ELLA-V: Stable Neural Codec Language Modeling with Alignment-guided Sequence Reordering

Anonymous ACL submission

Abstract

The language model (LM) approach based on acoustic and linguistic prompts, such as VALL-E, has achieved remarkable progress in the field of zero-shot audio generation. However, existing methods still have some limitations: 1) repetitions, transpositions, and omissions in the output synthesized speech due to limited alignment constraints between audio and phoneme tokens; 2) challenges of fine-grained control over the synthesized speech with autoregressive (AR) language model; 3) infinite silence generation due to the nature of AR-based decoding, especially under the greedy strategy. To alleviate these issues, we propose ELLA-V[1], a simple but efficient LM-based zero-shot text-to-speech (TTS) framework, which enables fine-grained control over synthesized audio at the phoneme level. The key to ELLA-V is interleaving sequences of acoustic and phoneme tokens, where phoneme tokens appear ahead of the corresponding acoustic tokens. The experimental findings reveal that our model outperforms VALL-E in terms of accuracy and delivers more stable results using both greedy and sampling-based decoding strategies. The code of ELLA-V will be open-sourced after cleanups[2]. Audio samples are available at `https://anonymous.4open.science/w/ELLAV_anonymous/`.

1 Introduction

Recently, deep generative AI has achieved remarkable results in various tasks, leading to the emergence of many transformative real-world applications (Brown et al., 2020; Ramesh et al., 2022; Ho et al., 2020). With the advancement of generative models, there have been rapid developments in the field of speech synthesis as well. In particular, zero-shot TTS technology has gained increasing attention because it can synthesize high-quality target voices without the need of specified speaker's training data. As a state-of-the-art generative model family, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song and Ermon, 2020) progressively add noise to the training data and then learn the reverse process to generate samples. By leveraging diffusion models and their variants (Sohl-Dickstein et al., 2015; Ho et al., 2020; Lipman et al., 2023), many works have successfully applied them to the audio domain (Huang et al., 2022, 2023; Shen et al., 2023). Another major class of generative models is language modeling based on Transformer (Vaswani et al., 2017). AR language models use a decoder-only architecture to predict the next token in a sequence as the training objective, which has demonstrated extremely powerful few-shot and zero-shot capabilities in many generative tasks (Brown et al., 2020; Thoppilan et al., 2022; Chowdhery et al., 2023). In light of this, VALL-E (Wang et al., 2023a) and subsequent works (Kharitonov et al., 2023; Rubenstein et al., 2023; Wang et al., 2023b) have successfully employed decoder-only language model for zero-shot TTS. These approaches first quantize the speech signal into a series of discrete acoustic tokens. Subsequently, they employ an AR language model to predict coarse-grained acoustic tokens, eliminating the necessity for explicit duration predictors or speaker encoders. Once trained on a large-scale corpus, these approaches are capable of synthesizing speech with competitive fidelity and naturalness in a zero-shot manner.

While VALL-E and its variants have achieved numerous impressive milestones, they still possess certain limitations that impact practical deployment. For instance, existing methods (Wang et al., 2023a; Kharitonov et al., 2023) directly concatenate phoneme tokens and acoustic tokens as a whole sequence to train language models. It is potentially unstable to learn the alignment be-

[1] ELLA-V is the word "VALL-E" spelled backwards, to highlight that our model changes the sequence order of VALL-E input.

[2] VALL-E is not officially open source. We reproduced it and open-sourced it in this repository.

tween audio and phoneme sequences through the self-attention in the transformer. Additionally, the decoder-only language model architecture can lead to potential attention degradation issues (Fu et al., 2023), where the alignment quality between the target audio sequence and the source phoneme sequence deteriorates as the generated sequence increases, resulting in inaccurate or low-quality speech outputs.

Another limitation stems from the nature of AR language modeling. AR models predict the current tokens solely based on the historical tokens without users' control in the inference process, and sometimes generate semantic repetitions or incoherence in the generated output (Yang et al., 2019; Brown et al., 2020). In the TTS task, correspondingly, VALL-E cannot directly determine which segment of the output audio corresponds to which prompt phoneme, thus there is no trivial way to promptly detect and prevent issues occurring in the generation process. These drawbacks can manifest as meaningless phoneme repetitions, transpositions, omissions, or even catastrophic *infinite silence*, *i.e.*, during the process of generation, the model anomalously outputs silence or noise tokens for an extended period of time without stopping. Specifically, Table 1 demonstrates the word error rate (WER) and the probability of the *infinite silence* in VALL-E samples at different threshold top-p for nuclear sampling (Holtzman et al., 2019). The detailed experimental setup is described in Section 4. Notably, a shift in the decoding strategy of VALL-E from fully sampling-based to fully greedy-based leads to a marked decline in sample quality. It should be emphasized that while sampling-based stochastic decoding strategies have advantages in terms of synthesis diversity, deterministic decoding strategies (*e.g.*, beam search and its variants) are more suitable for cases where there is less tolerance for synthesis errors and more emphasis on fluency and coherence (Ippolito et al., 2019).

