VOICESTAR: Robust Zero-Shot Autoregressive TTS with Duration Control and Extrapolation

Puyuan Peng[1], **Zhisheng Zheng**[1], **Shang-Wen Li**,
Abdelrahman Mohamed[2], **David Harwath**[1]
[1] The University of Texas at Austin, [2] Rembrand

Abstract

We present VOICESTAR, the first zero-shot TTS model that achieves both output duration control and extrapolation. VOICESTAR is an autoregressive encoder-decoder neural codec language model, that leverages a novel Progress-Monitoring Rotary Position Embedding (PM-ROPE) and is trained with Continuation-Prompt Mixed (CPM) training. PM-ROPE enables the model to better align text and speech tokens, indicates the target duration for the generated speech, and also allows the model to generate speech waveforms much longer in duration than those seen during training. CPM training also helps to mitigate the training/inference mismatch, and significantly improves the quality of the generated speech in terms of speaker similarity and intelligibility. VOICESTAR outperforms or is on par with current state-of-the-art models on short-form benchmarks such as Librispeech and Seed-TTS, and significantly outperforms these models on long-form/extrapolation benchmarks (20-50s) in terms of intelligibility and naturalness. Code and model: github.com/jasonppy/VoiceStar. Audio samples: jasonppy.github.io/VoiceStar_web.

1 Introduction

Neural codec language models (NCLMs) have rapidly become a state-of-the-art method for text-to-speech (TTS) generation. These models use neural network audio codecs (Zeghidour et al., 2021; Defossez et al., 2022) to tokenize speech waveforms into sequences of discrete symbols representing temporal frames. Next, a Transformer (Vaswani et al., 2017) language model is used to autoregressively model these token sequences. The success of this approach is due to combination of the modeling power of Transformer models and the ease of reconstructing high-fidelity waveforms from the generated token sequences. However, current NCLM-based TTS models fall short in several important ways, specifically their lack of fine-grained controllability (especially for duration control) and their inability to extrapolate to sequence lengths much longer than those seen during training.

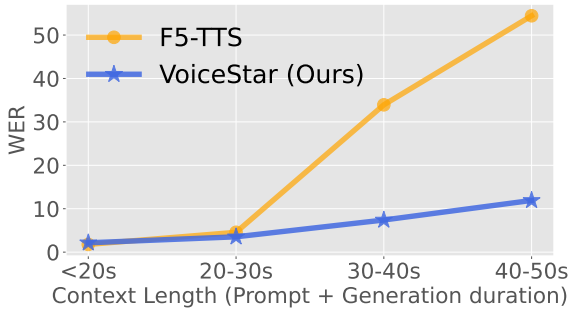


Figure 1: WER comparison between our VOICESTAR and F5-TTS (Chen et al., 2024c) under different context lengths. Both models are trained with maximal context length of 30 seconds.

In this paper, we propose solutions for these problems, and also propose several other novel techniques for improving the quality of speech generated by NCLM TTS models.

Specifically, we note that current NCLM-based TTS models (Wang et al., 2023a; Chen et al., 2024b; Peng et al., 2024) do not explicitly model the alignment between their input text and speech sequences, and instead simply concatenate these sequences and put the onus on the model to learn the alignment via standard positional encodings. Furthermore, standard positional encodings do not easily enable a user to specify the desired sequence length of a generation at inference time.

To fix these flaws, we propose to use an encoder-decoder architecture with a novel Progress-Monitoring Rotary Position Embedding (PM-ROPE). This provides the model with a form of flat-start alignment between text and speech tokens from the very beginning of training. The PM-ROPE embeddings also encode the desired sequence length for the generated speech, which implicitly informs the model at each timestep how

Findings of the Association for Computational Linguistics: ACL 2026, pages 11737–11754

far along the generation has progressed relative to this target length. Furthermore, we note that existing NCLM-based TTS models treat voice cloning TTS as speech continuation, which at test time can entangle the reference speaker's voice characteristics with the prosody present in the reference utterance. At inference, we may only wish to clone the speaker's vocal characteristics and let the model infer what the prosody or emotional delivery should be conditioned on the input text. To encourage this disentanglement, during training we perform random prompt mixing: sometimes the model is trained to perform continuation of an utterance, but other times we sample a random (different) utterance from the same speaker to serve as the reference prompt. We call this method continuation-prompt mixed training (CPM training). The use of CPM training additionally allows us to apply data augmentation techniques, such as speed perturbation, which we show improves intelligibility.

To summarize, our contributions are as follows:

1. We propose Progress-Monitoring Rotary Position Embeddings, or PM-RoPE, that leads to robust and duration controllable NCLM-based TTS, which further unlocks the capability to generate utterances much longer than those seen during training;
2. We propose continuation-prompt mixed training, or CPM training, which improves the intelligibility and naturalness of NCLM-based TTS models;
3. We propose two additional techniques that positively impact the performance of NCLM-based TTS models: 1) prompt speed perturbation during training; 2) prompt repetition during inference.

Combining the proposed techniques, **our model, VoiceStar, is the first zero-shot TTS model with duration control and length extrapolation capabilities**. VoiceStar achieves performance better than or on par with other state-of-the-art models on existing short-form benchmarks including Seed-TTS-eval and Librispeech, and significantly outperforms current SotA models on long-form/extrapolation benchmarks.

2 Related Work

Neural Codec Language Models Pioneered by Lakhotia et al. (2021); Borsos et al. (2022); Kreuk et al. (2022); Wang et al. (2023a); Kharitonov et al. (2023); Borsos et al. (2023), NCLMs have become one of two state-of-the-art approaches for audio generation, the other being flow-matching/diffusion models.

Model	Paradigm	Open Source	Voice Cloning	Duration Control	Extrapola.
VALL-E (Wang et al., 2023a)	AR+NAR		✓		
Voicebox (Le et al., 2023)	NAR		✓	✓	
MaskGCT (Wang et al., 2024c)	NAR	✓	✓	✓	
F5-TTS (Chen et al., 2024c)	NAR	✓	✓	✓	
VAT (Battenberg et al., 2024)	AR				✓
VoiceCraft (Peng et al., 2024)	AR	✓	✓		
CosyVoice (Du et al., 2024c)	AR	✓	✓		
Llasa (Ye et al., 2025)	AR	✓	✓		
VoiceStar (Ours)	AR	✓	✓	✓	✓

Table 1: Conceptual comparison of VoiceStar with other models. AR stands for autoregressive and NAR stands for non-autoregressive. Extrapola. stands for extrapolation.

Due to the strong in-context learning ability of Transformer LMs, NCLMs are a particularly effective approach for zero-shot voice-cloning TTS (Wang et al., 2023a). In this setting, a model must clone the vocal characteristics of a reference speaker that was unseen during training, using several seconds of reference speech provided as a prompt at inference time.

Enhancing the Robustness of NCLM. Despite their typically strong performance, NCLMs are known to have robustness issues, such as skipping words, inserting extra words, repeating words, and inserting unnaturally long silences (Wang et al., 2023a; Peng et al., 2024). Several papers have proposed to address these issues from the angle of text-speech alignment. Specifically, (Song et al., 2024; Han et al., 2024) use an external forced alignment model to tightly couple the text prompt and generated speech, while (Du et al., 2024a) makes use of a transducer architecture to implicitly learn text-speech alignment. Wang et al. (2024a) uses a constrained attention mechanism to enforce monotonic text-speech alignment. T5-TTS (Neekhara et al., 2024) loads weights from a textual encoder-decoder T5 model and introduces an auxiliary loss to encourage monotonic alignment in the text-speech cross-attention weights. Also, unlike speech-continuation training, T5-TTS uses a separate utterance drawn from the same speaker as the prompt during training. This differs from our proposed CPM training in two ways: 1) in addition to the reference speech, we also append the transcript to the text that is to be generated; 2) we stochastically mix these same-speaker-different-utterance prompts with utterance continuation-based prompt-

ing, which we show in our ablation experiments to be better than either style of prompting on its own. VAT (Battenberg et al., 2024) also uses an encoder-decoder architecture, and proposes a T5-like relative position embedding to enhance the text-speech alignment. Notably, VAT also achieves extrapolation similarly to our model, however it cannot simultaneously control the duration of the output as our model can. Another crucial difference between VoiceStar and VAT is that VAT is a conventional multi-talker TTS system that can only generate speech in the voices of speakers seen during training, and is not capable of zero-shot voice-cloning TTS.

Note that in addition to improving the text-speech alignment in NCLMs, there are other works that have tried other methods to improve robustness, such as using multi-scale generation to address recency bias (Guo et al., 2024b), enforcing disentanglement of speech attributes (Jiang et al., 2024), incorporating classifier-free guidance (Wang et al., 2024b), using chain-of-thought prompting (Xin et al., 2024), and reinforcement learning from human/AI feedback (Chen et al., 2024a; Hu et al., 2024; Hussain et al., 2025).

Adding new capabilities to NCLMs. Zhang et al. (2023); Zhu et al. (2024); Zheng et al. (2025) extended NCLM-based TTS to the multilingual case. Wang et al. (2023c,b); Maiti et al. (2023); Wu et al. (2024); Wang et al. (2025b) propose multi-task learning for generation and recognition tasks. Guo et al. (2022); Yang et al. (2023); Liu et al. (2023); Ji et al. (2023); Leng et al. (2023); Lyth and King (2024); Diwan et al. (2025) adapt NCLMs to style-controlled speech synthesis. D'efossez et al. (2024); Ma et al. (2024); Fang et al. (2024); Yu et al. (2024); Xu et al. (2024); Zhang et al. (2024, 2025) propose NCLM-based models for real-time interactive voice assistants. Lajszczak et al. (2024); Wang et al. (2024c); Ye et al. (2025) investigate scaling up NCLM-based models. Nishimura et al. (2024) also focuses on long-form generation, but our work differs from theirs in that (Nishimura et al., 2024) specifically trains on long-form speech, while our model is trained on short-form speech only, yet we show that it can generalize to long-form speech.

