WaveGen: Truly End-to-End Waveform Generation via Internal Spectral Trajectory Learning

Sang-Hoon Lee[1]* **Ha-Yeong Choi**[2]*

[1]Department of Artificial Intelligence, Ajou University, Suwon, Korea
[2]Frontier AI Lab, KT Corp., Seoul, Korea

Abstract

This work proposes WaveGen, a fully end-to-end diffusion-based text-to-waveform framework based on Spectral Forcing, which jointly learns an internal spectral trajectory and the target waveform trajectory within a single model, enabling direct end-to-end text-to-waveform generation. Unlike recent TTS pipelines, WaveGen does not rely on separately trained or externally decoded intermediate representations, such as neural audio codecs, Mel-spectrogram vocoders, or VAE-based latent models. In addition, WaveGen performs internal text-speech alignment within a single model, eliminating the need for external alignment modules such as duration predictors. To preserve a truly end-to-end formulation, our framework further avoids dependence on self-supervised representations, such as BERT-based text embeddings and Wav2Vec 2.0-based semantic representations, as well as semantic distillation methods for training optimization. To achieve this, we carefully design Spectral Forcing, a model architecture with disentangled diffusion heads, and a training objective based on a multi-scale spectral v-loss. We further validate the scalability of end-to-end TTS models by scaling the model size from 0.1B to 0.6B parameters. Using a single architecture, open-source data, and one-stage training, without external modules such as duration predictor or REPA, WaveGen demonstrates fully end-to-end waveform generation achieving state-of-the-art performance on the Emilia training dataset. In particular, WaveGen achieves a WER of 1.47 and a SIM of 0.75 on the Seed-en benchmark. Source code and checkpoints are available at https://github.com/sh-lee-prml/WaveGen.

1 Introduction

Recent progress in language models and generative models has substantially advanced text-to-speech (TTS) synthesis. Large language model (LLM)-based TTS systems [58, 22] have replaced conventional acoustic representations with neural audio codecs, enabling speech generation through discrete token modeling and in-context voice prompting. In parallel, diffusion and conditional flow matching (CFM)-based TTS models [33, 5] have improved the quality and flexibility of Mel-spectrogram generation through masked infilling and zero-shot voice conditioning. More recently, latent diffusion approaches [75, 47] further improve the efficiency of generative modeling by operating in compressed representation spaces rather than directly modeling high-dimensional audio signals.

Despite these advances, most recent TTS systems still rely on intermediate representations and external waveform decoders. Codec-based models [7, 6, 57] depend on the capacity and quality of neural audio codecs [72, 8, 32, 23, 70, 10, 63, 11]; Mel-spectrogram-based models require separately

*Equal contribution. Corresponding author: Sang-Hoon Lee (sanghoonlee@ajou.ac.kr)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

trained neural vocoders; and latent diffusion models require an additional pre-trained latent. These intermediate modules have been highly effective in practice, but they also introduce additional training stages, architectural dependencies, and upper-bound constraints on the final quality. In particular, high-fidelity waveform generation often requires GAN training, carefully designed architectures, or representation-specific optimization, making the overall pipeline increasingly complex.

Meanwhile, although end-to-end (E2E) TTS models [53, 26, 12] were explored in earlier studies, their limited scalability, training instability, and inferior zero-shot quality led the field to converge toward multi-stage pipelines. These limitations stem from the high temporal resolution of raw waveforms, which makes text-speech alignment difficult and often requires complex adversarial training for perceptually realistic waveform generation. Only E3-TTS [17] demonstrated the potential of waveform-level diffusion for TTS. However, it still relies on BERT-based text representations to facilitate text-waveform alignment, and directly applying diffusion modeling to waveform signals imposes a substantial computational burden due to the high temporal resolution of audio.

In this work, we propose **WaveGen**, a fully E2E diffusion-based text-to-waveform framework built upon **Spectral Forcing**. WaveGen directly generates high-resolution waveform from text within a single architecture. The key idea is to revisit the prediction and loss spaces of CFM for raw waveform generation. We found x-prediction with v-loss is essential for E2E waveform generation. Instead of forcing the model to learn waveform dynamics alone, Spectral Forcing jointly guides the model to learn an internal spectral trajectory while optimizing the target waveform trajectory. This design provides a structured inductive bias for high-resolution audio generation while preserving a truly E2E text-to-waveform formulation.

To this end, we carefully design WaveGen architecture with disentangled diffusion heads and introduce a multi-scale spectral loss for stable waveform-level flow matching. We further investigate the scalability of E2E TTS by increasing the model size from 0.1B to 0.6B parameters, showing that direct text-to-waveform generation can benefit from model scaling when the training objective and architecture are properly designed.
Our main contributions are summarized as follows:

- We propose **WaveGen**, a fully E2E text-to-waveform framework that directly generates raw waveform from text without separately trained intermediate representations.
- We introduce **Spectral Forcing**, which jointly learns an internal spectral trajectory and the target waveform trajectory, enabling stable CFM-based optimization in the high-resolution waveform domain.
- We investigate prediction and loss spaces for waveform generation and design a scalable waveform-level diffusion architecture with disentangled diffusion heads.
- We demonstrate that WaveGen achieves state-of-the-art zero-shot TTS performance using a single architecture, open-source data, and one-stage training, providing evidence that fully E2E text-to-waveform generation can be a viable alternative to increasingly complex multi-stage TTS pipelines.

2 Related Work

2.1 Conventional TTS

Traditional TTS systems consist of an acoustic model and a neural vocoder. Acoustic models generate acoustic features such as Mel-spectrogram, and neural vocoders convert acoustic features into raw waveform signals. Tacotron [64] paved the way of deep-learning based TTS system by introducing attention-based encoder-decoder structure to align text with speech, and Tacotron 2 [55] utilized WaveNet [44] to generate waveform from synthetic Mel-spectrogram. Transformer TTS [40] verified the effectiveness of encoder-decoder Transformer. However, these models suffer from unstable attention alignment and slow inference speed due to their autoregressive (AR) nature. To address these issues, FastSpeech [52] introduced parallel synthesis using a duration which is extracted from teacher AR models. Also, GAN-based neural vocoders accelerated TTS research in terms of both quality and inference speed. Specifically, PWG [67] and MelGAN [31] introduced GAN-based vocoder, and HiFi-GAN [29] introduced multi-period discriminator (MPD) which significantly enhances perceptual quality of synthetic speech. BigVGAN [35] scaled up the neural vocoder via snake periodic activation and an anti-aliasing low-pass filter. iSTFTNet [24] and Vocos [56] enhance the sampling efficiency by modeling features in a low-resolution space with iSTFT.

Table 1: Comparison of WaveGen with recent TTS systems. WaveGen uniquely achieves a fully E2E single-stage zero-shot TTS without relying on intermediate representations or auxiliary modules. T, C, W, M, L, and B denote text, codec, waveform, Mel, latent, and BERT, respectively.

Model	Qwen3-TTS	VoiceBox	E2/F5-TTS	CosyVoice3	VoxCPM	VITS	E3-TTS	**WaveGen**
End-to-End (T→W)	✗	✗	✗	✗	✗	✓	✗	✓
Single-stage	✗	✗	✗	✗	✗	✓	✗	✓
Zero-shot	✓	✓	✓	✓	✓	✗	✓	✓
Dependency on Intermediate Representations (Explicit Waveform Decoder)								
Codec	✓	✗	✗	✓	✗	✗	✗	✗
Vocoder	✗	✓	✓	✓	✗	✗	✗	✗
VAE / Latent Model	✗	✗	✗	✗	✓	✗	✗	✗
GAN-based Training	✓	✓	✓	✓	✓	✓	✗	✗
Dependency on External Alignment Modules								
Pre-trained LLM	✓	✗	✗	✓	✓	✗	✓	✗
Duration Predictor	✗	✓	✗	✗	✗	✓	✗	✗
Pipeline Complexity	T→C→W	T→M→W	T→M→W	T→C→M→W	T→L→W	T→W	T→B→W	**T→W**

Neural Codec Language Models With the emergence of large language models, AudioGen [30] introduces language model-based audio generative models based on text prompt and utilize RVQ-based neural audio codec as a discrete audio representation. Then, VALL-E [58] demonstrated neural codec language models can be a strong zero-shot TTS system with large-scale training and speech-level prompting. UniAudio [68] and Moshi [9] proposed efficient RVQ decoding method using multi-scale Transformer. Recently, most models utilize a pre-trained LLM for TTS backbone, and finetuned them to decode semantic token [13, 14] or RVQ tokens [27, 22, 18] which are extracted from large-scale high-quality audio codec via multi-token prediction (MTP) AR Transformer. Metis [62] and OmniVoice [78] employ a discrete diffusion language model using bidirectional Transformer.

Diffusion/CFM Grad-TTS [49] demonstrated the effectiveness of diffusion models for TTS. VoiceBox [33] and P-Flow [28] propose CFM-based TTS models via a masked-infilling training strategy, which significantly improve zero-shot speech synthesis through speech prompting. E2-TTS [15] introduces novel text-speech alignment learning with filler tokens. F5-TTS [5] and ZipVoice [77] introduce efficient CFM-based TTS models without any external duration predictor. Also, hybrid models such as CosyVoice [13, 14] utilized CFM-based Mel-spectrogram decoder conditioned on semantic tokens. Meanwhile, latent diffusion models (LDM) [34] have been investigated for efficient TTS modeling. Furthermore, continuous language models such as VoxCPM [75] and VibeVoice [47] show their effectiveness of AR diffusion via highly-compressed latent representation.