Faced with the pros and cons of the existing methods, we introduce ELLA-V, a simple but effective language model approach for zero-shot TTS. ELLA-V proposes a generalized AR (GAR) language model to generate the first layer of residual vector quantizer (RVQ) codes of a neural codec model. Then as with VALL-E, ELLA-V employs a non-autoregressive (NAR) language model to obtain codes of the other RVQs. Our core innovation can be summarized in three key aspects:

Table 1: Comparison of VALL-E's zero-shot TTS performance across various top-p thresholds in nuclear sampling. INF% denotes the probability of *infinite silence*, which refers to instances where generation continues without stopping even when its duration exceeds twice the original length.

Top-p	WER%	INF%
1	5.47	0.00
0.99	5.00	0.20
0.95	10.99	19.06
0.9	20.85	41.43
0.7	37.71	76.76
0.4	46.59	84.39
0.0 (greedy)	49.26	87.29

- **Phoneme Token Insertion**: ELLA-V inserts phoneme tokens into corresponding positions in the acoustic sequence, effectively capturing local dependencies between phoneme and acoustic modalities.

- **Loss Computation**: Instead of maximizing the expected log-likelihood of the hybrid sequence, ELLA-V computes loss only on acoustic and special tokens (*EndOfPhone* and *EndOfSentence*). This approach not only reduces redundant computation but also allows fine-grained control in inference, preventing abnormal phoneme synthesis.

- **Local Advance**: We introduce *local advance* by shifting *EndOfPhone* and next-word phoneme tokens a few frames ahead in the input sequence. This adjustment enhances the GAR model's utilization of local dependencies for better phoneme prediction.

Experimental results, using comparable model configurations and 960 hours of speech data from LibriSpeech (Panayotov et al., 2015) as a training set, demonstrate the superiority of ELLA-V. Compared to the strong zero-shot TTS baseline VALL-E, ELLA-V significantly improves the accuracy of synthesized speech, and demonstrates comparable or superior speaker similarity and speech naturalness on a series of subjective and objective experiments. ELLA-V achieves a WER of 2.28% on the test-clean set of LibriSpeech. Notably, ELLA-V works well on a wide spectrum of decoding strategies – even greedy decoding, and still has a substantially better speech accuracy than the best of VALL-E. We further conducted ablation experiments to investigate the effects of our proposed modifications. The results indicate that the *global advance* in ELLA-V significantly improves the model's performance, while the local advance enhances the stability of the generated output.

2 Related Work

speech synthesis Speech synthesis has long been a significant topic in the fields of natural language processing, and speech processing. Early methods were based on Statistical Parametric Speech Synthesis (Zen et al., 2009), typically involving complex components such as text analysis models, acoustic models, and vocoders. Later, end-to-end neural TTS models were introduced, which synthesize Mel spectrograms and employ a vocoder (Oord et al., 2017; Prenger et al., 2019) for speech synthesis (Wang et al., 2017; Arık et al., 2017; Ren et al., 2019). Some methods, utilizing techniques such as VAE (Hsu et al., 2019; Lee et al., 2022), flow (Miao et al., 2020; Kim et al., 2020), diffusion (Jeong et al., 2021; Kim et al., 2022; Popov et al., 2021), and others (Wu and Shi, 2022), have achieved promising performance in end-to-end speech synthesis. On the other hand, models like VALL-E (Wang et al., 2023a) and AudioLM (Borsos et al., 2023) utilize autoregressive Transformers to model discrete audio tokens, achieving great in-context learning performance. When it comes to zero-shot speech synthesis, autoregressive Transformer-based models can predict and generate audio without the need for an additional duration model, which strikes a favorable balance between efficiency and performance, and has been garnering increasing attention.

3 Method

3.1 Overview

Fig. 1 demonstrates the overall architecture of ELLA-V. ELLA-V primarily follows a two-stage framework similar to VALL-E, considering zero-shot TTS as a conditional codec language modeling task. ELLA-V maps input text prompts and speech prompts into a unified vocabulary space with a text encoder and a neural codec, respectively. Different from VALL-E, an additional sequence order rearranging step is performed to the text-audio token sequence, after which, ELLA-V utilizes a decoder-only language model to learn to perform conditional generation on the hybrid sequences of phoneme and audio tokens. Detailed information about the language model will be presented in Section 3.2.

To obtain discrete audio representations, we employ a pre-trained neural audio codec model, EnCodec (Défossez et al., 2023), following VALL-E (Wang et al., 2023a). EnCodec transforms 24 kHz raw waveforms into 75 Hz discrete tokens using L RVQ layers. In our experiments, we use the same settings as VALL-E, with $L = 8$. For each quantizer, we set the codebook size to 1024. In this setting, each second of the waveform is represented by 75×8 discrete tokens from RVQ.