Diffusion/flow-based TTS and Scaling TTS models. Diffusion/flow-based models represent another popular approach to zero-shot TTS. With masked reconstruction training, Shen et al. (2023); Le et al. (2023); Kim et al. (2023) show that diffusion/flow-based models can achieve SotA performance in zero-shot TTS. Li et al. (2024) distills efficient diffusion models via direct metric optimization. E2-TTS (Eskimez et al., 2024) simplifies the model pipeline by removing phoneme duration prediction and grapheme-to-phoneme modules. F5-TTS (Chen et al., 2024c) scales E2-TTS to 100k hours of bilingual (English and Mandarin) speech. Vyas et al. (2023) also scales a flow-based model to allow for a variety of capabilities such as style control and general audio generation. Finally, Du et al. (2024b); Guo et al. (2024a); Du et al. (2024c) proposed hybrid models that combine NCLM and flow-matching.

3 Method

The architecture of VOICESTAR along with a typical decoder-only, zero-shot TTS architecture are shown in Fig. 2. VOICESTAR adopts an encoder-decoder architecture, where the encoder takes as input IPA phonemes derived from the text transcript, and the decoder autoregressively predicts speech tokens produced by an Encodec model (Defossez et al., 2022; Peng et al., 2024). Text and speech inputs are separated with special tokens to indicate reference and target, and are positioned using PM-ROPE, enabling controllable, zero-shot TTS with duration extrapolation capability. The following two sections describe the key differences between our model and prior works in detail.

3.1 Progress-Monitoring RoPE

VOICESTAR builds upon the Rotary Position Embedding (RoPE) (Su et al., 2023). For a detailed mathematical review of standard dot-product attention and the original RoPE derivation, please refer to Appendix A.1.

Progress-Monitoring RoPE (PM-ROPE). While RoPE has been widely used in textual decoder-only LLMs to encode absolute positions, speech-text LMs utilize two distinct input sequences which often are implicitly aligned. We therefore propose a novel extension to RoPE, namely Progress-Monitoring RoPE (PM-ROPE). When combined with an encoder-decoder NCLM, PM-ROPE brings three benefits: 1) improved text-speech alignment, 2) duration control during generation, and 3) test-time extrapolation.

Standard RoPE calculates the query (f_q) and key (f_k) vectors by applying a rotation matrix R based on absolute positions t and s. PM-ROPE introduces a simple change to f_k and f_q by monitoring

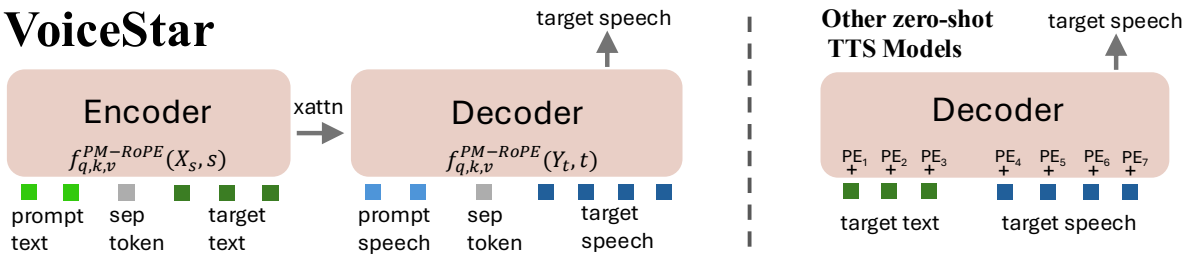


Figure 2: Left: The architecture of VOICESTAR. Right: the common general architecture for zero-shot TTS models, such as VALL-E (AR part), VoiceCraft, CosyVoice, Llasa etc. VOICESTAR differs from them in three aspects: 1) it uses an encoder-decoder architecture with PM-RoPE to provide text-speech alignment, duration control, and extrapolation capability; 2) it uses prompt-continuation mixed training to mitigate the training/inference mismatch and enhance robustness. Our speech tokenizer based on Encodec (Defossez et al., 2022) uses multiple codebooks and we apply the delay pattern (Copet et al., 2023) to them, which is not depicted in this figure for simplicity.

fractional progress rather than absolute position:

$$f_k^{\text{PM-RoPE}}(X_s, s) = R\left(\frac{s}{S}N\theta\right) W_k X_s,$$
$$f_q^{\text{PM-RoPE}}(Y_t, t) = R\left(\frac{t}{T}N\theta\right) W_q Y_t$$

where W_k and W_q are learnable projection matrices, and $R(\cdot)$ is the rotation matrix. Instead of using *positions* s and t, we measure the *fractional progress* of s towards some maximum value S, and the progress of t towards some maximum value T, i.e. $\frac{s}{S}$ and $\frac{t}{T}$. Specifically, S is the total sequence length of the source tokens, and T is the (desired) total sequence length of the target tokens. N is a hyperparameter that rescales the rotation angle, which can also be interpreted as the *pseudo* total sequence length. The inner product between a key vector and a value vector becomes:

$$\langle f_q^{\text{PM-RoPE}}(Y_t, t), f_k^{\text{PM-RoPE}}(X_s, s)\rangle = Y_t^\top W_q^\top R\left[\left(\frac{t}{T} - \frac{s}{S}\right) N\theta\right] W_k X_s$$

Note that the inner product is a function of Y_t, X_s and their *relative progress* $\frac{t}{T} - \frac{s}{S}$. Similar to ROPE, PM-ROPE also enjoys the long-term decay property, i.e. the inner product between two tokens decreases as the difference between their progresses increases.

In our encoder-decoder NCLM (the left hand side in Fig. 2), the encoder takes as input a phonemized text transcript, and the decoder predicts acoustic tokens in an autoregressive fashion. We treat the phonetic tokens as the source tokens X, and the acoustic tokens as the target tokens Y. When PM-ROPE is applied to the decoder's self-attention mechanism, it provides information about the location of the current acoustic token within the overall target sequence length, which enables output duration control - during autoregressive generation,

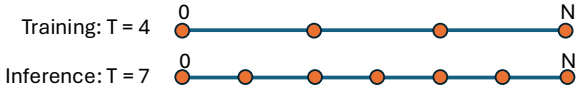


Figure 3: An example on how PM-ROPE turns extrapolation into interpolation: during training, the maximal training sequence length is 4, and during inference the target length is 7. The positional encodings for both can be expressed as sampling points inside the same interval $[0, N]$.

when the progress of the target token sequence reaches 100% (i.e. $\frac{t}{T} = 1$, where the value of T is specified in advance by the user), the model should learn to emit an end-of-generation token. When PM-ROPE is applied to the cross-attention connecting the encoder to the decoder (acoustic tokens attending to phonetic tokens), it provides information about the temporal alignment between the two modalities. This is helpful during training because it provides a flat-start initial alignment to the untrained model, since an acoustic token's cross-attention will be concentrated at the phonetic tokens occupying the same relative position within the phonetic sequence, i.e. where the difference $\frac{t}{T} - \frac{s}{S}$ is small. See Appendix A.3 for more details.

Applying PM-ROPE to the phonetic input is essential for extrapolation, as it unifies the encoder and decoder under a fixed pseudo length N. Consequently, longer sequences can be processed by sampling more points within $[0, N]$ via interpolation (see Fig.3).

3.2 Continuation-Prompt Mixed Training

Although framing zero-shot TTS as speech continuation effectively leverages the strong in-context learning capabilities of LLM and thus achieves high speaker similarity, it also introduces a degree of training/inference mismatch. During training, the emotion, semantics, prosody, and speaking

style are usually consistent within an utterance, and therefore will be consistent between the reference speech and the generation target. However, during inference the semantics or emotion expressed by the reference utterance may be very different than those implicitly expressed in the target transcript input by a user. In this case, the model must be able to adjust the prosody and emotional delivery to match the target transcript, even if they do not match those found in the reference utterance. We propose a simple method to mitigate this mismatch: continuation-prompt mixed (CPM) training (left hand side of Fig. 2). During each training iteration, with some probability p we randomly sample two different utterances from the same speaker and use one as the reference utterance and one as the target. With probability $1 - p$ we use standard speech continuation based-training, i.e. we only sample one utterance and mask it, using the unmasked portion of the utterance as the reference and the masked portion of the utterance as the target. When using two different utterances we also have the opportunity to perform data augmentation to one but not the other. Specifically, we apply speed perturbation to the reference utterance with probability p' and speed factor $\delta = 0.25$. This randomly speeds up or slows down the reference speech, keeping its total duration within $1 \pm \delta$ of its original duration. When a prompt is selected, we use learnable `sep_tokens` to separate the prompt and target, and only calculate the loss on target tokens.