2.2 End-to-End TTS

E2E TTS has long been investigated as a way to simplify conventional cascaded speech synthesis pipelines. Early neural TTS systems such as Tacotron [64] established a unified sequence-to-sequence framework for text-to-acoustic modeling, but still required a Griffin-Lim algorithm [19]. ClariNet [48] explored parallel waveform generation through knowledge distillation from autoregressive WaveNet models. FastSpeech 2s [53] extended non-autoregressive TTS toward direct waveform generation. EATS [12] jointly optimized learnable duration modeling and waveform generation with adversarial training. VITS [26] introduced a strong text-to-waveform framework by combining variational inference, normalizing flows [25], and adversarial waveform training [29].

Despite these advances, prior E2E TTS systems still have important limitations: many rely on adversarial training to obtain perceptually realistic waveforms, which require carefully designed multiple discriminators, auxiliary losses, and significantly long training. More importantly, most of these models were not primarily designed for zero-shot speech synthesis, where the model must generate high-quality speech for unseen speakers from text and speech prompts. As a result, recent zero-shot TTS research has largely shifted toward codec-based language models, Mel-spectrogram diffusion models, or LDM rather than direct waveform-level generation.

Recently, E3-TTS [17] revisited diffusion-based text-to-waveform generation, demonstrating the potential of waveform-level generative modeling for TTS; however, it still relies on pretrained BERT-based text representations to facilitate text-waveform alignment, and directly applying diffusion modeling to high-resolution waveform signals introduces substantial computational cost. In contrast, this paper revisits fully E2E TTS under a stricter formulation: a single model directly generates waveform from text without neural audio codecs, Mel-spectrogram decoders, VAE, neural vocoders, duration predictors, self-supervised representations, or adversarial training as indicated in Table 1.

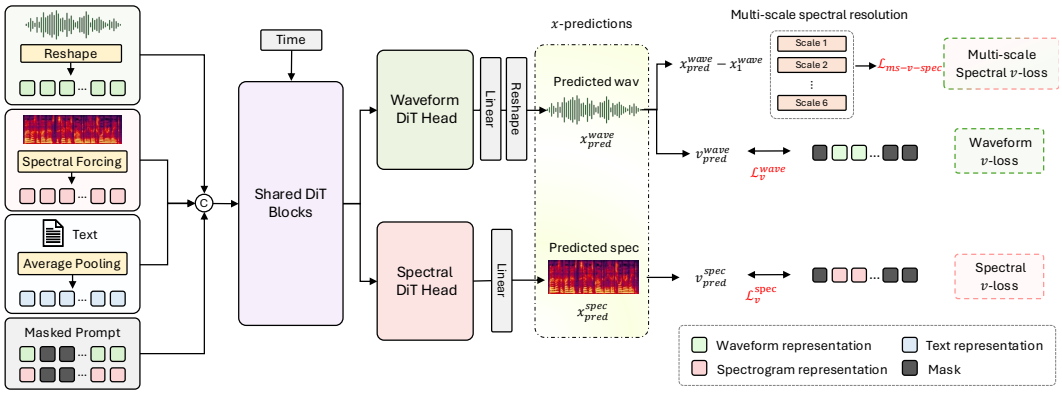


Figure 1: E2E text-to-waveform diffusion training details. We reshape high-resolution waveform signals (24,000 Hz) into a low-resolution space (100 Hz) for efficient E2E waveform diffusion. Then, we introduce Spectral Forcing for internal spectral guidance and simply concatenate average-upsampled text embeddings. We use shared DiT blocks for text-speech alignment and disentangled DiT heads for representation refinement. We adopt x-prediction as the prediction space, and v-loss as the target loss space, and introduce a new multi-scale v-space spectral loss

3 WaveGen

In this paper, we propose a fully E2E single-stage diffusion-based text-to-waveform generative model as illustrated in Figure 1. We analyze raw waveform as generative targets (Section 3.1), introduce a novel framework (Section 3.2), and efficient architecture (Section 3.3). Also, we revisit the prediction space and loss space for raw waveform (Section 3.4) and introduce the improved loss (Section 3.5).

3.1 CFM-based Waveform Generation

Recently, diffusion and CFM-based waveform generation models [54, 65, 38, 36] have emerged as a promising alternative to GAN-based models, offering a new direction for high-fidelity speech synthesis. PeriodWave [39] introduced a CFM-based periodic-aware generator for waveform generation. RFWave [42] presented efficient CFM-based models with STFT/iSTFT. Flow2GAN [69] improved CFM objective of waveform in spectral space. StreamFlow [6] proposed streaming flow matching for streaming waveform generation and efficient waveform diffusion Transformers using linear-reshape transformation. However, these works only employ waveform-level diffusion with strongly aligned acoustic conditions such as Mel-spectrogram or pre-trained neural audio codec. In our study, basic CFMs fail to align text-speech in high-resolution waveform space without any external modules.

3.2 Spectral Forcing

Due to the difficulty of waveform-level generation, previous works utilize intermediate representations such as Mel-spectrogram. To bridge the gap between text and waveform, we introduce Spectral Forcing, a method to jointly learn an internal spectral trajectory and the target waveform trajectory, enabling stable CFM-based optimization in the high-resolution waveform space. Specifically, we introduce a low-resolution spectral trajectory as an auxiliary generation target, which provides a structured inductive bias for text-speech alignment. Unlike cascaded approaches, the spectral trajectory is jointly generated within the same model and is used only as internal frequency-rich guidance, rather than as an external intermediate representation.

We adopt a simple DiT [45]-only architecture with AdaLN-SOLA [20]. The model takes as input the concatenation of the reshaped noisy waveform x_t^{wave} and its noisy spectral representation x_t^{spec} which is a lower-resolution Mel-spectrogram with 64 bins. For text conditioning, we adopt average pooling of ZipVoice [77, 76] that expands the average-pooled text embeddings along the temporal dimension to match the speech length without any external encoder. Following [33, 5], we utilize masked-infilling training for prompted generation by concatenating masked representations together.

3.3 Disentangled DiT Heads

We found that concatenating input representations is necessary for better text-speech alignment, but disentangling DiT heads for each representation is required for robust estimation. We use shared DiT blocks as an alignment backbone, and simply add shallow disentangled DiT heads as representation-specific generation. Baseline model uses 14 shared DiT blocks and two disentangled heads. We only compute the loss on the masked regions of the target representation.

3.4 Decoupled Prediction and Loss Spaces for Raw Waveform Generation

Recent flow matching-based waveform reconstruction models, such as PeriodWave [39] and RFWave [42], have demonstrated the potential of flow-based models for raw waveform reconstruction by improving periodic modeling and sampling efficiency. Despite these advances, we observe that conventional velocity-based objectives can be insufficient for preserving fine-grained waveform energy, often leading to degraded perceptual quality in generated audio.

Motivated by JiT [41], which highlights the effectiveness of clean-data prediction for high-dimensional raw signals, and Latent Forcing [2] and ReGen [37], which leverage latent-assisted generation trajectories to better structure raw-space generation, we revisit the design of prediction and loss spaces in raw waveform CFM. Specifically, our model predicts clean waveform and spectral targets through Waveform DiT and Spectral DiT, while computing the training objective in the velocity space. This decoupled x-prediction and v-loss strategy enables effective waveform energy modeling without introducing the severe audible artifacts observed with direct x-loss optimization. Following the standard flow interpolation, we construct the intermediate sample in each domain $d \in \{\text{wave}, \text{spec}\}$ as $x_t^d = (1-t)x_0^d + tx_1^d$, where x_0^d is Gaussian noise and x_1^d is the clean target. Instead of directly predicting the velocity field, the model predicts the clean target x_{pred}^d. We then convert this prediction into the velocity space as:

$$v_{pred}^d = \frac{x_{pred}^d - x_t^d}{1-t}, \qquad v_t^d = \frac{x_1^d - x_t^d}{1-t}. \tag{1}$$

The loss is computed only on the masked generation region:

$$\mathcal{L}_v^d = \frac{\left\| \mathcal{M}^d \odot \left(v_{pred}^d - v_t^d \right) \right\|_2^2}{\|\mathcal{M}^d\|_1}, \qquad d \in \{\text{wave}, \text{spec}\}, \tag{2}$$

where $\mathcal{M}^d$ denotes the domain-specific training mask and $\odot$ is element-wise multiplication. This yields the waveform velocity loss $\mathcal{L}_v^{wave}$ and the spectral velocity loss $\mathcal{L}_v^{spec}$.

3.5 Multi-Scale Energy-Normalized Spectral v-Loss

Flow2GAN [69] introduces spectral energy-adaptive loss scaling to alleviate the loss-perception mismatch in flow matching. Following this insight, we adopt energy-normalized spectral supervision for the CFM velocity objective and extend it to multiple spectral resolutions, enabling the model to capture time-frequency structures at different resolutions. As described in Section 3.4, our model adopts x-prediction while computing the objective in the velocity space. We define the waveform-domain velocity difference as:

$$\Delta_v^{wave} = x_{pred}^{wave} - x_1^{wave} = (1-t)\left(v_{pred}^{wave} - v_t^{wave} \right). \tag{3}$$

We use Δ_v^{wave} for spectral supervision. For each scale $k \in \mathcal{K}$, we first compute a power spectrogram and apply a linear filterbank projection:

$$\Phi_k(x) = \left| \text{STFT}_k(x) \right|^2 \mathbf{F}_k, \tag{4}$$

where $\mathbf{F}_k$ denotes the linear filterbank matrix at the k-th spectral resolution. The proposed multi-scale spectral v-loss is then defined as:

$$\mathcal{L}_{ms\text{-}v\text{-}spec} = \sum_{k \in \mathcal{K}} \frac{\left\| \mathcal{M}_k^{wave} \odot \Phi_k(\Delta_v^{wave}) \odot \left(\Phi_k(x_1^{wave}) + \epsilon \right)^{-\frac{1}{2}} \right\|_1}{\|\mathcal{M}_k^{wave}\|_1}, \tag{5}$$

where $\mathcal{M}_k^{wave}$ is the mask for the k-th spectral resolution. Here, $\mathcal{K}$ is a set of spectral configurations with $n_{\text{bin}} = [512, 256, 128, 64, 32, 16]$, $n_{\text{fft}} = w = [1920, 960, 480, 240, 120, 60]$, and $h = [480, 240, 120, 60, 30, 15]$, where w and h denote the window size and hop size, respectively. The linear filter projection decomposes wave-domain velocity difference across frequency bands, and the ground-truth energy normalization prevents high-energy regions from dominating the loss. We follow Flow2GAN [69] for the scaling hyperparameters, using $\epsilon = 10^{-7}$ and clamping the scale to $[0.01, 100]$. The final objective is denoted as:

$$\mathcal{L}_{total} = \mathcal{L}_v^{wave} + \lambda_{spec}\mathcal{L}_v^{spec} + \lambda_{ms}\mathcal{L}_{ms\text{-}v\text{-}spec}, \tag{6}$$

where $\lambda_{spec} = 0.1$ and $\lambda_{ms} = 0.1$.