To obtain phoneme sequences, we apply the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to the input audio and text transcriptions. Notably, MFA not only serves as a text tokenizer but also extracts alignment relationships between phonemes and the corresponding speech. The forced alignment information is essential for ELLA-V to **change sequence order**. In Section 3.2, we will provide a detailed explanation of how this information is used to construct the target sequence.

3.2 Training: Codec Language Model

ELLA-V employs a Generalized Autoregressive Codec language model for the prediction of the first quantization layer in the EnCodec, which corresponds to capturing semantic information and coarse-grained acoustic profiles. Subsequently, a non-autoregressive language model is utilized to generate codes for the subsequent quantization layers, aimed at reconstructing fine-grained acoustic details. Specifically, given a speech corpus $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}$, where $\mathbf{x}$ represents an audio sample, and $\mathbf{y}$ is its text transcription. We utilize the EnCodec to extract the discrete representation of $\mathbf{x}$, formulated as $\mathbf{C}^{T\times 8} = \text{EnCodec}(\mathbf{y})$, where $\mathbf{C}$ represents the two-dimensional acoustic code matrix, and T is the downsampled utterance length.

We employ MFA to obtain the phoneme sequence $\mathbf{P}_{1:n}$ corresponding to the transcription $\mathbf{y}$, while also extracting forced alignment information between the audio $\mathbf{x}$ and the transcription $\mathbf{y}$: $(\mathbf{P}_{1:n}, \boldsymbol{l}_{1:n}) = \text{MFA}(\mathbf{x}, \mathbf{y})$, where n is the number of phonemes of the audio sample $\mathbf{x}$, and l_i denotes the length of the i-th phoneme of the discrete audio sequence. MFA treats silence also as a kind of phoneme, so that the original audio sequence is partitioned into n consecutive intervals corresponding to n phonemes. Specifically, let $\langle \mathbf{C}_i \rangle^{l_i \times 8}$ represent the audio sequence corresponding to the i-th phoneme, $\mathbf{C}$ is the concatenation of $\langle \mathbf{C}_i \rangle$, and we have $\langle \mathbf{C}_k \rangle_{1:l_k} = \mathbf{C}_{\sum_{i=1}^{k-1} l_i + 1 : \sum_{i=1}^{k} l_i}$

After quantization, we utilize the EnCodec decoder to reconstruct the audio waveform from

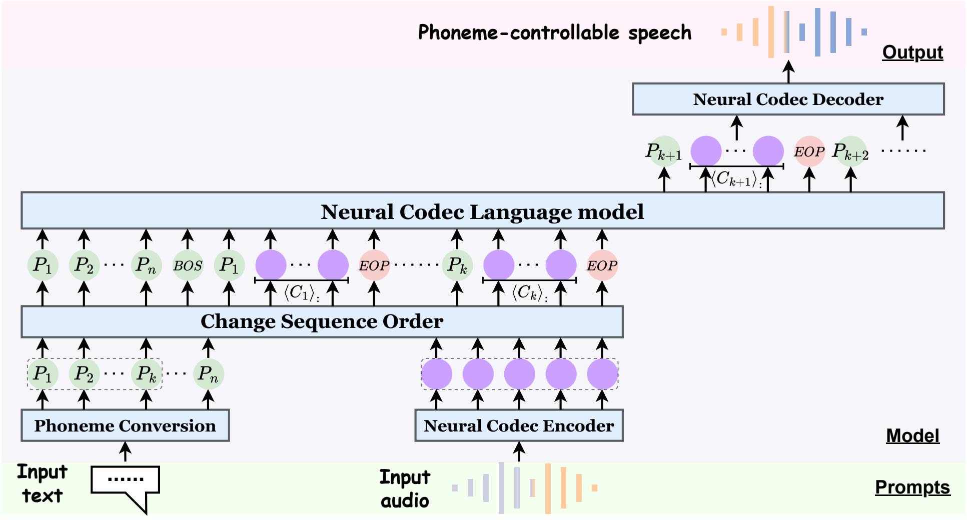


Figure 1: The overall architecture of ELLA-V. Input an audio prompts and text prompts, ELLA-V first changes sequence order – sandwiching each phoneme's audio $\langle \mathbf{C}_k \rangle_{:}$ between the k-th phoneme and a EOP token and prepending the phoneme sequence to the beginning. By learning on the mixed sequence, ELLA-V can generate audio sequence of the text prompts while maintaining the acoustic and environmental conditions of the audio prompts.

the discrete acoustic sequence $\mathbf{C}$, formulated as $\hat{\mathbf{x}} \approx \mathrm{DeCodec}(\mathbf{C})$.

For the zero-shot TTS task, the optimization objective is $\max p(\mathbf{C}|\mathbf{P}, \hat{\mathbf{C}})$, where $\hat{\mathbf{C}}$ is the acoustic prompt of the unseen speaker. We use language modeling to generate acoustic tokens for the unseen speaker, by learning on the mixed sequence composed of phonemes and codec codes, consistent with previous works (Wang et al., 2023a; Rubenstein et al., 2023).

