; et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36: 14005–14034.

Lee, K.; Kim, D. W.; Kim, J.; and Cho, J. 2024. DiTTo-TTS: Efficient and Scalable Zero-Shot Text-to-Speech with Diffusion Transformer. *CoRR*, abs/2406.11427.

Lee, S.; Cheong, S.; Han, S.; and Shin, J. W. 2025. FlowSE: Flow Matching-based Speech Enhancement. In *Proc. ICASSP*, 1–5. IEEE.

Liu, J.; Zou, C.; Lyu, Y.; Chen, J.; and Zhang, L. 2025. From reusing to forecasting: Accelerating diffusion models with taylorseers. *arXiv preprint arXiv:2503.06923*.

Liu, S. 2024. Zero-shot voice conversion with diffusion transformers. *arXiv preprint arXiv:2411.09943*.

Mehta, S.; Tu, R.; Beskow, J.; Székely, É.; and Henter, G. E. 2024. Matcha-TTS: A fast TTS architecture with conditional flow matching. In *Proc. ICASSP*, 11341–11345. IEEE.

Meister, A.; Novikov, M.; Karpov, N.; Bakhturina, E.; Lavrukhin, V.; and Ginsburg, B. 2023. Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, 1–7. IEEE.

Ning, Z.; Chen, H.; Jiang, Y.; Hao, C.; Ma, G.; Wang, S.; Yao, J.; and Xie, L. 2025. DiffRhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. *arXiv preprint arXiv:2503.01183*.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *ICLR*.

Selvaraju, P.; Ding, T.; Chen, T.; Zharkov, I.; and Liang, L. 2024. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*.

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al. 2018. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *Proc. ICASSP*, 4779–4783. IEEE.

Shen, K.; Ju, Z.; Tan, X.; Liu, E.; Leng, Y.; He, L.; Qin, T.; Zhao, S.; and Bian, J. 2024. NaturalSpeech 2: Latent Diffusion Models are Natural and Zero-Shot Speech and Singing Synthesizers. In *ICLR*.

Yao, J.; Ma, G.; Xue, H.; Chen, H.; Hao, C.; Jiang, Y.; Liu, H.; Yuan, R.; Xu, J.; Xue, W.; et al. 2025. SongEval: A Benchmark Dataset for Song Aesthetics Evaluation. *arXiv preprint arXiv:2505.10793*.

Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R. J.; Jia, Y.; Chen, Z.; and Wu, Y. 2019. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Proc. Interspeech*, 1526–1530.

Zheng, Q.; Chen, Y.; Niu, Z.; Ma, Z.; Wang, X.; Yu, K.; and Chen, X. 2025. Accelerating Flow-Matching-Based Text-to-Speech via Empirically Pruned Step Sampling. *arXiv preprint arXiv:2505.19931*.

Zhu, H.; Kang, W.; Yao, Z.; Guo, L.; Kuang, F.; Li, Z.; Zhuang, W.; Lin, L.; and Povey, D. 2025. ZipVoice: Fast and High-Quality Zero-Shot Text-to-Speech with Flow Matching. *arXiv preprint arXiv:2506.13053*.

Zhuang, J.; Dvornek, N. C.; Tatikonda, S.; and Duncan, J. S. 2021. Mali: A memory efficient and reverse accurate integrator for neural odes. *arXiv preprint arXiv:2102.04668*.

Zou, C.; Liu, X.; Liu, T.; Huang, S.; and Zhang, L. 2025. Accelerating Diffusion Transformers with Token-wise Feature Caching. In *ICLR*.

Zou, C.; Zhang, E.; Guo, R.; Xu, H.; He, C.; Hu, X.; and Zhang, L. 2024. Accelerating diffusion transformers with dual feature caching. *arXiv preprint arXiv:2412.18911*.