Table 2: Zero-shot TTS results on Seed-en Benchmark. Note that parameter size (Params) of baseline models only indicates their TTS module, but they require additional modules for waveform decoding.

Model	E2E	Params	Data	Vocoder	Codec	VAE	SPK Enc.	LLM	GAN	Diffusion	**WER** ↓	**SIM** ↑
GT	-	-	-	-	-	-	-	-	-	-	2.14	0.73
SparkTTS [59]	✗	0.5B	100k	✗	✓	✗	✓	✓	✓	✗	1.98	0.58
TadiCodec-AR [60]	✗	4B	100k	✓	✓	✗	✗	✓	✓	✓	2.28	0.65
F5-TTS [5]	✗	0.3B	100k	✓	✗	✗	✗	✗	✓	✓	2.00	0.67
ZipVoice [77]	✗	0.1B	100k	✓	✗	✗	✗	✗	✓	✓	1.70	0.69
CosyVoice2 [13]	✗	0.5B	166k	✓	✓	✗	✓	✓	✓	✓	3.09	0.65
CosyVoice3-RL [14]	✗	0.5B	-	✓	✓	✗	✓	✓	✓	✓	1.68	0.69
Qwen3-TTS [22]	✗	0.6B	-	✓	✓	✗	✓	✓	✓	✗	1.68	0.70
Qwen3-TTS [22]	✗	1.7B	-	✓	✓	✗	✓	✓	✓	✗	1.50	0.71
MOSS-TTS [18]	✗	1.7B	-	✓	✓	✗	✓	✓	✓	✗	1.93	0.73
VibeVoice [46]	✗	1.5B	-	✗	✗	✓	✗	✓	✓	✓	3.04	0.68
VibeVoice-RT [46]	✗	0.5B	-	✗	✗	✓	✗	✓	✓	✓	2.05	0.63
VoxCPM-Emilia [75]	✗	0.5B	100k	✗	✗	✓	✗	✓	✓	✓	2.34	0.68
VoxCPM [75]	✗	0.5B	-	✗	✗	✓	✗	✓	✓	✓	1.85	0.72
WaveGen-S	✓	0.1B	120k	✗	✗	✗	✗	✗	✗	✓	4.42	0.53
WaveGen-B	✓	0.4B	120k	✗	✗	✗	✗	✗	✗	✓	1.77	0.63
WaveGen	✓	0.6B	100k	✗	✗	✗	✗	✗	✗	✓	1.47	0.75

Table 3: Subjective evaluation results on Seed-en benchmark

Model	GT	F5-TTS	CosyVoice3	Qwen3-TTS-0.6B	Qwen3-TTS-1.7B	WaveGen-S	WaveGen-B
nMOS	4.117±0.112	3.938±0.119	3.917±0.128	3.883±0.123	3.954±0.117	3.958±0.129	4.000±0.126
sMOS	3.932±0.038	3.989±0.037	3.972±0.038	3.968±0.037	4.008±0.037	3.952±0.037	3.988±0.037

4 Experiment and Result

4.1 Experimental Setup

Dataset For small-scale TTS and ablation studies, we utilized LibriTTS (CC-BY-4.0) [73]. LibriTTS is a high-quality speech dataset with a sampling rate of 24,000 Hz in WAV format. We filtered LibriTTS to include only utterances with durations of up to 12s for efficient training (319,627 samples). For large-scale training, WaveGen-S/B (v0.1) use the English subsets of Emilia (CC-BY-NC-4.0) and Emilia-YODAS (CC-BY-4.0) [21] provided as MP3-compressed audio with a sampling rate of 24,000 Hz. Following [5], we filter out several speakers and use utterances with durations of up to 30s. We also released WaveGen (0.6B) trained with multilingual Emilia datasets.

Training We train LibriTTS models with a learning rate of 1×10^{-4} using the AdamW optimizer [43], a fixed batch size of 128 for 1.1M steps. We train Emilia models with a learning rate of 1×10^{-4} using AdamW optimizer [43], a dynamic batch size with total 480s frame of waveform for 1.1M steps. All models except for the 0.6B models are trained on four GPUs using mixed precision. Due to limited computational resources, we train the models only for four epochs. We train WaveGen (0.6B) on eight GPUs for 11 epochs. During training, we use a masked-infilling strategy by dropping the speech prompt and text condition (0.3 and 0.2, respectively).

Sampling We adopt the Euler method as the ODE solver. For CFG scheduling, we build upon the strategy used in ZipVoice, where the text condition is dropped only at early time steps, and all ablation studies are conducted using this. However, when using a threshold of $t < 0.5$, we found that this substantially degrades speaker similarity in our model. Therefore, we restrict text-condition dropout to $t < 0.02$. We fix NFE to 32 and CFG to 2 in our experiments.

4.2 Zero-shot TTS results

We evaluate zero-shot TTS performance on the Seed-en benchmark dataset [1]. We compute WER and SIM to assess speech intelligibility and speaker similarity using Whisper and WavLM-Large, respectively. Table 2 demonstrates the effectiveness of E2E TTS training without relying on external modules, and shows the scalability of the proposed models by achieving promising performance compared to large-scale TTS systems that require multi-stage training.

We also conduct a subjective evaluation using Amazon Mechanical Turk (MTurk) with details provided in Appendix D. We compare WaveGen with publicly available strong baselines, including F5-TTS [5], CosyVoice3 [14], and Qwen3-TTS [22]. Notably, our model achieves comparable mean opinion scores (MOS) for both naturalness (nMOS) and similarity (sMOS) as reported in Table 3. We encourage readers to listen to the audio samples on our demo page. Additionally, we provide extensive ablation studies in the following subsections.

4.3 Spectral Forcing

We found that learning the waveform trajectory alone does not guarantee text-speech alignment. To bridge the gap between text and speech, we introduce Spectral Forcing, which jointly learns an internal spectral trajectory within a single model. We first use a linear spectrogram as the target spectral representation, but it leads to unstable training. We therefore adopt the Mel-spectrogram as a target representation and analyze the effect of the number of Mel bins in Table 4. Larger Mel bins make the model focus more on waveform reconstruction rather than text-speech alignment. In contrast, smaller Mel bins provide a stronger information bottleneck, encouraging earlier alignment and faster convergence. We adopt 64 Mel bins, which offers a favorable trade-off between alignment, convergence, and high-frequency detail generation.

Table 4: Comparison of Spectral Forcing

Type	Bin	WER	SIM	DNSMOS	Analysis
No SF	-	123.67	0.19	3.20	Fails to align
Mel	256	126.50	0.35	3.17	Denoising
Mel	128	18.67	0.52	3.55	Reconstruction
Mel	80	6.80	0.50	3.62	Mel Bins ↓↑
Mel	64	4.96	0.47	3.72	Faster
Mel	32	4.26	0.48	3.64	Convergence
Linear	481		NaN		
Linear	128		NaN		Unstable
Linear	64		NaN		

4.4 End-to-End Waveform Generation

To evaluate the effectiveness of E2E training, we train a two-stage TTS system consisting of a text-to-Mel model and a vocoder. For both stages, we use the same DiT architecture with masked-infilling training. Table 5 shows that the vocoder can be successfully trained even without GAN-based training. However, the two-stage pipeline still suffers from the inherent limitations of cascaded generation, where the intermediate Mel representation constrains the waveform generation process and errors from the acoustic model can propagate to the vocoder. In contrast, E2E models directly optimize text-to-waveform generation within a single model, achieving better WER, SIM, and DNSMOS under the same NFEs (total NFE of 32). These results demonstrate the advantage of E2E over cascaded TTS systems. We also report the real-time factor (RTF) on NVIDIA A100 without inference optimization.

Table 5: Comparison of E2E and cascaded models

Method	Params.	NFE	RTF	WER	SIM	DNSMOS
End-to-End	407M	32	0.24	2.91	0.53	3.67
End-to-End	210M	32	0.12	4.68	0.50	3.71
End-to-End	113M	32	0.08	4.96	0.47	3.72
Vocoding (GT Mel)	113M	32	0.08	1.47	0.70	3.32
Vocoding (GT Mel)	113M	16	0.04	1.57	0.70	3.33
Two-Stage	226M	32 + 32	0.16	4.39	0.46	3.53
Two-Stage	226M	16 + 16	0.09	5.17	0.47	3.59

4.5 Architecture Design

We compare the effectiveness of disentangled DiT heads for multi-representation generation. Using only shared DiT blocks shows poor performance, suggesting that a fully shared architecture is insufficient for generating multiple representations. We therefore add shallow disentangled DiT heads on top of shared DiT blocks, enabling the model to maintain a shared alignment backbone while learning representation-specific refinement. Table 6 shows that this design substantially improves WER, SIM, and DNSMOS. However, using too many disentangled blocks degrades performance due to degraded alignment. We therefore adopt 14 shared DiT blocks with 2 disentangled DiT blocks.