Unlike existing approaches, ELLA-V does not concatenate phoneme tokens and acoustic tokens directly to form the target sequence for training the language model. Instead, ELLA-V **interleaves phoneme and acoustic tokens** in order to make it easier for language models to learn the alignment between audio and text. Specifically, we insert each phoneme token P_i (except the *silence phoneme*) into the corresponding position of the audio sequence, so that each phoneme's audio $\langle \mathbf{C}_i \rangle$ is sandwiched between P_i and EOP tokens. We also prepend the phoneme sequence to the beginning of the mixed sequence, which is referred to as *global advance*. In Section 3.4, we further propose a variant sequence order with higher generation stability, named *local advance*, which moves the non-acoustic tokens of the sequence several frames forward.

3.2.1 Generalized Autoregressive (GAR) Codec Language Model

As shown in Figure 2, ELLA-V first constructs a hybrid sequence $\mathbf{H}_{:,1}$ of acoustic and phoneme

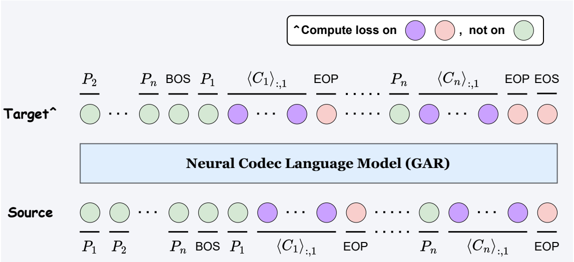

^Compute loss on ●, ● not on ●
Target^
P_2 P_n BOS P_1 $\langle C_1\rangle_{:,1}$ EOP P_n $\langle C_n\rangle_{:,1}$ EOP EOS
Neural Codec Language Model (GAR)
Source
P_1 P_2 P_n BOS P_1 $\langle C_1\rangle_{:,1}$ EOP P_n $\langle C_n\rangle_{:,1}$ EOP

Figure 2: The illustration of Generalized Autoregressive language model of ELLA-V.

tokens, structured as: $[P_1, P_2, \ldots, P_n,$ BOS, $P_1, \langle \mathbf{C}_1 \rangle_{:,1},$ EOP, $P_2, \langle \mathbf{C}_2 \rangle_{:,1},$ EOP, $\ldots, P_n, \langle \mathbf{C}_n \rangle_{:,1},$ EOP, EOS$]$. It is worth noting that the MFA (Montreal Forced Aligner) treats silence as a distinct phoneme, whereas our phoneme sequence $\mathbf{P}$ exclusively comprises phonemes other than silence. To clarify, we retain the acoustic component associated with silence but do not sandwich it with an EOP and a specific silence phoneme, nor do we use a silence phoneme in the *global advance* part.

We design a GAR language model to learn the continuation task on the aforementioned hybrid sequence, to generate the discrete acoustic code sequence $\mathbf{C}_{:,1}$. The GAR model consists of multiple Transformer decoder layers (Vaswani et al., 2017). After training, it can generate discrete audio codes for a specified text prompt and acoustic prompt. GAR is also responsible for predicting EOP and EOS to indicate the conclusion of a phoneme and the entire sentence, respectively.

The optimization of GAR is achieved by maximizing the likelihood of the acoustic part $\mathbf{C}_{:,1}$ of the hybrid sequence $\mathbf{H}_{:,1}$, as well as the special EOP and EOS tokens. Under forward factorization, this process is formulated as:

$$\begin{aligned}
&\max_{\theta_{GAR}} \ \log p(\tilde{\mathbf{C}}_{:,1} \mid \mathbf{P}; \theta_{GAR}) \\
&= \sum_{i=1}^{n} \sum_{t=0}^{l_i} \log p\big(\langle \tilde{\mathbf{C}}_i \rangle_{t,1} | \langle \tilde{\mathbf{C}}_i \rangle_{<t,1}, \langle \tilde{\mathbf{C}}_{<i} \rangle_{:,1}, \\
&\quad \mathbf{P}; \theta_{GAR}\big) \\
&= \sum_{\substack{t=0 \\ \mathbf{H}_{t,1} \neq \texttt{BOS} \\ \mathbf{H}_{t,1} \notin \{\mathbf{P}\}}}^{T_{\boldsymbol{H}}} \log p\left(\mathbf{H}_{t,1} \mid \mathbf{H}_{<t,1}; \theta_{GAR}\right)
\end{aligned}$$

where $\mathbf{H}$ has a size of $T_{\boldsymbol{H}} \times 8$, $\{\mathbf{P}\}$ denotes the phoneme set, $\langle \tilde{\mathbf{C}}_i \rangle$ is the concatenation of $\langle \mathbf{C}_i \rangle$ along with its broadcast trailing EOP and/or EOS tokens, $\tilde{\mathbf{C}}$ is then the concatenation of $\langle \mathbf{C}_i \rangle$, and θ_{GAR} represents neural network parameters of GAR model. The factorization of the training objective naturally encapsulates the core intuition of the GAR model: GAR generates the audio sequence phoneme-by-phoneme. GAR produces maximum likelihood predictions for each phoneme token successively, indicating the end of generating a specified phoneme by predicting EOP. Through *global advancement*, GAR can directly infer the next phoneme to be generated without relying on network predictions. After the prediction for the last phoneme is completed, GAR stops the generation process by predicting EOS. The generated sequence by GAR is **self-aligned**, as it can instantly know the corresponding position of any generated acoustic token in relation to the phoneme prompt.