Prompt repetition for improving speaker similarity. Another benefit of CPM training is that it allows us to improve speaker similarity between the generated speech and the reference speech by simply repeating the reference utterance and its transcript multiple times, without harming intelligibility. Counterintuitive as it might be, since no additional information is provided by repeating the prompt, we hypothesize that repeating the reference utterance increases the influence of the ground-truth speech tokens when performing the weighted sum over tokens within attention layers, biasing the model to attend more to the reference speech tokens than its previous generations, resulting in generated speech that is more similar to the reference prompt. As we quantitatively show in our experiments (Fig. 7b), when a model is trained solely on speech continuation, reference prompt repetition degrades the intelligibility (WER) of the generated speech because it biases the model to exhibit a repetitive style. On the other hand, when using CPM training this issue is mitigated because 1) the model has been trained on examples where the reference speech and target speech express different styles and 2) `sep_tokens` explicitly separates the reference and target utterances, makes it easier to align the text and the corresponding speech.

4 Experiments

4.1 Setup

Training data. Our training set consists of the English portion of Emilia (He et al., 2024) and a subset of the training splits of Libriheavy and Libriheavy-long (Kang et al., 2023), totaling 65K hours. Emilia is a multilingual dataset comprised of in-the-wild speech of diverse styles. Its English portion contains 46K hours of speech, and after filtering out low quality data following Chen et al. (2024c), approximately 40K hours remain with a maximum utterance duration of 30 seconds. Libriheavy is an automatically transcribed version of the LibriLight audiobook corpus. The original release of Libriheavy contains 50k hours of speech, and we also make use of the recently released Libriheavy-long dataset, which is a re-segmented version that has longer segments ranging from 20 seconds to 100 seconds, totaling 42K hours. Note that in order to fairly compare with other models on extrapolation, we only train our models on utterances with duration less than or equal to 30 seconds. We randomly sampled 25K hours of speech from Libriheavy. Crucially, both Emilia and Libriheavy contain speaker information (although speaker information in Emilia is provided using automatic speaker diarization, we found it to be very reliable), and therefore we are able to sample multiple utterances from the same speaker for CPM training. We phonemize all text transcripts into the IPA phoneme set using espeak-ng (Duddington et al.).

Evaluation tasks and data. We evaluate our models on two English-language, zero-shot TTS tasks: short-form TTS and long-form TTS. For short-form TTS ($\leq$20s): we use Seed-TTS eval set (Anastassiou et al., 2024) and Librispeech-PC (Chen et al., 2024c), each containing around 1000 prompt-target pairs with prompts and targets $<$10s. For Long-form TTS, we consider three duration ranges (20-30s, 30-40s, and 40s-50s) and source the evaluation data from the libriheavy test and validation sets (we use validation set due to the scarcity of long-form speech samples in the test set. We thus avoid using the validation set to do any hyperpa-

rameter tuning or early stopping of model training). We sample 1000 prompt target pairs for the 20-30s range, 500 pairs for the 30-40s range, and 100 pairs for the 40-50s duration range. For our ablation studies, we sample a 1000 utterance evaluation set from the Libriheavy validation set with lengths shorter than 20 seconds for short-form ablations (Tab. 2), and lengths between 20 and 30 seconds for extrapolation (Tab. 3, no overlap with the test-sets). For duration specification with the $\leq$20s test sets, following Chen et al. (2024c) and estimate the duration of the target sequence by calculating the speaking rate of the reference prompt (in terms of seconds-per-character) and multiplying it by the character length of the target text; for ablations and long-form TTS, we use the ground truth duration because the estimated duration sometimes falls outside of the target range (for example, when testing model performance on 40-50s speech, the estimated duration could be only 35 seconds), and we only compare against models that can also control the duration of their generations. We also show in Appendix B.3 our model's performance on long-form TTS when using the automatically estimated durations.

Model, training, inference, and baselines. Our main model has 840 million parameters, composed of 12 standard Transformer encoder layers and 40 Transformer decoder layers, where the dot-product attention mechanism is replaced by our proposed PM-RoPE. The hidden dimensions for both the encoder and decoder are 1024, and the model has 16 attention heads. For our ablation studies, we train 230 million parameter models with a hidden states dimension of 768, composed of 10 encoder layers and 16 decoder layers. The decoder-only model used in our ablation studies has 16 layers with a hidden dimension of 1024. We use the 4-codebook 50Hz Encodec model released by VoiceCraft (Peng et al., 2024) as our speech tokenizer. The pseudo sequence length N is set to 2000 for all models where PM-RoPEis used. Models are trained with ScaledAdam (Yao et al., 2024) with a maximum batch size of 0.3(ablation)/1.78(main) hours of audio. The base learning rate is 0.03 and scheduled by the Eden scheduler (Yao et al., 2024). All models are trained for 50k steps with codebook loss weights of {5, 1, 0.5, 0.1}, similarly to Peng et al. (2024). Our main model is further trained for an additional 18k steps with codebook weights {2.5, 2, 1.5, 0.6}, as we found it to improve both intelligibility and speaker similarity metrics. Our main model is trained for 8 days on 8 L40 and 16 GH200 GPUs. The main model is trained on the entire 65k hour training set with maximal context length of 30 seconds, and the ablation models are trained on a 16k hour subset with maximal context length of 20 seconds. We use top-k sampling during inference with $k = 10$. For our baselines, we compare against VoiceCraft (Peng et al., 2024), FireRedTTS (Guo et al., 2024a), CosyVoice (Du et al., 2024b), MaskGCT (Chen et al., 2024c), F5-TTS (Chen et al., 2024c), CosyVoice2 (Du et al., 2024c), and Llasa (Ye et al., 2025). We additionally report in Appendix B.2 the performance of closed source models: Seed-TTS (Anastassiou et al., 2024), and concurrent work Metis (Wang et al., 2025b) and SparkTTS (Wang et al., 2025a).

Evaluation Metrics. We use word error rate (WER) and speaker similarity (SpkSim) as automatic evaluation metrics following prior works. For our ablation studies, we additionally use UTMOS (Saeki et al., 2022) to measure naturalness and DurDiff defined as the absolute difference between target generation duration and actual duration, to measure duration controllability. For comparison with other SoTA models on short-form TTS, we follow Seed-TTS eval setup and use Whisper-v3 (Radford et al., 2022) for ASR and the WavLM speaker verification model (Chen et al., 2021) for SpkSim. For long-form TTS, we used Whisper Large-v3-turbo for ASR as we found it to be both more accurate and faster. We additionally conduct subjective human evaluation using Amazon Mechanical Turk, focusing on three aspects: intelligibility, naturalness, and speaker similarity. For $\leq$20s testsets, since most models performance similarly, we adapt the Comparative MOS protocol from (Loizou, 2011), where we compare each generated sample with ground truth target sample on a 7-point Likert scale (-3 to 3). For 20-50s testsets, we use the regular MOS protocol, where we ask humans to rate each generated sample on a 5-point Likert scale (1 to 5). For each sample, we collect 10 human ratings for CMOS and 5 for MOS. To facilitate a fairer comparison, we also resample all speech waveforms to 16 kHz.

4.2 Ablations

Tab. 2 shows how each component of our model contributes to its overall performance. We first compare the performance of the decoder-only model with sinusoidal position embedding (PE) and the encoder-decoder model with RoPE. The encoder-

Architecture	PE	Training	WER ↓	UTMOS ↑	SpkSim ↑	DurDiff ↓
Dec	Sinusoid	Continuation	11.04	2.767	0.537	1.245
Enc-Dec	RoPE	Continuation	8.82	3.301	0.592	1.812
Enc-Dec	PM-RoPE	Continuation	8.49	3.337	0.601	0.009
Enc-Dec	PM-RoPE	CPM	6.42	3.365	0.587	0.009
Enc-Dec	PM-RoPE	CPM+SA	5.66	3.345	0.583	0.009

Table 2: Ablation studies on enc-dec architecture, PM-RoPE, and CPM training. CPM stands for continuation-prompt mixed training. SA stands for speech augmentation on prompt during training.

decoder model outperforms the decoder-only in all metrics except for DurDiff. But when RoPE is replaced by PM-RoPE(line 3 v.s. line 4), we see DurDiff drops from 1.812 to 0.009, which is a lower bound for this metric under our setup, because the codec token resolution is 0.02s per token. This illustrates the dramatic effectiveness of PM-RoPE in enabling precise duration control. We also see a slight improvement on all other metrics, indicating the benefit of the improved text-speech alignment introduced by PM-RoPE. Lastly, we see further improvement when CPM and speed augmentation is applied. Fig. 4 further shows the impact of different probability of using prompt as prefix and speed augmentation on prompt during training. Tab. 3 shows the benefit of PM-RoPE on test time extrapolation, where the models trained on a maximal 20 seconds context length (prompt + target speech duration) are tested on 20 to 30 seconds context length. We see that to ensure extrapolation, it is necessary to apply PM-RoPE on both the encoder and the decoder, additionally we see that PM-RoPE improves the extrapolation WER from 11.46 to 6.75 compared to RoPE.

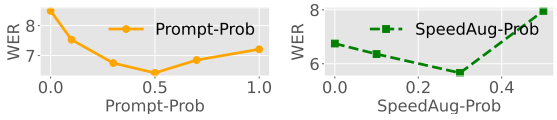


Figure 4: Effects of prompt-prob and speed augmentation for CPM training.

Enc PE	Dec PE	WER↓	SpkSim ↑
RoPE	RoPE	11.46	0.630
RoPE	PM-RoPE	28.67	0.605
PM-RoPE	PM-RoPE	6.75	0.640

Table 3: Extrapolation capabilities of RoPE and PM-RoPE. Max training context length: 20s, test: 20s - 30s.

Fig. 5 shows the impact of prompt repetition on generation quality, measured by WER (left in red) and SpkSim (right in blue). The red and blue dotted horizontal lines indicate the performance when we repeat prompt for each sample so that the context length (i.e. prompt + generation length) reaches the maximum seen during training, i.e. 20 seconds.