Table 6: Disentangled DiT Heads

#Shared DiT	# Disentangled DiT	WER	SIM	DNSMOS
18	0	7.83	0.44	3.60
14	2 (4)	4.96	0.47	3.72
10	4 (8)	7.64	0.46	3.65

4.6 Decoupled Prediction and Loss Spaces

We investigate the prediction and loss spaces for E2E TTS. Unlike previous CFM-based waveform generation models, which generate waveforms conditioned on Mel-spectrograms, E2E TTS must directly learn the energy distribution of target waveforms without relying on ground-truth acoustic conditioning during generation. Although v-prediction yields slightly better text-speech alignment and speaker similarity where v-prediction focuses low-frequency information due to noisy target velocity, we observe that it lacks the ability to accurately estimate waveform energy, resulting in significant degradation in perceptual quality. In contrast, Table 7 shows that x-prediction generates waveforms with more accurate energy distribution and significantly improves the perceptual quality of speech. This observation is particularly important because it suggests that high-quality waveform generation can be achieved without adversarial training. Note that the x-loss space introduces significant metallic noise and should therefore be avoided in current systems.

Table 7: Comparison of prediction space

Pred	Loss	SF	WER	SIM	DNSMOS
x-pred	v-loss	✗	123.67	0.19	3.20
v-pred	v-loss	✗	119.52	0.16	2.65
x-pred	v-loss	✓	5.57	0.46	3.45
v-pred	v-loss	✓	3.93	0.48	2.94

4.7 Multi-scale Spectral v-loss

To further improve high-frequency detail reconstruction in CFM-based waveform generation, we investigate an enhanced CFM objective in the spectral domain. Flow2GAN [69] introduced a novel spectral loss that minimizes waveform errors with spectral energy scaling, enabling balanced frequency-wise optimization. This differs from conventional spectrogram losses, which directly minimize the discrepancy between predicted and target spectrograms. However, we found that such a loss tends to overemphasize high-frequency details at the expense of low-frequency structure, which can make text-speech alignment more difficult in E2E TTS. Unlike neural vocoders, E2E TTS models cannot rely on ground-truth spectral inputs during generation. To address this issue, we propose a multi-scale spectral v-loss that balances time-frequency optimization across multiple resolutions. The proposed loss improves overall frequency details and leads to better speaker similarity and perceptual quality as reported in Table 8. The final training objective combines the original CFM losses for wave and spec domains with our proposed multi-scale spectral v-loss.

Table 8: Loss Comparison

Method	Loss	WER	SIM	DNSMOS
Orig. CFM [39]	$\mathcal{L}_v$	5.57	0.46	3.45
Modified CFM [69]	$\mathcal{L}_{v\text{-}spec}$	68.82	0.46	3.49
Ours (Ablation)	$\mathcal{L}_{ms\text{-}v\text{-}spec}$	16.98	0.51	3.57
Ours (Final)	$\mathcal{L}_v + \lambda_{ms}\mathcal{L}_{ms\text{-}v\text{-}spec}$	4.96	0.47	3.72

4.8 Preliminary Study on REPA

The results reported for our main WaveGen models do not use representation alignment (REPA) [71] or any external modules. However, recent studies on REPA suggest that external representation guidance can improve diffusion training. We conduct a small-scale preliminary experiment (WaveGen-S, LibriTTS) to examine whether REPA can further improve E2E waveform generation. We apply REPA to the middle layer (10th layer) using semantic representations extracted from the encoder output of Whisper-small, and adopt a cosine similarity-based alignment loss weighted by $\lambda_{repa} = 0.1$. As shown in Table 9, REPA improves WER and SIM while maintaining comparable DNSMOS, suggesting faster convergence and improved training stability. We leave further improvements with REPA for future work.

Table 9: REPA result

Method	Loss	WER	SIM	DNSMOS
Ours (Final)	$\mathcal{L}_v + \lambda_{ms}\mathcal{L}_{ms\text{-}v\text{-}spec}$	4.96	0.47	3.72
+REPA (Optional)	$+\lambda_{repa}\mathcal{L}_{repa}$ (Whisper)	3.54	0.54	3.70

4.9 Scalability

Table 10: Model comparison with respect to model size and dataset scale

Model	Params.	Data (hours)	Training Steps	Basic CFG (2.0) WER ↓	Basic CFG (2.0) SIM ↑	Basic CFG (2.0) DNSMOS ↑	Modified CFG (2.0) WER ↓	Modified CFG (2.0) SIM ↑	Modified CFG (2.0) DNSMOS ↑
WaveGen-S	113M	0.5k	1.1M	4.96	0.47	3.72	4.64	0.49	3.57
WaveGen-B	407M	0.5k	1.1M	2.91	0.53	3.67	2.93	0.55	3.59
WaveGen-S	113M	120k	1.1M	4.27	0.49	3.60	4.42	0.53	3.56
WaveGen-B	407M	120k	1.1M	2.37	0.59	3.60	1.77	0.63	3.60

We investigate the scalability of E2E TTS models with respect to both model size and dataset scale. Specifically, we train two model variants, S and B, with 0.1B and 0.4B parameters, respectively. Each model is trained on both LibriTTS (0.5k hours) and Emilia (120k hours). As shown in Table 10, the proposed models show consistent improvement in intelligibility and similarity as both model size and dataset scale increase. Furthermore, we find that the modified CFG strategy improves speaker similarity, suggesting that the initial sampling steps play an important role in adapting speaker characteristics. We provide additional results with different training steps in Table 14. Based on the trends observed from extended training and the REPA results in Table 9, we believe that further training can improve the overall quality of our models. However, due to limited computational resources, we train the models for only four epochs in this study.

We further investigate the effect of NFE on E2E TTS. Table 11 presents the results of WaveGen-B with different NFE settings. We find that speaking style is adapted in the early sampling stages, and high-quality waveform generation requires at least eight steps. Increasing the number of sampling steps generally improves similarity and perceptual quality. These results suggest that early sampling steps play a crucial role in establishing text-speech alignment and adapting speaking style, whereas additional steps mainly refine waveform.

Table 11: Analysis of NFE (Seed-en)

Model	NFE	WER	SIM	DNSMOS
WaveGen-B	4	52.0	0.39	3.06
	8	4.85	0.61	3.48
	16	2.16	0.63	3.56
	32	1.77	0.63	3.60
	50	1.87	0.64	3.61
	100	1.84	0.64	3.63

Table 12: Multilingual TTS results using F5-TTS architecture and large-scale training configuration. For a fair comparison, all models are trained with the open-source Emilia dataset about 100k hours (English and Chinese). With end-to-end waveform generation, our models achieve the state-of-the-art performance on Seed-TTS-Eval benchmarks using public-available datasets.

Model	E2E	SF (Drop)	Anchoring	Hz	Hop (Frame)	Params	Data	seed-en			seed-zh		
								WER ↓	SIM↑	UTMOS ↑	CER ↓	SIM↑	UTMOS ↑
GT	-	-	-	24k	-	-	-	2.06	0.73	3.53	1.26	0.76	2.78
Seed-TTS [1]	✗	✗	-	24k	-	-	-	2.25	0.76	-	1.12	0.79	-
VoxCPM-Emilia [74]	✗	✗	-	16k	-	0.5B	100K	2.34	0.68	-	1.11	0.74	-
MaskGCT [61]	✗	✗	-	24k	-	2.2B	100K	2.88	0.71	3.55	2.40	0.77	2.63
OmniVoice-Emilia [78]	✗	✗	-	24k	-	0.8B	100K	1.72	0.71	3.88	0.89	0.76	3.05
ZipVoice [77]	✗	✗	-	24k	256 (93.75)	123M	100K	1.70	0.69	3.82	1.40	0.75	3.15
E2-TTS [15]	✗	✗	-	24k	256 (93.75)	333M	100K	2.19	0.71	3.20	1.97	0.73	2.27
F5-TTS [5]	✗	✗	-	24k	256 (93.75)	336M	100K	1.83	0.67	3.76	1.56	0.76	2.95
BareWave [16]	✓	✗	-	24k	768 (31.25)	983M	19.4K	1.75	0.60	3.72	-	-	-
WavTTS [4]	✓	✗	-	16k	160 (100)	673M	100K	1.50	0.65	3.92	1.59	0.73	3.08
WaveGen-16k	✓	✓(0.5)	✗	16k	160 (100)	676M	100K	1.51	0.71	3.90	1.65	0.77	3.09
WaveGen-16k	✓	✓(0.5)	✓	16k	160 (100)	676M	100K	1.69	0.75	3.79	1.92	0.80	3.01
WaveGen	✓	✓(0.2)	✗	24k	480 (50)	680M	100K	1.55	0.72	3.68	1.66	0.78	2.84
WaveGen	✓	✓(0.2)	✓	24k	480 (50)	680M	100K	1.47	0.75	3.68	1.75	0.80	2.87
WaveGen (Ablation)	✓	✓(0)	✗	24k	480 (50)	680M	100K	2.07	0.73	3.60	2.33	0.79	2.76
WaveGen (Ablation)	✓	✓(0)	✓	24k	480 (50)	680M	100K	1.97	0.75	3.70	2.11	0.80	2.85
WaveGen (Ablation)	✓	✓(0.5)	✗	24k	480 (50)	680M	100K	1.39	0.70	3.81	1.58	0.76	2.97
WaveGen (Ablation)	✓	✓(0.5)	✓	24k	480 (50)	680M	100K	1.56	0.74	3.70	1.74	0.79	2.87

4.10 Multilingual TTS results

We conduct large-scale multilingual TTS experiments on the English (En) and Chinese (Zh) subsets of the Emilia dataset, and adopt F5-TTS-based architectures including ConvNeXt-based text encoder and filler token-based conditioning and hyperparameters of WavTTS including waveform scaling and time scheduling for fair comparison.[2] Table 12 shows that WaveGen achieves strong performance across all evaluation metrics, demonstrating the effectiveness of direct waveform modeling and spectral forcing. Specifically, WaveGen achieves speaker similarities of 0.75 and 0.80 on the seed-en and seed-zh benchmark using only the publicly available Emilia dataset. Notably, we do not employ REPA when training WaveGen, suggesting further room for improvement through additional representation alignment techniques. In the following subsections and Appendix F, we further provide detailed analyses of the waveform diffusion optimization techniques used in WaveGen.