During training, we apply a bidirectional mask to the phoneme sequence before BOS in the hybrid sequence, while a unidirectional mask is used for the part after BOS. We frame the training as a next-token-prediction language modeling task on the hybrid sequence. However, it's important to note that the model does not predict phonemes (or BOS). In other words, as shown in Figure 2, we only compute loss when the token to be predicted is not a phoneme (or BOS). During inference, whenever the model predicts an EOP for a phoneme, the next phoneme token is directly appended to the end of the sequence, as discussed in Section 4.

3.2.2 Non-Autoregressive (NAR) Codec Language Model

In the second stage, the NAR language model is employed to predict the codes from the second to the last quantization layers in parallel. The input-output sequence construction of the NAR model follows the same pattern as used in the GAR model discussed in Section 3.2.1. Specifically, the i-th column $\mathbf{H}_{:,i}$ of the hybrid sequence matrix $\mathbf{H}$ is structured as: $[P_1, P_2, \ldots, P_n$, BOS, $P_1, \langle \mathbf{C}_1 \rangle_{:,i}$, EOP, $P_2, \langle \mathbf{C}_2 \rangle_{:,i}$, EOP, $\ldots, P_n, \langle \mathbf{C}_n \rangle_{:,i}$, EOP, EOS]. And in practice if P_i represents the silence, $\mathbf{C}_{:,i}$ will not be sandwiched by P_i and EOP.

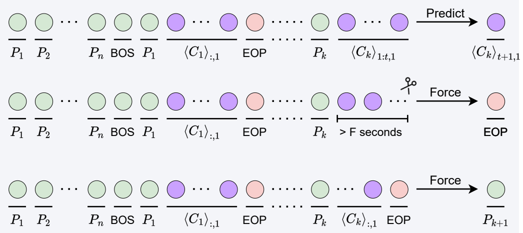

Predict
Force
Force
P_1
P_2
P_n
BOS
P_1
$\langle C_1 \rangle_{:,1}$
EOP
P_k
$\langle C_k \rangle_{1:t,1}$
$\langle C_k \rangle_{t+1,1}$
> F seconds
EOP
$\langle C_k \rangle_{:,1}$
P_{k+1}

Figure 3: The illustration of the inference process of ELLA-V.

The NAR model takes the previously generated hybrid sequence of the previous $j-1$ layers as input and predicts the codes of the j-th layer in parallel, formulated as:

$$\begin{aligned}
&\max_{\theta_{NAR}} \sum_{j=2}^{8} \log p(\mathbf{C}_{:,j} \mid \mathbf{H}_{:,<j}, \mathbf{P}; \theta_{NAR}) \\
&= \sum_{j=2}^{8} \sum_{\substack{t=0 \\ \mathbf{H}_{t,j} \in \{\mathbf{C}_{:,j}\}}}^{T_{\boldsymbol{H}}} \log p(\mathbf{H}_{t,j} \mid \mathbf{H}_{:,<j}, \mathbf{P}; \theta_{NAR})
\end{aligned}$$

where $\{\mathbf{C}_{:,j}\}$ denotes the acoustic token set of the j-th quantizer. In this formulation, The embeddings of tokens from the previous $j-1$ quantizers are summed up to feed the NAR model to predict the j-th layer. Intuitively, both the GAR and NAR model of ELLA-V compute the loss on the acoustic tokens of the target sequence, and GAR additionally computes loss for EOP and EOS.

3.3 Inference

ELLA-V can use a short clip of speech from an unseen speaker as an acoustic prompt to synthesize speech for a specified text prompt. Figure 3 illustrates the inference process of the GAR model. While VALL-E may get stuck in an infinite loop during inference, resulting in the synthesis of either *infinite silence* or repetitive pronunciation (Wang et al., 2023a), ELLA-V is capable of generating EOP and promptly truncating abnormally long phonemes. Following an EOP, we can directly append the next phoneme token to the end of the generated sequence, ensuring the proper generation of speech without abnormal pauses or repetitions. For the GAR model, we employ a sampling-based decoding strategy, whereas for the NAR model, we

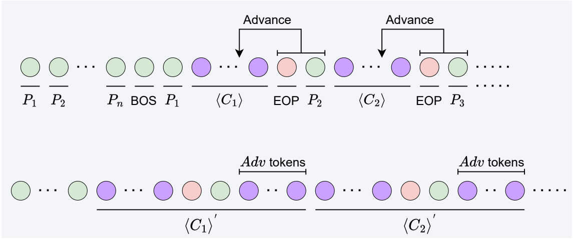


Figure 4: *Local advance*. A phoneme can locally have access to information about the next phoneme token advanced by Adv frames, allowing it to anticipate the upcoming phoneme token's characteristics.

use a greedy decoding approach to strike a balance between efficiency and performance.