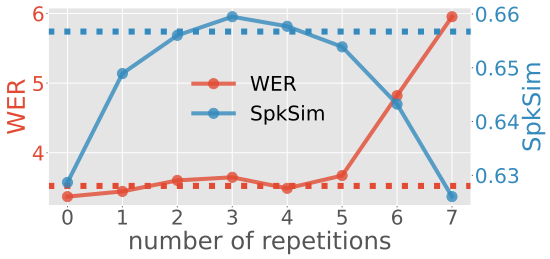


Figure 5: Ablation on the impact prompt repetition on generation quality measured by WER and SpkSim. The red and blue horizontal dotted lines are the performance when we repeat prompt so that the context length reach the maximal training duration.

We see that as the number of repetitions increases from 0 to 3, WER sees very little degradation, while SpkSim improves from around 0.63 to 0.66. But as we further increase the number of repetitions, both WER and SpkSim start to degrade. In our main experiments on short-form TTS, we choose to repeat each sample until the context length reaches the maximal training duration (i.e. the approach that produces the dotted lines in Fig. 5). On long-form TTS, we repeat the prompt once for the 20-30s range and do not apply prompt repetition to longer ranges. To test whether prompt repetition can also improve the performance of models trained with speech continuation, we tested F5-TTS on the same dataset using the same prompt repetition technique.

We found that for F5-TTS, prompt repetition has a significantly negative impact on intelligibility: repeating prompt to reach the maximum duration seen during training duration degrades WER from 3.1 to 75.4. Further experiments on the impact of prompt repetition can be found in Appendix B.1.

4.3 Short-form TTS

As shown in Tab. 4, on Librispeech-PC (Chen et al., 2024c), VoiceStar outperforms other models during human evaluation on both naturalness

(N-CMOS) and speaker similarity (S-CMOS). Notably, VOICESTAR (along with several other models) even outperforms the ground truth speech. Anecdotally, we hypothesize that this is due to the fact that audiobook speech is highly regular and sometimes monotonic, but since our model is also trained on in-the-wild conversational data, it tends to generate speech with a more natural flow. On Seed-TTS (en) (Anastassiou et al., 2024), VOICESTAR is slightly worse than the best performing models Llasa 1B and MaskGCT. However, we find that the automatic WER and SpkSim metrics are not perfectly correlated with human judgements: for example, Llasa 1B achieves a SpkSim of 0.58, which is lower than VOICESTAR, but humans rate it to have a higher speaker similarity than VOICESTAR. Overall, most models perform similarly on the two test sets and are often rated as equal or even better than ground truth.

Model	WER	SpkSim	S-CMOS	N-CMOS
LibriSpeech-PC				
Ground Truth	2.23	0.69	0.00	0.00
FireRedTTS (Guo et al., 2024a)	7.73	0.45	-0.48$_{\pm 0.22}$	-0.89$_{\pm 0.21}$
Llasa 1B (Ye et al., 2025)	4.28	0.47	0.27$_{\pm 0.19}$	0.13$_{\pm 0.20}$
CosyVoice (Du et al., 2024b)	3.40	0.66	0.37$_{\pm 0.19}$	0.07$_{\pm 0.21}$
VoiceCraft (Peng et al., 2024)	3.09	0.52	0.15$_{\pm 0.20}$	-0.37$_{\pm 0.20}$
CosyVoice2 (Du et al., 2024c)	2.66	0.66	0.40$_{\pm 0.19}$	0.06$_{\pm 0.21}$
MaskGCT (Wang et al., 2024c)	2.66	0.66	0.21$_{\pm 0.20}$	-0.25$_{\pm 0.21}$
F5-TTS (Chen et al., 2024c)	2.57	0.65	0.12$_{\pm 0.20}$	-0.43$_{\pm 0.21}$
VOICESTAR	2.64	0.63	**0.60**$_{\pm \mathbf{0.19}}$	**0.18**$_{\pm \mathbf{0.21}}$
Seed-TTS (en)				
Ground Truth	2.15	0.73	0.00	0.00
FireRedTTS (Guo et al., 2024a)	9.09	0.45	-1.24$_{\pm 0.16}$	-0.92$_{\pm 0.18}$
Llasa 1B (Ye et al., 2025)	5.13	0.58	-0.23$_{\pm 0.14}$	**0.05**$_{\pm \mathbf{0.16}}$
CosyVoice (Du et al., 2024b)	4.20	0.64	-0.42$_{\pm 0.16}$	-0.36$_{\pm 0.18}$
VoiceCraft (Peng et al., 2024)	3.45	0.52	-0.84$_{\pm 0.19}$	-0.73$_{\pm 0.18}$
CosyVoice2 (Du et al., 2024c)	2.69	0.66	**-0.11**$_{\pm \mathbf{0.16}}$	-0.19$_{\pm 0.15}$
MaskGCT (Wang et al., 2024c)	2.52	0.71	-0.13$_{\pm 0.15}$	-0.14$_{\pm 0.15}$
F5-TTS (Chen et al., 2024c)	1.78	0.66	-0.18$_{\pm 0.14}$	-0.21$_{\pm 0.16}$
VOICESTAR	2.15	0.63	-0.32$_{\pm 0.15}$	-0.18$_{\pm 0.14}$

Table 4: Short context evaluation sets ($\leq$ 20 seconds): LibriSpeech-PC and Seed-TTS. N-CMOS stands for naturalness comparative MOS, and S-CMOS stands for Speaker Similarity Comparative MOS, where model generated samples are compared to ground truth. Except for ground truth which are taken from Anastassiou et al. (2024) and Chen et al. (2024c), numbers for other models are reproduced by us using the corresponding official codebases, and a comparison between the reported numbers in other papers and our reproduced numbers can be found in App. B.2.

4.4 Long-form TTS

Long-form TTS is much more challenging than short-form. To force a model to generate speech of a predefined duration, we only compare models that support duration control. We see in Tab. 5 that across different context length ranges, VOICESTAR significantly outperforms other models on WER, intelligibility MOS (I-MOS), and naturalness MOS (N-MOS), and the gap is especially large as the context length increases. As for speaker similarity, VOICESTAR is a close second on every range, and is on par with or better than ground truth target speech in the 20-40s range. By listening to bad samples, we found that different models have different failure modes - for F5-TTS and MaskGCT, the models tend to clone the prompt voice well but devolve into unintelligible speech or random words. In contrast, VOICESTAR sometimes generates speech in a voice that slightly deviates from that of the prompt, but the generated words follow the target transcript closely and with a high naturalness.

Model	WER	SpkSim	S-MOS	I-MOS	N-MOS
20s-30s, within training duration					
Ground Truth	3.78	0.86	3.95$_{\pm 0.16}$	4.01$_{\pm 0.13}$	3.63$_{\pm 0.14}$
F5-TTS (Chen et al., 2024c)	5.31	0.70	4.03$_{\pm 0.15}$	3.47$_{\pm 0.15}$	2.97$_{\pm 0.17}$
MaskGCT (Wang et al., 2024c)	3.91	0.76	**4.29**$_{\pm \mathbf{0.13}}$	3.55$_{\pm 0.16}$	3.07$_{\pm 0.18}$
VOICESTAR	3.53	0.72	4.13$_{\pm 0.14}$	**3.75**$_{\pm \mathbf{0.14}}$	**3.45**$_{\pm \mathbf{0.15}}$
30s-40s, extrapolation					
Ground Truth	3.13	0.86	4.12$_{\pm 0.19}$	4.21$_{\pm 0.15}$	4.05$_{\pm 0.16}$
F5-TTS (Chen et al., 2024c)	34.15	0.70	3.97$_{\pm 0.15}$	2.21$_{\pm 0.15}$	2.15$_{\pm 0.16}$
MaskGCT (Wang et al., 2024c)	13.81	0.75	**4.27**$_{\pm \mathbf{0.14}}$	3.06$_{\pm 0.18}$	2.79$_{\pm 0.18}$
VOICESTAR	7.27	0.70	4.11$_{\pm 0.17}$	**4.16**$_{\pm \mathbf{0.14}}$	**3.69**$_{\pm \mathbf{0.16}}$
40s-50s, extrapolation					
Ground Truth	2.52	0.87	4.33$_{\pm 0.16}$	4.52$_{\pm 0.11}$	3.99$_{\pm 0.16}$
F5-TTS (Chen et al., 2024c)	52.44	0.70	**4.04**$_{\pm \mathbf{0.17}}$	1.94$_{\pm 0.15}$	2.35$_{\pm 0.14}$
MaskGCT (Wang et al., 2024c)	82.29	0.65	3.95$_{\pm 0.20}$	1.42$_{\pm 0.10}$	1.61$_{\pm 0.13}$
VOICESTAR	11.91	0.70	3.94$_{\pm 0.18}$	**3.23**$_{\pm \mathbf{0.19}}$	**3.23**$_{\pm \mathbf{0.19}}$

Table 5: Long context evaluation results. All models are trained with a maximal training length of 30 seconds. I in I-MOS stands for intelligibility, N stands for naturalness, and S stands for speaker similarity.

5 Conclusion

We introduce VOICESTAR, a novel autoregressive NCLM-based, zero-shot TTS model that achieves robust and duration-controllable speech synthesis, as well as the ability to generate speech longer than what was seen during training. The key novel contributions of our approach include the Progress-Monitoring Rotary Position Embedding (PM-ROPE) for effective text-speech alignment, duration control, and extrapolation; and continuation-prompt mixed training (CPM) to address the training/inference mismatch. By scaling up the training data and model size, VOICESTAR sets new state-of-the-art results on both short-form and long-form TTS benchmarks.