4.11 Information bottleneck pre-net for multi-representation generation

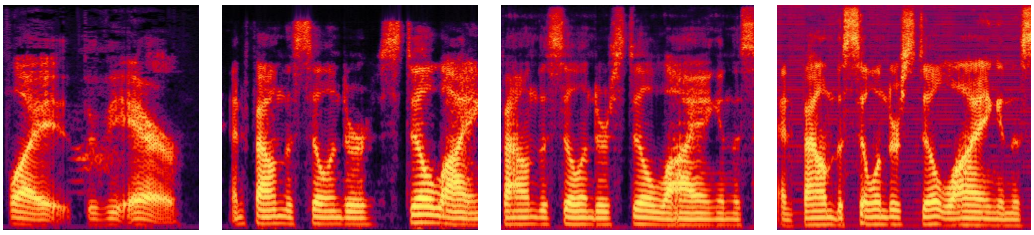
We further investigate the effect of Spectral Forcing during large-scale training. When using a large-scale dataset, the dependency of Spectral Forcing increases because the model has difficulty aligning text with speech at the initial stage. While Spectral Forcing is weak guidance, the model is prone to learn to reconstruct waveforms from Spectral Forcing, resulting in delayed emergence of text-speech alignment. To reduce this limitation, we revisit the simple techniques, pre-net used in Tacotron [64] by adding information bottleneck layers with two MLP layers and dropout to reduce dependency on teacher forcing for AR training. Specifically, we utilize 8n Mel bins and smaller bottleneck dimension of 32 similar to [3]. We apply this bottleneck layer with dropout of 0.2 during training and inference. It improves the robustness of multi-representation generation by learning to generate waveform from multiple conditions. We investigated the effect of dropout ratio on Spectral Forcing in Appendix F.2.

4.12 Prompt Anchor

Following [66], we observe a train-inference mismatch in iterative generation under masked-infilling training. During training, the prompt region is fixed to the GT speech and is therefore not explicitly optimized for iterative updates. At inference time, however, repeatedly updating x_t in the prompt region can introduce undesirable deviations from the GT prompt trajectory.

To address this issue, we anchor the prompt region to a GT-based intermediate state during iterative sampling. Specifically, at each sampling step, we construct the prompt-region x_t by interpolating between the corresponding x_0 and x_1, rather than using the model-updated values. This simple prompt anchoring strategy substantially improves speaker similarity by reducing the train-inference mismatch introduced by masked-infilling training.

[2]To the best of our knowledge, WaveGen is the first CFM-based end-to-end waveform TTS model to outperform conventional multi-stage TTS systems on the Emilia dataset. WaveGen also achieves the best performance among the evaluated waveform diffusion TTS models, including WavTTS and BareWave, both of which were released after our NeurIPS 2026 submission. Nevertheless, we acknowledge that WavTTS provides valuable analyses of optimization strategies for waveform diffusion, which complement our findings and offer useful insights into improving the training of waveform TTS systems.



(a) Prompt Audio (MP3 Format) (b) Gen. Audio (Trained with Emilia) (c) Gen. Audio (Trained with LibriTTS) (d) Gen. Audio (Trained with v-pred)

Figure 2: Spectral analysis with respect to audio format and prediction space

5 Spectral Analysis of Audio Format and Prediction Space

Impact of Audio Compression Neural vocoders and audio codecs are commonly trained on high-quality waveform datasets, and their training can often be effective even with relatively small datasets. By contrast, modern TTS models require large-scale training datasets to model diverse speaking styles including prosody and timbre in zero-shot scenarios. Although Emilia enables large-scale training, its MP3-compressed audio can limit the amount of recoverable high-frequency detail, which is particularly important for end-to-end waveform generation. As a result, models trained on such data tend to struggle with generating high-frequency components above 8 kHz. In contrast, models trained on LibriTTS preserve and generate richer high-frequency information as illustrated in Figure 2. This observation highlights the importance of carefully curating training data for E2E TTS, where the model must directly learn both linguistic alignment and high-fidelity waveform generation.

Effect of Prediction Space x-prediction plays a crucial role in high-quality waveform generation. As shown in Figure 2, the model trained with v-prediction fails to recover a reliable energy distribution because the target velocity is highly noisy, resulting in substantially degraded waveform quality.

6 Potential Broader Impact

Societal Positive Impact Our work proposes an E2E TTS model that directly generates high-quality waveform from text, thereby simplifying existing complex multi-stage TTS pipelines and enhancing training and inference efficiency. This efficient simplification, which enables speech generation with a single model without relying on external modules, enhances the accessibility of speech technologies and provides a foundation for broader adoption in both academic and industrial settings. Furthermore, in terms of application technology, this method can have a positive impact on diverse fields such as voice interfaces, education, and media content. In particular, the generation of clear and natural speech can improve the user experience and contribute to the advancement of speech language model.

Societal Negative Impact Although the model and method proposed in this work do not directly raise ethical issues, there is a possibility of misuse when combined with voice cloning due to their ability to generate high-quality speech. For example, the model could be used to imitate a specific speaker's voice or generate realistic fake speech to convey misleading information, which may lead to issues such as voice phishing or impersonation. To mitigate these potential risks, it is necessary to pursue complementary research on fake audio detection as well as speech safety, including methods for identification and authenticity verification. As part of future work, we plan to further explore watermarking and authentication techniques to enable tracing the origin of generated audio.

Limitation We introduce an E2E TTS model that does not rely on any external modules, including SSL models. Nevertheless, SSL-based guidance, such as REPA, can be incorporated to accelerate training. Due to limited computational resources, we restrict large-scale training to four epochs. The observed scaling trend suggests that longer training may further improve performance, which we leave for future work. In addition, we plan to extend our model to 48 kHz waveform generation.

7 Conclusion

We proposed WaveGen, a fully E2E diffusion-based text-to-waveform TTS system. Unlike previous TTS systems, WaveGen directly generates raw waveforms from text without separately trained intermediate representations. By introducing Spectral Forcing, disentangled DiT heads, and a multi-scale spectral v-loss, WaveGen enables stable CFM-based optimization in the high-resolution waveform domain. Our results show that direct text-to-waveform generation can achieve promising zero-shot TTS performance and benefit from model and dataset scaling, suggesting a simple and scalable alternative to increasingly complex cascaded TTS pipelines.

8 Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02283048, Developing the Next-Generation General AI with Reliability, Ethics, and Adaptability, IITP-2026-RS-2023-00255968, the Artificial Intelligence Convergence Innovation Human Resources Development, RS-2026-25617480, AI Star Fellowship Support Program(Gyeongsang National University)), National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25518808), the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT), and Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea)&Gwangju Metropolitan City.

References

[1] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

[2] Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

[3] Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, and Tao Qin. MultiSpeech: Multi-Speaker Text to Speech with Transformer. In *Interspeech 2020*, pages 4024–4028, 2020. doi: 10.21437/Interspeech.2020-3139.

[4] Wenxi Chen, Dongya Jia, Yushen Chen, Zhikang Niu, Yuzhe Liang, Xiquan Li, Ruiqi Yan, Ziyang Ma, Guanrou Yang, Sanyuan Chen, et al. Wavtts: Towards high-quality zero-shot tts via direct raw waveform modeling. *arXiv preprint arXiv:2606.03455*, 2026.

[5] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*, 2024.

[6] Ha-Yeong Choi and Sang-Hoon Lee. Streamflow: Streaming audio generation from discrete tokens via streaming flow matching. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 4669–4697. Curran Associates, Inc., 2025. doi: 10.52202/085713-0165. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0713495297dab18fabca4795cb9ef8ef-Paper-Conference.pdf.

[7] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

[8] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ivCd8z8zR2. Featured Certification, Reproducibility Certification.

[9] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[10] Luca Della Libera, Francesco Paissan, Cem Subakan, and Mirco Ravanelli. Focalcodec: Low-bitrate speech coding via focal modulation networks. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 23742–23767. Curran Associates, Inc., 2025. doi: 10.52202/085713-0801. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/220e6dbed88faa83bd9f9e8fa359e43d-Paper-Conference.pdf.

[11] Luca Della Libera, Cem Subakan, and Mirco Ravanelli. Beyond fixed frames: Dynamic character-aligned speech tokenization. *arXiv preprint arXiv:2601.23174*, 2026.

[12] Jeff Donahue, Sander Dieleman, Mikolaj Binkowski, Erich Elsen, and Karen Simonyan. End-to-end adversarial text-to-speech. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rsf1z-JSj87.

[13] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

[14] Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

[15] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. *arXiv preprint arXiv:2406.18009*, 2024.

[16] Wei Fan, Chao-Hong Tan, Qian Chen, Wen Wang, Xiangang Li, Kejiang Chen, Weiming Zhang, and Nenghai Yu. Barewave: Waveform-native flow-matching text-to-speech. *arXiv preprint arXiv:2606.09048*, 2026.