3.4 Local Advance

One intuition is that the pronunciation of a phoneme is strongly related to the pronunciation of the phonemes just before and after it. However, due to the autoregressive nature of the GAR model, an acoustic token cannot attend to the following phoneme tokens, even though we can leverage the transformer's ability to model long-term dependencies through *global advance* to provide complete context for the acoustic token generation. To further harness the powerful capability of the transformer in modeling local dependencies, ELLA-V introduces an additional change in the sequence order based on Section 3.2. Specifically, we move the phoneme token and the EOP token ahead by a few frames, referred to as *local advance*.

4 Experiments

4.1 Experimental Setup

Data & Tasks: We trained ELLA-V using the publicly available Librispeech (Panayotov et al., 2015) 960h training dataset. We utilized Montreal Forced Aligner (MFA)[3] (McAuliffe et al., 2017) to obtain forced alignment information for the audio-transcription pairs. Sentences with unrecognized or unknown phones by MFA were excluded. The open-source 24kHz checkpoint[4] of EnCodec(Défossez et al., 2023) was used as the codec to generate discrete acoustic tokens. The LibriSpeech training data was upsampled to 24 kHz before feeding it into EnCodec.

In evaluating the model, two zero-shot TTS tasks were considered. For the zero-shot TTS continuation task, we adhered to methodologies established by previous works (Wang et al., 2023a; Le et al., 2023; Wang et al., 2023c), selecting examples ranging from 4 seconds to 10 seconds from the LibriSpeech test-clean dataset as our test set. In this task, we used the complete phoneme transcription as the text prompt and the first 3 seconds of the test audio sample as the acoustic prompt. The model was required to generate continuations.

For the zero-shot TTS cross-speaker task, we designed a hard case set comprising 100 hard sentences, as outlined in the demo page. These sentences included challenging phonetic patterns and unusual (abnormal) combinations of words that might pose difficulties for a TTS system to generate natural-sounding speech. In this case, we randomly picked 3-second sentences from the LibriSpeech test-clean set as the acoustic prompt. We then concatenated the transcription of this segment and the target phoneme sequence in the hard case set to form the text prompt. The model was tasked with cloning the voice of the speaker to say the specified hard target sentence.

Training configuration: For both GAR and NAR models, we stacked 12 Transformer decoder layers with an embedding dimension of 1024, a hidden state dimension of 1024, and a feed-forward layer dimension of 4096. All models were trained in parallel using 8 NVIDIA Tesla V100 GPUs with a batch size of 16384 tokens for GAR and 12288 tokens for NAR per GPU, respectively, learning a total of 320k steps. We used the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-9}$. We employed an inverse-sqrt learning rate scheduler with warm-up. For the first 32000 updates, we linearly increased the learning rate from 10^{-7} to a peak of 5×10^{-4}. The weight decay was 0.01.

Baseline: In this paper, we benchmarked the performance of zero-shot speech synthesis against VALL-E (Wang et al., 2023a). This system was originally trained on a substantial 60k hours of audio from the Librilight dataset (Kahn et al., 2020). To ensure a rigorous evaluation, we reproduced the VALL-E model and adapted it to train on the LibriSpeech 960h dataset. We also adjusted the model dimensions and the number of layers to match the parameter settings of ELLA-V and VALL-E. Both GAR (AR) and NAR models of VALL-E and ELLA-V have 154.3M parameters. Moreover, to mitigate potential bias introduced by the audio codec, we processed the authentic speech samples using EnCodec's encoder and decoder. We also include the result for Encodec reconstructed speech

[3] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

[4] https://github.com/facebookresearch/encodec

Table 2: Subjective and Objective performance comparison between ELLA-V and VALL-E on zero-shot TTS continuation task. † indicates that ground-truth audios were passed through the encoder and decoder of Encodec to evaluate the influence of neural audio codec.

Models	WER(%) (↓)	SPK (↑)	CMOS (↑)	SMOS (↑)
Ground Truth	1.41	0.923	0.29	4.39
Ground Truth-Encodec†	1.62	0.913	0.22	4.33
VALL-E	5.00	0.868	0.00	**3.56**
ELLA-V*(ours)*	**2.28**	**0.870**	**0.10**	**3.56**

for reference, denoted as Ground-Truth Encodec.

Evaluation Metrics: We evaluated our system with several objective metrics. Speaker similarity (SPK) and WER served as our primary measures. SPK was assessed using the fine-tuned WavLM-TDNN model[5] (Chen et al., 2022), scoring similarity on a scale of -1 to 1, with values above 0.86 indicate the same speaker identity (This value comes from the released model card page). The WER was determined by comparing the synthesized speech to the original text using the Conformer-Transducer model[6] (Gulati et al., 2020).