Limitations

Speaker similarity. On short-form TTS benchmarks, thanks to the prompt repetition trick, our model is either SotA or on par with SotA on speaker similarity. However, on long-form TTS, since the context length is already at or exceeds maximal training duration, excessive prompt repetition can significantly degrade intelligibility and therefore our model exhibits a gap with SotA speaker similarity. We argue that speaker similarity is not a weakness of the main ideas proposed in this paper, and can be separately addressed by e.g. using more advanced neural codec model (Ye et al., 2025), or adapting multistage modeling approaches (Wang et al., 2024c; Du et al., 2024c).

Generation speed. Our 840M model has a real-time factor (RTF) larger than 1 and therefore cannot perform faster-than-real-time generation. Although this could be mitigated by techniques such as quantization (Dettmers et al., 2022), grouped prediction (Chen et al., 2024b), speculative decoding (Li et al., 2025; Nguyen et al., 2025), etc.

References

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, and 27 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *ArXiv*, abs/2406.02430.

Eric Battenberg, R. J. Skerry-Ryan, Daisy Stanton, Soroosh Mariooryad, Matt Shannon, Julian Salazar, and David Kao. 2024. Very attentive tacotron: Robust and unbounded length generalization in autoregressive transformer-based text-to-speech. *ArXiv*, abs/2410.22179.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Zalán Borsos, Matthew Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. 2023. Soundstorm: Efficient parallel audio generation. *ArXiv*, abs/2305.09636.

Chen Chen, Yuchen Hu, Wen Wu, Helin Wang, Chng Eng Siong, and Chao Zhang. 2024a. Enhancing zero-shot text-to-speech synthesis with human feedback. *ArXiv*, abs/2406.00654.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024b. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *Preprint*, arXiv:2406.05370.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. 2021. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16:1505–1518.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024c. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *ArXiv*, abs/2410.06885.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Defossez. 2023. Simple and controllable music generation. *ArXiv*, abs/2306.05284.

Alexandre Defossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *ArXiv*, abs/2210.13438.

Alexandre D'efossez, Laurent Mazar'e, Manu Orsini, Am'elie Royer, Patrick P'erez, Herv'e J'egou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *ArXiv*, abs/2410.00037.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Preprint*, arXiv:2208.07339.

Anuj Diwan, Zhisheng Zheng, David Harwath, and Eunsol Choi. 2025. Scaling rich style-prompted text-to-speech datasets.

Chenpeng Du, Yiwei Guo, Hankun Wang, Yifan Yang, Zhikang Niu, Shuai Wang, Hui Zhang, Xie Chen, and Kai Yu. 2024a. Vall-t: Decoder-only generative transducer for robust and decoding-controllable text-to-speech.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. 2024b. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *ArXiv*, abs/2407.05407.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jing-Ru Zhou. 2024c. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *ArXiv*, abs/2412.10117.

Jonathan Duddington, Reece Dunn, and the eSpeak NG Community. eSpeak NG: Speech synthesiser. https://github.com/espeak-ng/espeak-ng. Accessed: March 15, 2025.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 682–689.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024. Llama-omni: Seamless speech interaction with large language models. *ArXiv*, abs/2409.06666.

Hao-Han Guo, Kun Liu, Fei-Yu Shen, Yi-Chen Wu, Fenglong Xie, Kun Xie, and Kai-Tuo Xu. 2024a. Fireredtts: A foundation text-to-speech framework for industry-level generative speech applications. *ArXiv*, abs/2409.03283.

Haohan Guo, Fenglong Xie, Dongchao Yang, Xixin Wu, and Helen M. Meng. 2024b. Speaking from coarse to fine: Improving neural codec language model via multi-scale speech coding and generation. *ArXiv*, abs/2409.11630.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xuejiao Tan. 2022. Prompttts: Controllable text-to-speech with text descriptions. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Bing Han, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Yanmin Qian, Yanqing Liu, Sheng Zhao, Jinyu Li, and Furu Wei. 2024. Vall-e r: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment. *ArXiv*, abs/2406.07855.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890.

Yuchen Hu, Chen Chen, Siyin Wang, Chng Eng Siong, and Chao Zhang. 2024. Robust zero-shot text-to-speech synthesis with reverse inference optimization. *ArXiv*, abs/2407.02243.

Shehzeen Samarah Hussain, Paarth Neekhara, Xuesong Yang, Edresson Casanova, Subhankar Ghosh, Mikyas T. Desta, Roy Fejgin, Rafael Valle, and Jason Li. 2025. Koel-tts: Enhancing llm based speech generation with preference alignment and classifier free guidance.

Shengpeng Ji, Jia li Zuo, Minghui Fang, Ziyue Jiang, Feiyang Chen, Xinyu Duan, Baoxing Huai, and Zhou Zhao. 2023. Textrolspeech: A text style control speech corpus with codec language text-to-speech models. *ArXiv*, abs/2308.14430.

Ziyue Jiang, Jinglin Liu, Yi Ren, Jinzheng He, Zhenhui Ye, Shengpeng Ji, Qian Yang, Chen Zhang, Pengfei Wei, Chunfeng Wang, Xiang Yin, Zejun MA, and Zhou Zhao. 2024. Mega-TTS 2: Boosting prompting mechanisms for zero-shot speech synthesis. In *The Twelfth International Conference on Learning Representations*.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2023. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matthew Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718.

Sungwon Kim, Kevin J. Shih, Rohan Badlani, João Felipe Santos, Evelina Bakhturina, Mikyas T. Desta, Rafael Valle, Sungroh Yoon, and Bryan Catanzaro. 2023. P-flow: A fast and data-efficient zero-shot tts through speech prompting. In *Neural Information Processing Systems*.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Defossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2022. Audiogen: Textually guided audio generation. *ArXiv*, abs/2209.15352.

Mateusz Lajszczak, Guillermo Cámbara, Yang Li, Fatih Beyhan, Arent van Korlaar, Fan Yang, Arnaud Joly, Álvaro Martín-Cortinas, Ammar Abbas, Adam Michalski, Alexis Moinet, Sri Karlapati, Ewa Muszy'nska, Haohan Guo, Bartosz Putrycz, Soledad López Gambino, Kayeon Yoo, Elena Sokolova, and Thomas Drugman. 2024. Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data. *ArXiv*, abs/2402.08093.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu Nguyen, Jade Copet, Alexei Baevski, Adel Ben Mohamed, and Emmanuel Dupoux. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Matt Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *ArXiv*, abs/2306.15687.

Keon Lee, Dong Won Kim, Jaehyeon Kim, Seungjun Chung, and Jaewoong Cho. 2024. Ditto-tts: Diffusion transformers for scalable text-to-speech without domain-specific factors.

Yichong Leng, Zhifang Guo, Kai Shen, Xu Tan, Zeqian Ju, Yanqing Liu, Yufei Liu, Dongchao Yang, Leying Zhang, Kaitao Song, Lei He, Xiang-Yang Li, Sheng Zhao, Tao Qin, and Jiang Bian. 2023. Prompttts 2: Describing and generating voices with text prompt. *ArXiv*, abs/2309.02285.

Bohan Li, Hankun Wang, Situo Zhang, Yiwei Guo, and Kai Yu. 2025. Fast and high-quality auto-regressive speech synthesis via speculative decoding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yingahao Aaron Li, Rithesh Kumar, and Zeyu Jin. 2024. Dmospeech: Direct metric optimization via distilled diffusion model in zero-shot speech synthesis.

Guanghou Liu, Yongmao Zhang, Yinjiao Lei, Yunlin Chen, Rui Wang, Zhifei Li, and Linfu Xie. 2023. Promptstyle: Controllable style transfer for text-to-speech with natural language descriptions. *ArXiv*, abs/2305.19522.

Philipos C Loizou. 2011. Speech quality assessment. In *Multimedia analysis, processing and communications*, pages 623–654. Springer.

Daniel Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *ArXiv*, abs/2402.01912.

Ziyang Ma, Ya-Zhen Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024. Language model can listen while speaking. *ArXiv*, abs/2408.02622.

Soumi Maiti, Yifan Peng, Shukjae Choi, Jee weon Jung, Xuankai Chang, and Shinji Watanabe. 2023. Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13326–13330.

Paarth Neekhara, Shehzeen Samarah Hussain, Subhankar Ghosh, Jason Li, Rafael Valle, Rohan Badlani, and Boris Ginsburg. 2024. Improving robustness of llm-based speech synthesis by learning monotonic alignment. *ArXiv*, abs/2406.17957.

Tan Dat Nguyen, Ji-Hoon Kim, Jeongsoo Choi, Shukjae Choi, Jinseok Park, Younglo Lee, and Joon Son Chung. 2025. Accelerating codec-based speech synthesis with multi-token prediction and speculative decoding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yuto Nishimura, Takumi Hirose, Masanari Ohi, Hideki Nakayama, and Nakamasa Inoue. 2024. Hall-e: Hierarchical neural codec language model for minute-long zero-shot text-to-speech synthesis. *ArXiv*, abs/2410.04380.

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. Voicecraft: Zero-shot speech editing and text-to-speech in the wild. In *Annual Meeting of the Association for Computational Linguistics*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. *Preprint*, arXiv:2108.12409.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *ArXiv*, abs/2212.04356.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *Preprint*, arXiv:2204.02152.

Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. 2023. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *ArXiv*, abs/2304.09116.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. Ella-v: Stable neural codec language modeling with alignment-guided sequence reordering.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Apoorv Vyas, Bowen Shi, Matt Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, W.K.F. Ngan, Jeff Wang, Ivan Cruz, Bapi Akula, Akinniyi Tunde Akinyemi, Brian

Ellis, Rashel Moritz, Yael Yungster, Alice Rakotoarison, Liang Tan, and 5 others. 2023. Audiobox: Unified audio generation with natural language prompts. *ArXiv*, abs/2312.15821.

Chengyi Wang, Sanyuan Chen, Yu Wu, Zi-Hua Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *ArXiv*, abs/2301.02111.

Hankun Wang, Chenpeng Du, Yiwei Guo, Shuai Wang, Xie Chen, and Kai Yu. 2024a. Attention-constrained inference for robust decoder-only text-to-speech. *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 630–637.

Helin Wang, Meng Yu, Jiarui Hai, Chen Chen, Yuchen Hu, Rilin Chen, Najim Dehak, and Dong Yu. 2024b. Ssr-speech: Towards stable, safe and robust zero-shot text-based speech editing and synthesis. *ArXiv*, abs/2409.07556.

Jiaming Wang, Zhihao Du, Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, Wen Wang, Siqi Zheng, Chang Zhou, Zhijie Yan, and Shiliang Zhang. 2023b. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *ArXiv*, abs/2310.04673.

Tianrui Wang, Long Zhou, Zi-Hua Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023c. Viola: Unified codec language models for speech recognition, synthesis, and translation. *ArXiv*, abs/2305.16107.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, Weizhen Bian, Zhen Ye, Sitong Cheng, Ruibin Yuan, Zhixian Zhao, Xinfa Zhu, Jiahao Pan, Liumeng Xue, Pengcheng Zhu, and 6 others. 2025a. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2024c. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *ArXiv*, abs/2409.00750.

Yuancheng Wang, Jiachen Zheng, Junan Zhang, Xueyao Zhang, Huan Liao, and Zhizheng Wu. 2025b. Metis: A foundation speech generation model with masked generative pre-training.

Yihan Wu, Soumi Maiti, Yifan Peng, Wangyou Zhang, Chenda Li, Yuyue Wang, Xihua Wang, Shinji Watanabe, and Ruihua Song. 2024. Speechcomposer: Unifying multiple speech tasks with prompt composition. *ArXiv*, abs/2401.18045.

Detai Xin, Xu Tan, Kai Shen, Zeqian Ju, Dongchao Yang, Yuancheng Wang, Shinnosuke Takamichi, Hiroshi Saruwatari, Shujie Liu, Jinyu Li, and Sheng Zhao. 2024. Rall-e: Robust codec language modeling with chain-of-thought prompting for text-to-speech synthesis. *ArXiv*, abs/2404.03204.

Wang Xu, Shuo Wang, Weilin Zhao, Xu Han, Yukun Yan, Yudi Zhang, Zhe Tao, Zhiyuan Liu, and Wanxiang Che. 2024. Enabling real-time conversations with minimal training costs. *ArXiv*, abs/2409.11727.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Guangzhi Lei, Chao Weng, Helen M. Meng, and Dong Yu. 2023. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *ArXiv*, abs/2301.13662.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *ICLR*.

Zhen Ye, Xinfa Zhu, Chi min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, Hongzhan Lin, Jianyi Chen, Xingjian Du, Liumeng Xue, Yunlin Chen, Zhifei Li, Lei Xie, Qiuqiang Kong, Yi-Ting Guo, and Wei Xue. 2025. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2024. Salmonn-omni: A codec-free llm for full-duplex speech understanding and generation. *ArXiv*, abs/2411.18138.

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

Hao Zhang, Weiwei Li, Rilin Chen, Vinay Kothapally, Meng Yu, and Dong Yu. 2025. Llm-enhanced dialogue management for full-duplex spoken dialogue systems.

Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao, Yuxuan Wang, Bin Zhang, Heng Lu, Yaqian Zhou, and Xipeng Qiu. 2024. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *ArXiv*, abs/2410.08035.

Zi-Hua Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *ArXiv*, abs/2303.03926.

Zhisheng Zheng, Puyuan Peng, Anuj Diwan, Cong Phuoc Huynh, Xiaohang Sun, Zhu Liu, Vimal Bhat, and David Harwath. 2025. Voicecraft-x:

Unifying multilingual, voice-cloning speech synthesis and speech editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2737–2756.

Yongxin Zhu, Dan Su, Liqiang He, Linli Xu, and Dong Yu. 2024. Generative pre-trained speech language model with efficient hierarchical transformer. *ArXiv*, abs/2406.00976.

A Background and Model Details

A.1 Preliminaries: Dot-product Attention and RoPE

Here we provide the mathematical background for the standard attention mechanisms and Rotary Position Embeddings that serve as the foundation for our PM-RoPE.

Dot-product attention. We use the commonly used "source-target" terminology throughout to describe different attention mechanisms. Note that target and source can reference to the same sequence, in which case attention refers to self-attention. Denote the source token embedding at position s as $X_s \in \mathbb{R}^D, 1 \leq s \leq S$ and the embedding of target token at position t as $Y_t \in \mathbb{R}^D, 1 \leq t \leq T$. The embeddings first get projected into key, query, and value vectors as

$$K_s = f_k(X_s, s) := W_k X_s,$$
$$Q_t = f_q(Y_t, t) := W_q Y_t,$$
$$V_s = f_v(X_s, s) := W_v X_s$$

Where W_k, W_q, and W_v are learnable matrices.

We denote the inner product operator as $\langle \cdot, \cdot \rangle$. Also let the attention weight from target token at position t to source token at position s be $W_{t,s}$, and the output of the attention layer at position t be O_t, which are calculated according to:

$$E_{t,s} = \frac{\langle f_q(Y_t, t), f_k(X_s, s) \rangle}{\sqrt{D}}$$
$$W_{t,s} = \frac{\exp(E_{t,s})}{\sum_{s'} \exp(E_{t,s'})}$$

and $O_t = \sum_{s'} W_{t,s'} V_{s'}$.

The attention mechanism is a powerful way to provide contextualization, but it does not explicitly include token position information. Therefore, various position embedding methods have been proposed to address this issue (Vaswani et al., 2023; Raffel et al., 2023; Su et al., 2023; Press et al., 2022).

RoPE (Su et al., 2023). Rotary Position Embedding (RoPE) redefines the projection functions $f_q(Y_t, t)$ and $f_k(X_s, s)$ such that their inner product is a function $g()$ of the corresponding target and source token embeddings, and their relative position, which can be formally written as:

$$\langle f_q^{\text{RoPE}}(Y_t, t), f_k^{\text{RoPE}}(X_s, s) \rangle = g(Y_t, X_s, t - s)$$

For simplicity, we first explain RoPE in the 2-dimensional case. The 2-D rotation matrix $R(\gamma)$ is

defined as:

$$R(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}$$

RoPE applies this rotation matrix to key and query calculation according to:

$$f_k^{\text{RoPE}}(X_s, s) = R(s\theta)W_k X_s,$$
$$f_q^{\text{RoPE}}(Y_t, t) = R(t\theta)W_q Y_t$$

Where θ is a hyperparameter that specifies the per position rotation angle. The inner product between the key and the query becomes:

$$\langle f_q^{\text{RoPE}}(Y_t, t), f_k^{\text{RoPE}}(X_s, s)\rangle = Y_t^\top W_q^\top R((t-s)\theta)W_k X_s$$

Observe that due to the application of the rotation matrix, the inner product is a function of Y_t, X_s, and their relative position within the input sequence $t - s$.

The value vector is calculated the same as in vanilla attention, i.e. $f_v^{\text{RoPE}}(X_s, s) := f_v(X_s, s)$. Also the attention weights $W_{t,s}$ and overall attention output O_t are calculated similarly to vanilla attention.

A.2 RoPE and PM-RoPE for higher dimensions

For higher dimensions, rotation matrix is defined by dividing the space into D/2 2-dimensional spaces, and applying a 2-dimensional rotation matrix (with different θs) to each space. Mathematically, see Eq. 1. Similarly, the rotation matrix for PM-RoPE in higher dimension is show in Eq. 2, where T is the total length of the sequence.

where $\Theta = \{\theta_i = 10000^{-2(i-1)/D}, i \in [1, 2, ..., D/2]\}$. RoPE requires D is even.

A.3 text-speech alignment in different models

for decoder only model, text tokens usually preceed speech tokens; for encoder-decoder model, text tokens are the input to the encoder and speech tokens are the input to decoder, and speech token attends to text tokens via cross-attention. Here we visualize self-attention in encoder only model and cross-attention in encoder-decoder model. We set text and speech token embeddings to unit vectors, so that visualized attention maps solely reveal the inductive bias introduce by model architecture and position embeddings.

B More Results

B.1 Comparing impact of prompt repetition on VOICESTAR v.s. F5-TTS

Fig. 7a and Fig. 7b show the effect of prompt repetition technique on VOICESTAR and F5-TTS. We see that prompt repetition is not suitable for F5-TTS, because although it improves speaker similarity slightly when the number of repetitions is small, it quickly degrades WER significantly as the number of repetition increases.

B.2 reported and reproduced results

Tab. 6 and Tab. 7 show the reported and reproduced results of different models on LibriSpeech-PC and Seed-TTS (en). We see that most reported numbers are closely reproduced, except for FireRedTTS and VoiceCraft. For VoiceCraft, we found that instead of using the recommended top p sampling with 0.9 probability, using top k sampling with top k of 40 significantly improves results.