[17] Yuan Gao, Nobuyuki Morioka, Yu Zhang, and Nanxin Chen. E3 tts: Easy end-to-end diffusion-based text to speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.

[18] Yitian Gong, Botian Jiang, Yiwei Zhao, Yucheng Yuan, Kuangwei Chen, Yaozhou Jiang, Cheng Chang, Dong Hong, Mingshu Chen, Ruixiao Li, et al. Moss-tts technical report. *arXiv preprint arXiv:2603.18090*, 2026.

[19] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[20] Jiarui Hai, Yong Xu, Hao Zhang, Chenxing Li, Helin Wang, Mounya Elhilali, and Dong Yu. EzAudio: Enhancing Text-to-Audio Generation with Efficient Diffusion Transformer. In *Interspeech 2025*, pages 4233–4237, 2025. doi: 10.21437/Interspeech.2025-1137.

[21] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: A large-scale, extensive, multilingual, and diverse dataset for speech generation. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[22] Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026.

[23] Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, and Zhou Zhao. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yBlVlS2Fd9.

[24] Takuhiro Kaneko, Kou Tanaka, Hirokazu Kameoka, and Shogo Seki. istftnet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time fourier transform. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6207–6211. IEEE, 2022.

[25] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020.

[26] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR, 2021.

[27] Jaehyeon Kim, Taehong Moon, Keon Lee, and Jaewoong Cho. Efficient generative modeling with residual vector quantization-based tokens. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f8uPzuMNDq.

[28] Sungwon Kim, Kevin Shih, rohan badlani, Joao Felipe Santos, Evelina Bakhturina, Mikyas Desta, Rafael Valle, Sungroh Yoon, and Bryan Catanzaro. P-flow: A fast and data-efficient zero-shot tts through speech prompting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 74213–74228. Curran Associates, Inc., 2023. doi: 10.52202/075280-3246. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/eb0965da1d2cb3fbbbb8dbbad5fa0bfc-Paper-Conference.pdf.

[29] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

[30] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CYK7RfcOzQ4.

[31] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32, 2019.

[32] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993, 2023.

[33] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. Voicebox: Text-guided multilingual universal speech generation at scale. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 14005–14034. Curran Associates, Inc., 2023. doi: 10.52202/075280-0618. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2d8911db9ecedf866015091b28946e15-Paper-Conference.pdf.

[34] Keon Lee, Dong Won Kim, Jaehyeon Kim, and Jaewoong Cho. Ditto-tts: Efficient and scalable zero-shot text-to-speech with diffusion transformer. *arXiv preprint arXiv:2406.11427*, 2024.

[35] Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. BigVGAN: A universal neural vocoder with large-scale training. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iTtGCMDEzS_.

[36] Sang-Hoon Lee and Ha-Yeong Choi. Hierarchical representation alignment learning of diffusion transformers for neural audio codec. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32410–32426, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1622. URL https://aclanthology.org/2026.findings-acl.1622/.

[37] Sang-Hoon Lee and Ha-Yeong Choi. Regen: Hierarchical multi-prompt representation generation for efficient waveform diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=T8Y9elIrXa.

[38] Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. Accelerating high-fidelity waveform generation via adversarial flow matching optimization. *arXiv preprint arXiv:2408.08019*, 2024.

[39] Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. Periodwave: Multi-period flow matching for high-fidelity waveform generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tQ1PmLfPBL.

[40] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713, 2019.

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

[42] Peng Liu, Dongyang Dai, and Zhiyong Wu. RFWave: Multi-band rectified flow for audio waveform reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gRmWtOnTLK.

[43] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[44] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[46] Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, et al. Vibevoice technical report. *arXiv preprint arXiv:2508.19205*, 2025.

[47] Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, Shaohan Huang, Yan Xia, and Furu Wei. Vibevoice: Expressive podcast generation with next-token diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FihSkzyxdv.

[48] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HklY120cYm.

[49] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International conference on machine learning*, pages 8599–8608. PMLR, 2021.

[50] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[51] Chandan KA Reddy, Vishak Gopal, and Ross Cutler. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6493–6497. IEEE, 2021.

[52] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. *Advances in neural information processing systems*, 32, 2019.

[53] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=piLPYqxtWuA.

[54] Robin San Roman, Yossi Adi, Antoine Deleforge, Romain Serizel, Gabriel Synnaeve, and Alexandre Défossez. From discrete tokens to high-fidelity audio using multi-band diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dOanKg3jKS.

[55] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[56] Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vY9nzQmQBw.

[57] Jinchuan Tian, Sang gil Lee, Zhifeng Kong, Sreyan Ghosh, Arushi Goel, Chao-Han Huck Yang, Wenliang Dai, Zihan Liu, Hanrong Ye, Shinji Watanabe, Mohammad Shoeybi, Bryan Catanzaro, Rafael Valle, and Wei Ping. UALM: Unified audio language model for understanding, generation and reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TsdlOjcQNu.

[58] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[59] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, et al. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*, 2025.

[60] Yuancheng Wang, Dekun Chen, Xueyao Zhang, Junan Zhang, Jiaqi Li, and Zhizheng Wu. Tadicodec: Text-aware diffusion speech tokenizer for speech language modeling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 147494–147523. Curran Associates, Inc., 2025. doi: 10.52202/085713-4932. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d8a12fde9e72444e1b356e8c37e53753-Paper-Conference.pdf.

[61] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. In *International Conference on Learning Representations*, volume 2025, pages 47127–47150, 2025.

[62] Yuancheng Wang, Jiachen Zheng, Junan Zhang, Xueyao Zhang, Huan Liao, and Zhizheng Wu. Metis: A foundation speech generation model with masked generative pre-training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 131615–131645. Curran Associates, Inc., 2025. doi: 10.52202/085713-4381. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/be8987a75afeeadfa65007cf7ee25de0-Paper-Conference.pdf.

[63] Yuancheng Wang, Zhenyu Tang, Yun Wang, Arthur Hinsvark, Yingru Liu, Yinghao Aaron Li, Kainan Peng, Junyi Ao, Mingbo Ma, Mike Seltzer, Qing He, and Xubo Liu. Scaling speech tokenizers with diffusion autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=llMfmDtWka.

[64] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards End-to-End Speech Synthesis. In *Interspeech 2017*, pages 4006–4010, 2017. doi: 10.21437/Interspeech.2017-1452.

[65] Simon Welker, Matthew Le, Ricky T. Q. Chen, Wei-Ning Hsu, Timo Gerkmann, Alexander Richard, and YI-CHIAO WU. Flowdec: A flow-based full-band general audio codec with high perceptual quality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uxDFlPGRLX.

[66] Detai Xin, Shujie Hu, Chengzuo Yang, Chen Huang, Guoqiao Yu, Guanglu Wan, and Xunliang Cai. Longcat-audiodit: High-fidelity diffusion text-to-speech in the waveform latent space. *arXiv preprint arXiv:2603.29339*, 2026.

[67] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[68] Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

[69] Zengwei Yao, Wei Kang, Han Zhu, Liyong Guo, Lingxuan Ye, Fangjun Kuang, Weiji Zhuang, Zhaoqing Li, Zhifeng Han, Long Lin, and Daniel Povey. Flow2GAN: Hybrid flow matching and GAN with multi-resolution network for few-step high-fidelity audio generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5eTpRIULtb.

[70] Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, et al. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25697–25705, 2025.

[71] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[72] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[73] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Proc. Interspeech 2019*, pages 1526–1530, 2019. doi: 10.21437/Interspeech.2019-2441.

[74] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, et al. Voxcpm: Tokenizer-free tts for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*, 2025.

[75] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, Zhiyong Wu, and Zhiyuan Liu. Hierarchical semantic-acoustic modeling via semi-discrete residual representations for expressive end-to-end speech synthesis. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=h5KLpGoqzC.

[76] Han Zhu, Wei Kang, Liyong Guo, Zengwei Yao, Fangjun Kuang, Weiji Zhuang, Zhaoqing Li, Zhifeng Han, Dong Zhang, Xin Zhang, et al. Zipvoice-dialog: Non-autoregressive spoken dialogue generation with flow matching. *arXiv preprint arXiv:2507.09318*, 2025.

[77] Han Zhu, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhaoqing Li, Weiji Zhuang, Long Lin, and Daniel Povey. Zipvoice: Fast and high-quality zero-shot text-to-speech with flow matching. *arXiv preprint arXiv:2506.13053*, 2025.

[78] Han Zhu, Lingxuan Ye, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhifeng Han, Weiji Zhuang, Long Lin, and Daniel Povey. Omnivoice: Towards omnilingual zero-shot text-to-speech with diffusion language models. *arXiv preprint arXiv:2604.00688*, 2026.