In addition to these standard metrics, we introduced two novel measures: INF% and CUT%. INF% quantified the frequency of generating infinitely long audio, indicative of a failure in synthesis. A higher INF% indicates poorer generation stability. In the practical implementation, generation that takes more than twice as long as the original is approximated as infinite generation. On the other hand, for ELLA-V, we forcibly truncate the synthesis of phonemes with a length greater than 0.4 seconds. CUT% is used to measure the frequency of forced cuts of phonemes in synthesis. For each objective metric, we reported average values over three experimental runs with different random seeds.

For subjective analysis, we relied on the mean opinion score (MOS). 30 test samples were chosen for this purpose, with each sample being evaluated by at least 15 listeners for aspects like naturalness and speaker similarity. The comparative mean option score (CMOS) and the similarity mean option score (SMOS) were the key subjective metrics used. SMOS was rated on a 1 to 5 scale, in 0.5-point increments, to gauge speaker similarity, while CMOS, ranging from -1 to 1, assessed the overall naturalness and quality of the synthesized speech against the baseline.

Table 3: WER comparison between ELLA-V and VALL-E on 100 particularly hard synthesis cases. Sub, Del, and Ins refer to Substitution, Deletion, and Insertion error rates, respectively.

Models	WER(%)	Sub(%)	Del(%)	Ins(%)
VALL-E	28.39	17.79	5.36	5.24
ELLA-V	**12.79**	**7.76**	**3.40**	**1.63**

Table 4: The ablation study to investigate the impact of global and local phoneme information.

Models	WER(%) (↓)	SPK (↑)
VALL-E	5.00	0.868
ELLA-V	**2.28**	**0.870**
ELLA-V-noglobal	5.00	0.859
ELLA-V-nophn	3.51	0.868

4.2 Results

Zero-Shot TTS continuation task. We present the evaluation results in Table 2, where a comparison between ELLA-V and VALL-E is shown. First, regarding speaker similarity, both subjective (SMOS) and objective (SPK) results indicate that ELLA-V and VALL-E performed similarly, which can be attributed to their shared backbone approach, combining (G)AR and NAR. Meanwhile, CMOS testing shows that ELLA-V achieved a +0.10 score, demonstrating a higher generation quality (i.e., naturalness) compared to VALL-E. Additionally, WERs calculated between the recognized text of synthesized audio and the ground-truth text show that ELLA-V is significantly better than VALL-E (2.28 versus 5.00). This underscores ELLA-V's enhanced capability in synthesizing higher-quality and more robust speech. Overall, ELLA-V substantially improved the synthesis accuracy and robustness of the language model-based TTS framework without affecting the naturalness and speaker similarity. This conclusion is not only corroborated by this easy continuation task, but also validated via the challenging synthesis sets in the subsequent section.

Zero-shot TTS cross-speaker task on hard cases. VALL-E utilized a traditional AR model that frequently resulted in alignment errors, including repetitions, transpositions, and omissions, particularly in more challenging synthesis cases (see Section 4.1 for details of the challenging synthesis set). Table 3 presents the WER comparison of VALL-E and ELLA-V on the 100 particularly hard synthesis sentences. In contrast to VALL-E, ELLA-V demonstrates markedly lower WER, signifying its enhanced robustness. This substantial reduction in errors translates to more accurate and reliable voice

[5] https://huggingface.co/microsoft/wavlm-base-plus-sv

[6] https://huggingface.co/nvidia/stt_en_conformer_transducer_xlarge

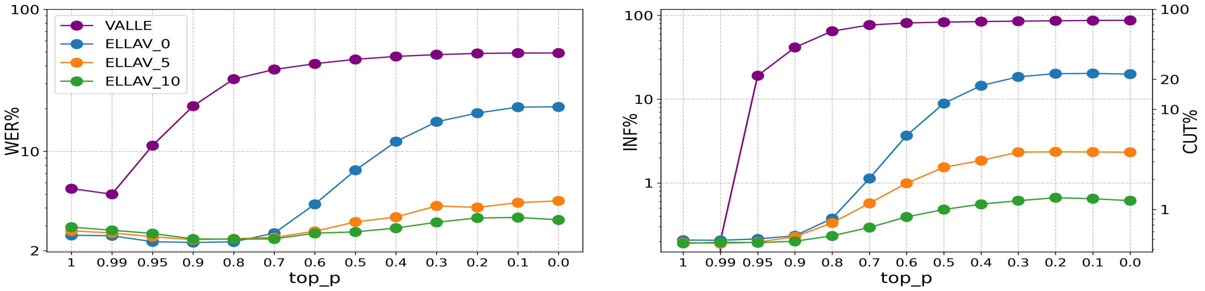


Figure 5: Ablations on decoding strategies. The figures demonstrate the trends of three metrics, INF (for VALL-E), CUT (for ELLA-V), and WER (for both), with respect to the variations in top_p in nuclear sampling.

synthesis applications, significantly improving user experience in real-world scenarios.