Model	WER	SpkSim
Reported		
Ground Truth	2.23	0.69
FireRedTTS	3.82	0.46
CosyVoice	3.39	0.64
F5-TTS	2.42	0.66
Reproduced		
FireRedTTS	7.73	0.45
Llasa 1B	4.28	0.47
CosyVoice	3.40	0.66
VoiceCraft	3.09	0.52
MaskGCT	2.66	0.66
CosyVoice2	2.66	0.66
F5-TTS	2.57	0.65
VOICESTAR	2.64	0.63

Table 6: LibriSpeech-PC (Chen et al., 2024c). Reported numbers are taken from (Chen et al., 2024c).

B.3 Using Character duration-based estimation v.s. ground truth duration

Tab. 8 shows objective metrics comparing model performance using ground truth duration versus using estimated duration (Chen et al., 2024c). We see that in most cases, using estimated duration will degrade the performance. We argue that ground truth duration is not necessary for good performance in that we could train duration predictor (Le et al.,

$$R^{D}_{\Theta,t} = \begin{pmatrix} \cos t\theta_1 & -\sin t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin t\theta_1 & \cos t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos t\theta_2 & -\sin t\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin t\theta_2 & \cos t\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos t\theta_{D/2} & -\sin t\theta_{D/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin t\theta_{D/2} & \cos t\theta_{D/2} \end{pmatrix} \tag{1}$$

$$R^{D}_{\Theta,t,T} = \begin{pmatrix} \cos \frac{t}{T}N\theta_1 & -\sin \frac{t}{T}N\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin \frac{t}{T}N\theta_1 & \cos \frac{t}{T}N\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos \frac{t}{T}N\theta_2 & -\sin \frac{t}{T}N\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin \frac{t}{T}N\theta_2 & \cos \frac{t}{T}N\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos \frac{t}{T}N\theta_{D/2} & -\sin \frac{t}{T}N\theta_{D/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin \frac{t}{T}N\theta_{D/2} & \cos \frac{t}{T}N\theta_{D/2} \end{pmatrix} \tag{2}$$

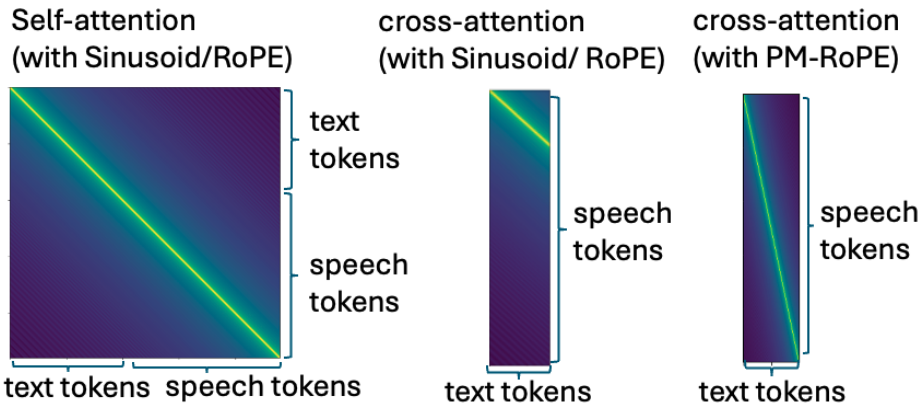


Figure 6: attention maps

2023; Lee et al., 2024) that can potentially achieve much better duration estimation than the current naive approach.

C Instructions for human listening test

Screenshots of instructions for all human listening test we used on Amazon Mechanical Turk is shown in figure 9 to figure 12. We used 13 dollar per hour to reward the workers. We only allow Turkers who are resident of the US, Australia, Canada, and UK to participate, in hope that the workers are native English speakers. We acknowledge that this is not a perfect approach and might lead to bias in judgment, but since Amazon Mechanical Turk doesn't allow selection on native language, this is the best approach we could think of as a proxy to constraining the native language.

D Broader Societal Impact

D.1 Potential Risks and Mitigation

VOICESTAR introduces novel capabilities in zero-shot voice cloning and length extrapolation, allowing for the generation of high-fidelity, long-form speech. We acknowledge that these advancements carry potential risks of misuse, including impersonation, fraud, and the creation of misleading deepfakes. To mitigate these risks, we are actively devoted to developing audio watermarking and anti-spoofing techniques to help identify synthesized content. Furthermore, by open-sourcing our code and weights, we aim to assist the research community in developing robust detection systems against deepfakes.

Reported			Reproduced		
Model	WER	SpkSim	Model	WER	SpkSim
Ground Truth	2.15	0.73	FireRedTTS	9.09	0.45
VoiceCraft	7.56	0.47	Llasa 1B	5.13	0.58
FireRedTTS	3.82	0.46	CosyVoice	4.20	0.64
CosyVoice	3.39	0.64	VoiceCraft	3.45	0.52
Llasa 1B	3.22	0.57	CosyVoice2	2.69	0.66
MaskGCT	2.47	0.72	MaskGCT	2.52	0.71
CosyVoice2	2.38	0.65	F5-TTS	1.78	0.66
Metis	2.28	0.72	VOICESTAR	2.15	0.63
Seed-TTS	2.14	0.76			
SparkTTS	1.98	0.58			
F5-TTS	1.83	0.67			

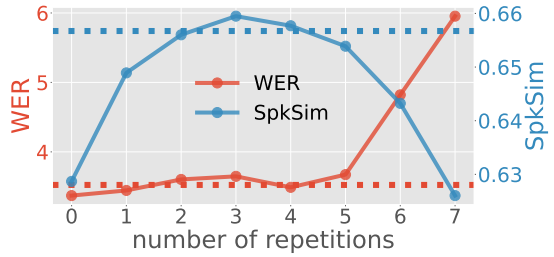
Table 7: Seed-TTS English performance comparison. The left section shows Reported numbers taken from corresponding cited papers (Anastassiou et al., 2024; Wang et al., 2024c; Chen et al., 2024c; Du et al., 2024c; Ye et al., 2025; Wang et al., 2025b,a). The right section shows our Reproduced results. For VoiceCraft, our reproduced version uses topk of 40 for sampling which leads to better performance. Models in the Reported section are ordered by arXiv submission time.

Model	WER		SpkSim	
Duration	GT	Est.	GT	Est.
20s-30s, within training duration				
Ground Truth	3.78	-	0.86	-
F5-TTS (Chen et al., 2024c)	5.31	7.94	0.70	0.70
MaskGCT (Wang et al., 2024c)	3.91	4.74	0.76	0.76
VOICESTAR	3.53	4.60	0.72	0.71
30s-40s, extrapolation				
Ground Truth	3.13	-	0.86	-
F5-TTS (Chen et al., 2024c)	34.15	33.34	0.70	0.70
MaskGCT (Wang et al., 2024c)	13.81	26.39	0.75	0.73
VOICESTAR	7.27	8.29	0.70	0.69
40s-50s, extrapolation				
Ground Truth	2.52	-	0.87	-
F5-TTS (Chen et al., 2024c)	52.44	50.28	0.70	0.70
MaskGCT (Wang et al., 2024c)	82.29	77.24	0.65	0.64
VOICESTAR	11.91	17.33	0.70	0.66

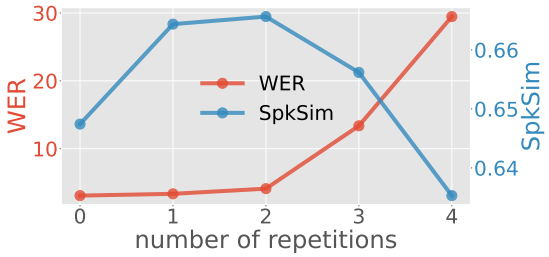
Table 8: Long context evaluation results listed as given duration estimated by character duration of prompt versus ground truth duration. All models are trained with a maximal training length of 30 seconds. I in I-MOS stands for intelligibility, N stands for naturalness, and S stands for speaker similarity.

D.2 Licensing and Data Usage

Our model utilizes the Emilia and Libriheavy datasets for training. We will release our model weights under a responsible use license that explicitly prohibits illegal applications, such as non-consensual voice cloning or the spread of disinformation.



(a) VOICESTAR(same figure shown in main text).



(b) F5-TTS.

Figure 7: Comparison of the impact of prompt repetition on VOICESTAR (left) and F5-TTS (right). In the VOICESTAR figure, the red and blue horizontal dotted lines indicate performance when we repeat the prompt so the context length (prompt + generation) reaches the maximal training duration. We do not show the dotted lines for F5-TTS because the model does not produce intelligible speech (WER is 75.4). We see that VOICESTAR's WER stays stably low while SpkSim benefits from repeating the prompt until repeating for more than 6 times, while for F5-TTS, repeating the prompt for more than 2 times degrades WER significantly. Note that the WER y-axis ticks for VOICESTAR and F5-TTS are at very different scale.