A Implementation Details

Table 13: Hyperparameters of WaveGen

Module	Hyperparameter	WaveGen-S	WaveGen-B	WaveGen
Waveform	Sampling Rate	24000	24000	24000
	Reshape	(100, 240)	(100, 240)	(50,480)
	Diffusion Resolution	100Hz	100Hz	50Hz
Spectral Forcing	Hop	240	240	480
	Window	960	960	1920
	FFT	960	960	1920
	Filter	Mel	Mel	Mel
	Bin	64	64	80
	IB	-	-	32
Text Embedding	Dimension	384	384	512
	ConvNeXt	-	-	4
	Conditioning	Average	Average	Filler Token
DiT	Input Dim.	768	1024	1158
	Hidden Dim.	1536	4096	4632
	Shared Layer	14	24	24
	Disentangled Layer	2	2	2
	Head	12	16	16
	Type	Scale-DiT	Scale-DiT	DiT
	AdaLN	AdaLN-SOLA (rank=32)	AdaLN-SOLA (rank=32)	AdaLN-Zero
Training	Training Step	1.1M	1.1M	1.3M
	Epoch	4	4	11
	GPU	4	4	8
	Learning Rate	1×10^{-4}	1×10^{-4}	7.5×10^{-5}
	Warm-up Scheduling	20,000 steps	20,000 steps	20,000 steps
	Learning Scheduling	Cosine Decay	Cosine Decay	Linear Decay
	λ_{sf}	0.1	0.1	0.1
	$\lambda_{\text{v-stft}}$	0.1	0.1	-
	$\lambda_{\text{x-stft}}$	-	-	0.05
	Audio/Token Drop	0.3/0.2	0.3/0.2	0.1 (Joint)
	Model Size	113M	407M	680M
Dataset	Max Frame	480s	480s	384s
	Length	3-30s	3-30s	3-30s
t scheduling	-	Uniform	Uniform	Logit normal (μ=-0.8, σ=0.8)
Data Scaling	Waveform	Max Normalization + x0.95	Max Normalization + x0.95	x9
Noise Scaling	Waveform Noise	0.25	0.25	1
	Spectral Noise	1	1	1
Sampling	ODE Method	Euler	Euler	Euler
	Steps	32	32	50
	CFG	2	2	3
Objective	Prediction Space	x-pred	x-pred	x-pred
	Loss Space	v-loss	v-loss	v-loss
	x to v conversion	$(x_\theta - x_t)/(1-t)$	$(x_\theta - x_t)/(1-t)$	$(x_\theta - x_t)/(1-t)$
	t_{clip}	0.05	0.05	0.02
Multi-scale Spectral v-loss	Hop	[480, 240, 120, 60, 30, 15]	[480, 240, 120, 60, 30, 15]	-
	Window	[1920, 960, 480, 240, 120, 60]	[1920, 960, 480, 240, 120, 60]	-
	FFT	[1920, 960, 480, 240, 120, 60]	[1920, 960, 480, 240, 120, 60]	-
	Filter	Linear	Linear	-
	Bin	[512,256,128,64,32,16]	[512,256,128,64,32,16]	-
Multi-scale Spectral x-loss	Hop	-	-	[512, 256, 128, 64, 32, 16, 8]
	Window	-	-	[2048, 1024, 512, 256, 128, 64, 32]
	FFT	-	-	[2048, 1024, 512, 256, 128, 64, 32]
	Filter	-	-	Mel
	Bin	-	-	[320,160,80,40,20,10,5]

B Implementation Details for Baselines

Spark-TTS Spark-TTS[3] [59] is an LLM-based zero-shot TTS with BiCodec, a single-stream speech codec that represents speech with semantic tokens for linguistic content and global tokens for speaker attributes.

TadiCodec-AR TaDiCodec[4] [60] is a text-aware diffusion speech tokenizer that compresses speech into a low frame rate discrete representation using a diffusion autoencoder with text-guided decoding.

F5-TTS F5-TTS [5] is a non-autoregressive zero-shot TTS model based on flow matching with a Diffusion Transformer. It removes explicit duration modeling and phoneme-level alignment by padding text tokens to the speech length, and uses ConvNeXt-based text refinement to improve alignment robustness. For evaluation, we utilize the official implementation and released checkpoint.[5]

ZipVoice ZipVoice[6] [77] is a compact zero-shot TTS model based on flow matching. It employs a Zipformer-based text encoder and vector-field estimator, uses average upsampling for initial speech-text alignment, and introduces flow distillation to reduce the number of sampling steps.

CosyVoice2 CosyVoice2[7] [13] is an LLM-based TTS model with a chunk-aware causal flow matching decoder. It improves speech token utilization with FSQ and supports both streaming and non-streaming synthesis within a unified framework.

CosyVoice3 CosyVoice3 [14] extends CosyVoice2 toward in-the-wild multilingual zero-shot TTS through larger-scale pre-training, a supervised multi-task speech tokenizer, and RL post-training. In our comparison, CosyVoice3-RL refers to the reward-optimized with differentiable reward optimization. For evaluation, we utilize the official implementation and released checkpoint.[8]

Qwen3-TTS Qwen3-TTS [22] is an LLM-based TTS system built on the Qwen3 backbone. It employs a 12Hz speech tokenizer with a multi-codebook RVQ consisting of 16 codebooks to represent both semantic and acoustic information. To efficiently model the multi-codebook sequence, it adopts multi-token prediction decoding. For evaluation, we utilize the official implementation and evaluate both the 0.6B and 1.7B checkpoints.[9]

MOSS-TTS MOSS-TTS[10] [18] employs the MOSS-Audio-Tokenizer to convert waveform into a unified sequence of discrete representations capturing both semantic and acoustic information.

VibeVoice VibeVoice [46] is a long-form and multi-speaker TTS model designed to generate expressive conversational speech from text. It employs continuous speech tokenizers operating at low frame rate and uses next-token diffusion to autoregressively generate continuous speech representations. They release their open-source implementation[11] and real-time model[12].

VoxCPM VoxCPM[13] [75] uses hierarchical semantic-acoustic modeling, where semantic-prosodic planning and residual acoustic modeling are coupled with a local diffusion decoder for high-fidelity speech generation.

[3] https://github.com/SparkAudio/Spark-TTS
[4] https://github.com/AmphionTeam/TaDiCodec
[5] https://github.com/SWivid/F5-TTS
[6] https://github.com/k2-fsa/ZipVoice
[7] https://github.com/FunAudioLLM/CosyVoice
[8] https://github.com/FunAudioLLM/CosyVoice
[9] https://github.com/QwenLM/Qwen3-TTS
[10] https://github.com/OpenMOSS/MOSS-TTS
[11] https://github.com/microsoft/VibeVoice
[12] https://huggingface.co/microsoft/VibeVoice-Realtime-0.5B
[13] https://github.com/OpenBMB/VoxCPM

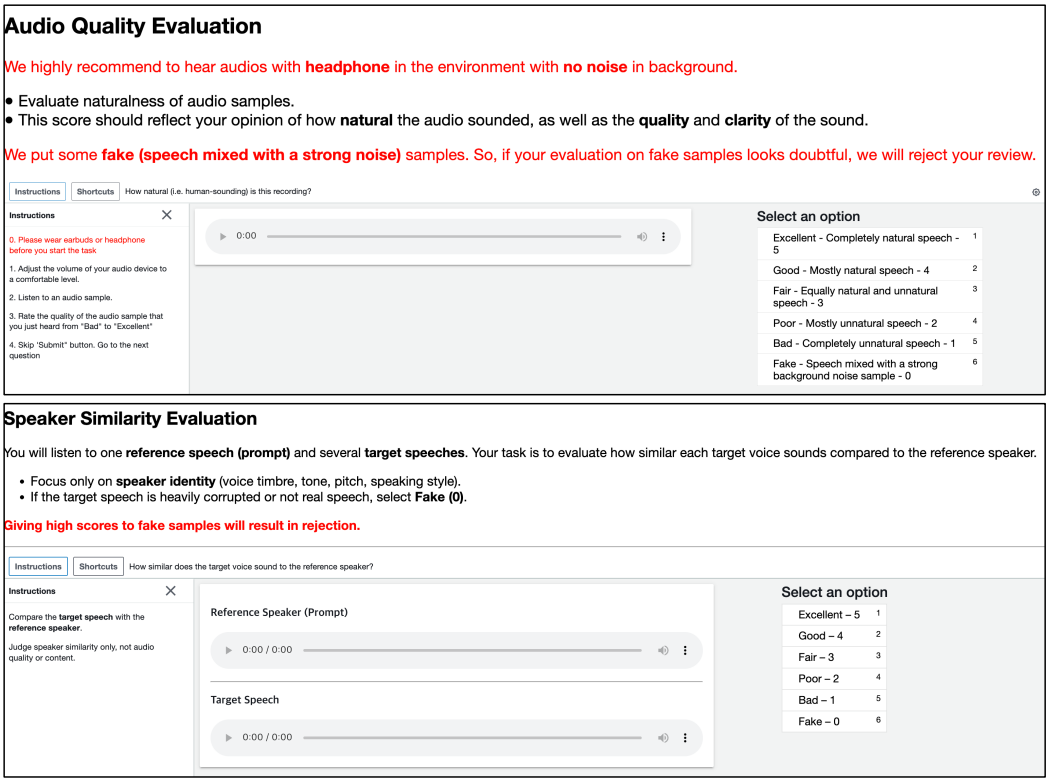


Figure 3: MOS interfaces

C Evaluation Details

WER We calculate word error rate (WER) using open-source automatic speech recognition model, Whisper [50]. We utilize the official implementation of Whisper-large-v3[14] to evaluate speech intelligibility.

SIM For evaluating speaker similarity quality, we utilized WavLM-large which is fine-tuned by speaker verification task[15]. We calculate cosine similarity between prompt and generated samples.

DNSMOS To assess the naturalness of the synthetic samples, we utilized the open-source Mean Opinion Score (MOS) prediction model, DNSMOS[16][51].

D Crowdsourcing Details

We conducted Mean Opinion Score (MOS) evaluations using Amazon Mechanical Turk (MTurk)[17] to qualitatively assess the quality of synthesized speech from the TTS. Specifically, we carried out two types of evaluations: nMOS, which measures the naturalness of the generated speech, and sMOS, which measures the similarity of the generated speaker timbre to that of the prompt speaker. For each evaluation, 20 native speakers of American English participated. Each participant rated 100 samples per model on a 5-point scale ranging from 1 to 5. To reduce unreliable responses, such as participants randomly selecting scores, and to improve the reliability of the MTurk evaluation, we included Gaussian-noise samples as control samples. Responses from participants who failed to identify these samples as fake were excluded from the final analysis. In addition, only MTurk workers with a prior task approval rate of at least 80% and at least 100 approved HITs were allowed to participate in the evaluation.

[14] https://github.com/openai/whisper
[15] https://drive.google.com/file/d/1-aE1NfzpRCLxA4GUxX9ITI3F9LlbtEGP/view.
[16] https://github.com/Lightning-AI/torchmetrics
[17] https://www.mturk.com/

Table 14: Comparison with respect to training steps.

Model	Params.	Data	Training Steps	Basic CFG (2.0) WER(↓)	Basic CFG (2.0) SIM(↑)	Basic CFG (2.0) DNSMOS(↑)	No CFG WER(↓)	No CFG SIM(↑)	No CFG DNSMOS(↑)
WaveGen-S	113M	0.5k	0.3M	104.23	0.32	3.53	129.85	0.25	3.31
WaveGen-S	113M	0.5k	0.5M	26.33	0.40	3.55	92.88	0.27	3.22
WaveGen-S	113M	0.5k	0.7M	8.75	0.45	3.68	76.66	0.33	3.38
WaveGen-S	113M	0.5k	0.9M	6.02	0.47	3.65	58.53	0.35	3.37
WaveGen-S	113M	0.5k	1.1M	4.96	0.47	3.72	56.58	0.36	3.36
WaveGen-B	407M	0.5k	0.3M	28.19	0.42	3.60	93.01	0.31	3.33
WaveGen-B	407M	0.5k	0.5M	9.19	0.49	3.60	76.53	0.33	3.20
WaveGen-B	407M	0.5k	0.7M	4.32	0.51	3.66	51.43	0.39	3.32
WaveGen-B	407M	0.5k	0.9M	3.69	0.52	3.65	44.82	0.39	3.33
WaveGen-B	407M	0.5k	1.1M	2.91	0.53	3.67	41.56	0.40	3.30
WaveGen-S	113M	120k	0.3M	91.23	0.32	3.27	148.32	0.21	3.02
WaveGen-S	113M	120k	0.5M	23.40	0.43	3.47	108.22	0.29	3.17
WaveGen-S	113M	120k	0.7M	8.50	0.45	3.60	77.88	0.33	3.25
WaveGen-S	113M	120k	0.9M	5.00	0.48	3.59	69.43	0.34	3.25
WaveGen-S	113M	120k	1.1M	4.27	0.49	3.60	69.69	0.35	3.28
WaveGen-B	407M	120k	0.3M	63.51	0.44	3.47	143.51	0.29	3.15
WaveGen-B	407M	120k	0.5M	18.37	0.53	3.48	90.07	0.38	3.23
WaveGen-B	407M	120k	0.7M	4.42	0.56	3.58	58.73	0.40	3.22
WaveGen-B	407M	120k	0.9M	2.91	0.57	3.56	60.91	0.42	3.22
WaveGen-B	407M	120k	1.1M	2.37	0.59	3.60	41.91	0.46	3.29

Table 15: Additional Training Results

Model	Params.	Data	Training Steps	Basic CFG (2.0) WER(↓)	Basic CFG (2.0) SIM(↑)	Basic CFG (2.0) DNSMOS(↑)	No CFG WER(↓)	No CFG SIM(↑)	No CFG DNSMOS(↑)
WaveGen-B	407M	120k	1.1M	2.37	0.59	3.60	41.91	0.46	3.29
WaveGen-B	407M	120k	1.2M	1.98	0.59	3.61	39.67	0.46	3.31
WaveGen-B	407M	120k	1.3M	1.85	0.60	3.60	36.67	0.46	3.30

Table 16: CFG results

Model	Params.	Data	Training Steps	CFG	Modified CFG (2.0) WER(↓)	Modified CFG (2.0) SIM(↑)	Modified CFG (2.0) DNSMOS(↑)
WaveGen-B	407M	120k	1.1M	0.5	4.26	0.605	3.55
WaveGen-B	407M	120k	1.1M	1	2.61	0.625	3.60
WaveGen-B	407M	120k	1.1M	2	1.77	0.635	3.60
WaveGen-B	407M	120k	1.1M	3	1.93	0.637	3.52
WaveGen-B	407M	120k	1.1M	4	1.87	0.633	3.37

E Additional results

Table 14 shows the objective evaluation results of each model with respect to the number of training steps. The results indicate that model performance consistently improves as the number of training steps increases.

Due to limited computational resources, we trained the model only up to 1.1M steps, corresponding to four epochs on the Emilia dataset. Nevertheless, WaveGen-B continued to improve with additional training steps, as shown in Table 15. These results suggest that end-to-end TTS models have strong potential for further improvement with larger-scale training.

Table 16 shows the effect of CFG on the WaveGen-B model. Increasing the CFG scale substantially improves intelligibility, reducing WER from 4.26 to 1.77 when the CFG scale is increased from 0.5 to 2.0. Speaker similarity also improves consistently up to CFG 3.0, indicating that stronger conditioning helps the model better preserve the target speaker characteristics. However, excessively large CFG values slightly degrade perceptual quality, as reflected by the decrease in DNSMOS at CFG 3.0 and 4.0. These results show that CFG provides an effective trade-off between intelligibility, speaker similarity, and audio quality, with CFG 2.0 achieving the best overall balance in our experiments.

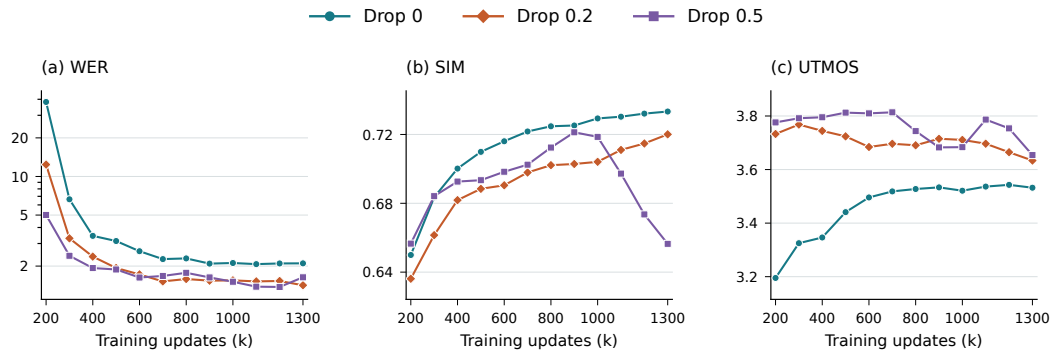


Figure 4: Spectral Dropout

F Multilingual WaveGen

We conducted multilingual text-to-speech comparison using Emilia (EN and ZH) datasets. For a fair comparison, we utilize the same architecture of F5-TTS to demonstrate the effectiveness of waveform-level diffusion training and Spectral Forcing. Then, we adopt the time scheduling methods of WavTTS to compare the effect of Spectral Forcing. For efficient waveform generation, we employ 50Hz resolutions using hop size of 480 for 24 kHz waveform diffusion.

F.1 Spectral Loss

Basically, we introduce multi-scale spectral v-loss for our models. However, we adopt multi-scale Mel-spectrogram x-loss to compare Spectral Forcing based on WavTTS implementation. We observe that multi-scale Mel-spectrogram loss could improve the initial alignment learning and entire objective performances including WER and SIM. However, this loss induces buzzing sound on the generated waveform. To address this perceptual limitation, we released the perceptually-improved models together using multi-scale spectral v-loss which could significantly improve the perceptual quality. While it shows lower performance in objective metrics, we highly recommend adopting our new objectives for perceptual quality. We released the model finetuned with multi-scale spectral v-loss for perceptually improved speech synthesis.

F.2 Spectral Dropout

We further investigated the use of spectral dropout in spectral forcing to improve robustness to noise in multilingual datasets. Figure 4 presents the objective evaluation metrics for different spectral dropout rates. The results suggest that higher dropout rates can improve initial alignment when training on noisy waveforms, but may also cause the spectral path to collapse during training. Without spectral dropout, the model tends to rely on the noisy Mel-spectrogram for waveform reconstruction. We therefore selected a dropout rate of 0.2 to reduce this reliance while maintaining the stability of the spectral path.

F.3 Sampling Steps

With large-scale training, Table 17 shows that the model achieves better performance across all evaluated NFEs than that reported in Table 11. Furthermore, we plan to release an accelerated version of WaveGen that enables efficient speech synthesis in only a few sampling steps.

Table 17: Analysis of NFE (Seed-en, WaveGen)

Model	NFE	WER	SIM	UTMOS
WaveGen	1	93.10	0.19	1.30
	2	23.71	0.42	1.31
	4	2.76	0.64	2.34
	8	1.70	0.75	3.44
	10	1.72	0.75	3.53
	16	1.69	0.76	3.65
	32	1.50	0.75	3.68
	50	1.47	0.75	3.68

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize our contribution as the first end-to-end diffusion-based text-to-waveform TTS framework.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe the limitations of our work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not present new theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe all architecture details in the main manuscript and appendix with all hyperparameters and dataset details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We utilized open-source datasets including LibriTTS and Emilia datasets, and we released our source code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the dataset details and hyperparameters in Section 4 and Appendix. Also, we utilize public-available benchmark datasets for evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported MOS with 95% CI together.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information on the computer resources in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: All authors reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss them in Section 6

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We released the source code and checkpoint with detailed safeguards instructions together.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We described the license of LibriTTS and Emilia dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the code and model with GitHub and Huggingface documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We describe the details of MOS experiments with screenshots in Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We conduct our experiments via Amazon Mturk service.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We used LLMs only for grammar correction and polishing.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.