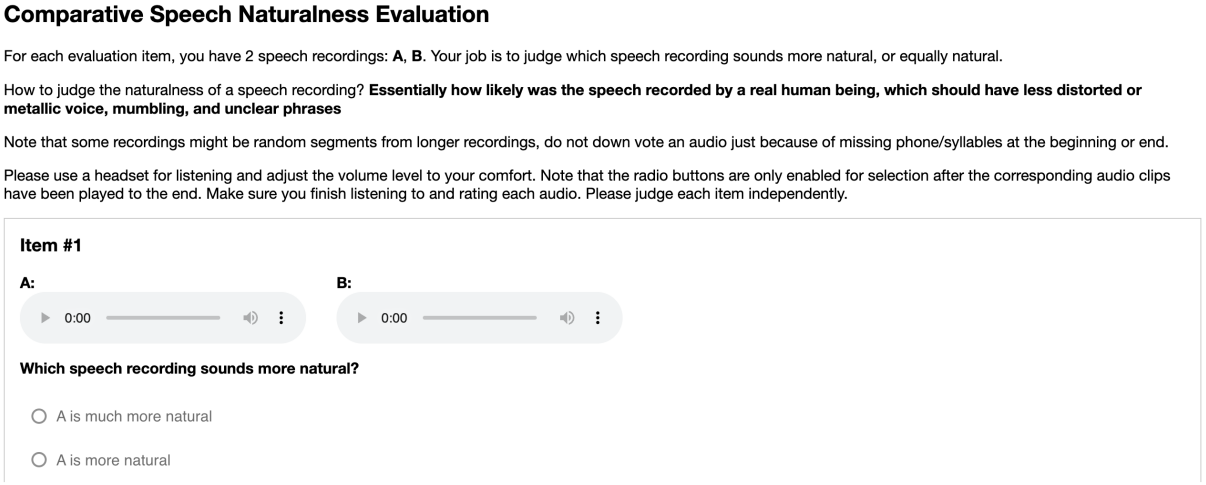


Figure 8: screenshots for comparative speech naturalness human evaluation. There are seven options: A is much more natural, A is more natural, A is slightly more natural, equally natural, B is much more natural, B is more natural, B is slightly more natural.

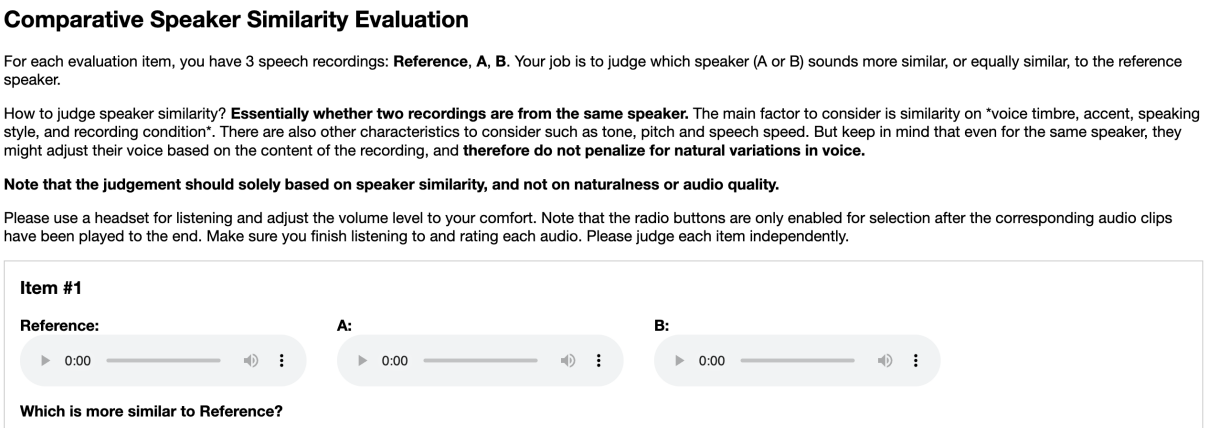


Figure 9: screenshots for comparative speaker similarity human evaluation. There are also seven options, replacing the word "natural" in comparative naturalness task with "similar"

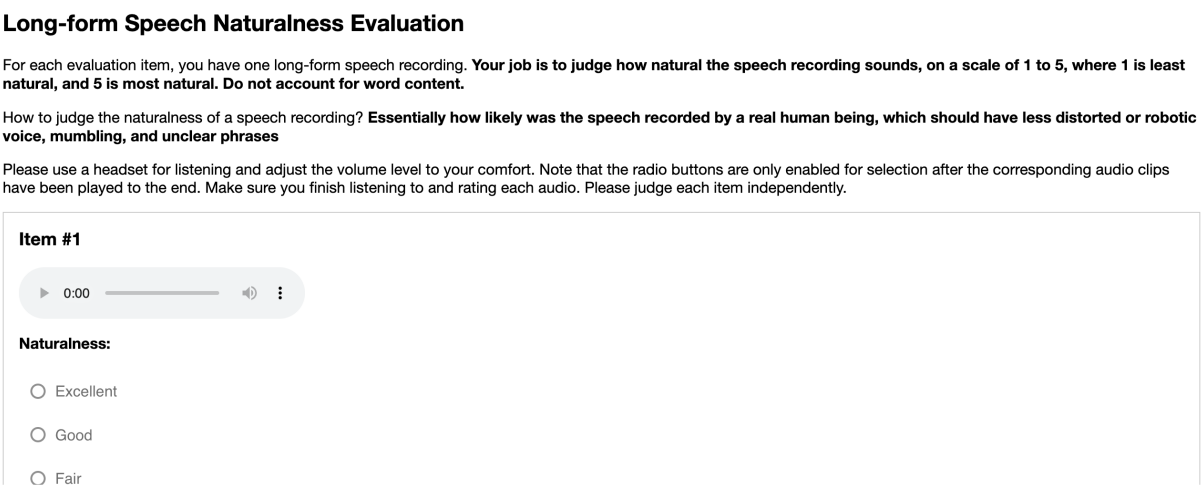


Figure 10: screenshots for speech naturalness human evaluation. There are five options, Excellent, Good, Fair, Poor, Bad

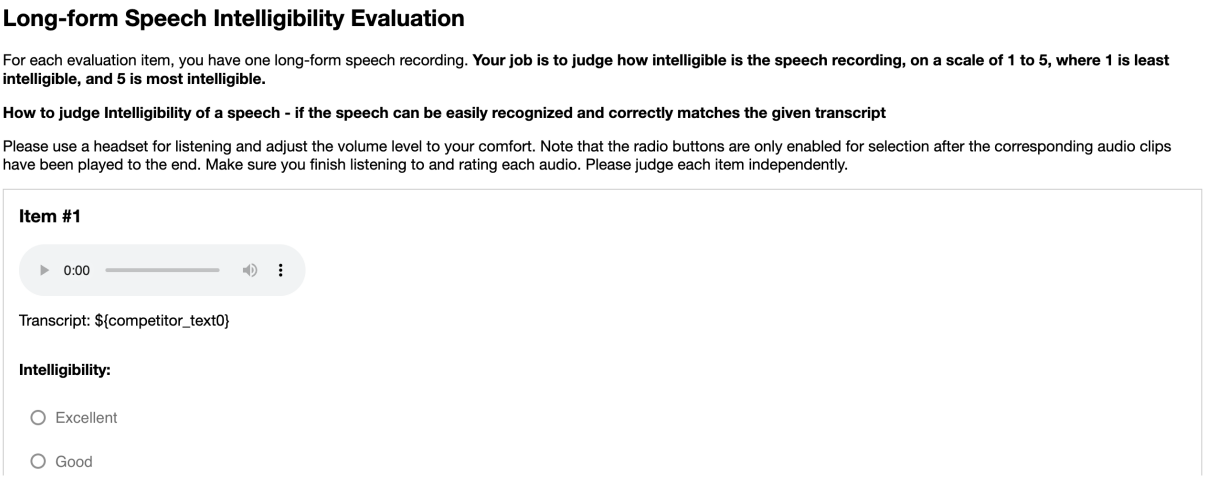


Figure 11: screenshots for speech intelligibility human evaluation. There are five options, Excellent, Good, Fair, Poor, Bad

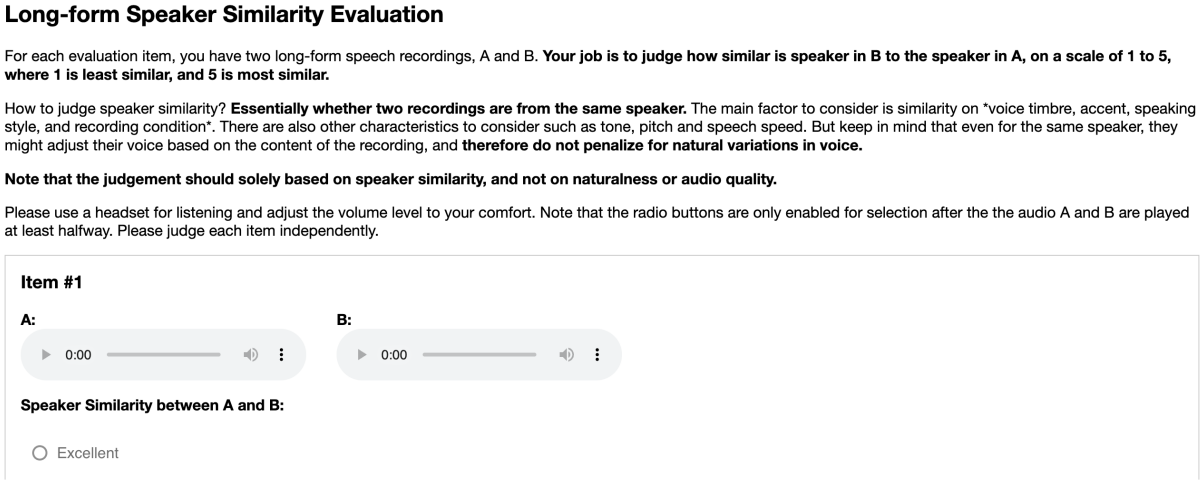


Figure 12: screenshots for speech intelligibility human evaluation. There are five options, Excellent, Good, Fair, Poor, Bad