Regarding VALL-E's tendency to fall into infinite silence, an intuitive explanation is that the silence patterns in the training data are relatively simple and many of them are repetitive. In this case, a traditional language model is prone to overfitting to these patterns. During testing, when the model encounters silence, it assigns a high probability to silence. This leads to issues such as beam search, which is based on maximum likelihood, getting stuck in a loop. However, ELLA-V does not face this problem.

Analysis of Decoding Strategies. To demonstrate the stability of ELLA-V under different decoding strategies, we conducted an ablation study, testing the decoding performance with different top-p values for nuclear sampling, by varying $p \in \{1, 0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0(\text{greedy})\}$. The results are shown in Figure 5. We can observe that as top_p decreases, the accuracy of VALL-E's synthesized speech significantly decreases. At this point, VALL-E is more prone to generating a large number of overfit silence tokens, leading to a significant increase in INF%. And compared to VALL-E, the audio synthesized by ELLA-V is less sensitive to rate changes in the top_p sampling strategy, whose WER consistently outperforms VALL-E. When the local advance is set to 5 or 10 tokens, the generated audio exhibits significant stronger robustness. On the other hand, as shown in Figure 5 (right), as top_p decreases, VALL-E tends to get stuck in infinite loops of failed generation, while the generation of ELLA-V remains significantly stable. Moreover, ELLA-V can promptly handle (truncate) the synthesis of exceptional phonemes, resulting in significantly higher robustness.

Ablation Study. In this paragraph, we conduct ablation experiments. (1) To investigate the impact of global phoneme information on synthesized speech, we removed the global phoneme sequence at the beginning of the trained sequence (abbr. **ELLA-V**-noglobal). (2) To investigate whether it is necessary to provide the specific phoneme token before its corresponding acoustic tokens during both training and inference, rather than just using the EOP separator, we removed all phoneme tokens following BOS in the mixed sequence (abbr. **ELLA-V**-nophn). The experimental results are shown in Table 4. It is observed that the accuracy of synthesized speech significantly deteriorated either when global phoneme tokens were not used or when local phoneme tokens were disabled within the hybrid sequence. It is also notable that even in the absence of global advance (i.e., in the ELLA-V-noglobal configuration), the SPK and WER of the synthesized audio were comparable to those of VALL-E. These findings indicate the importance of both local and global information in achieving more accurate synthesized audios, meanwhile, combining both of them potentially leads to further enhancements in accuracy.

5 Conclusion

In this paper, we introduce ELLA-V, a simple and efficient two-stage zero-shot TTS framework based on language modeling. By learning interleaved sequences of acoustic and text tokens, our proposed GAR model can provide fine-grained control over synthesized audio at the phoneme level and can better leverage local dependencies to predict the pronunciation of the current phoneme. Experimental results demonstrate that ELLA-V achieves higher accuracy and more stable results under different threshold top-p for nuclear sampling. We aspire for this work to advance research in enhancing the robustness of speech generation.

Limitations

Scalability. ELLA-V was trained on the publicly available LibriSpeech (Panayotov et al., 2015) dataset, which comprises 960 hours of English speech. We were unable to scale up the model and data, and we hope that future work can test a larger scale. In a similar vein, we have not validated whether ELLA-V performs well on languages other than English. Additionally, we did not explore the impact of training audio quality on the model. Generally, low-quality speech audio containing noise and interference can disrupt the model's understanding of speech signals, hindering model training. Further experiments are still needed to determine the sensitivity of ELLA-V to the quality of training data.

Expressiveness. ELLA-V primarily focuses on improving the accuracy of synthesized speech but lacks means to control the emotional aspects of synthesized speech. Specifically, ELLA-V can only implicitly maintain environmental and emotional information through audio prompts but cannot explicitly and accurately specify the emotion of synthesized speech. Future work will explore this further.

Ethics Statement

This paper introduced a zero-shot TTS model based on language modeling approach, named ELLA-V. The model can maintain the speaker identity of any unseen speaker and synthesize specified speech. Due to this capability, ELLA-V has potential risks of misuse, such as synthesizing fraudulent information, hate speech, or other harmful content. To minimize these potential negative impacts, we anticipate that the next step in our research direction will be to efficiently add watermarks to distinguish whether the audio is generated by the generative model.

References

Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, et al. 2017. Deep Voice: Real-time neural text-to-speech. In *Proc. ICML*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, et al. 2023. AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, et al. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, et al. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High fidelity neural audio compression. *Transactions on Machine Learning Research*.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, et al. 2023. Decoder-only or encoder-decoder? Interpreting language model as a regularized encoder-decoder. *arXiv preprint arXiv:2304.04052*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, et al. 2020. Conformer: Convolution-augmented Transformer for speech recognition. In *Proc. INTERSPEECH*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proc. NeurIPS*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, et al. 2019. The curious case of neural text degeneration. In *Proc. ICLR*.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, et al. 2019. Hierarchical generative modeling for controllable speech synthesis. In *Proc. ICLR*.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, et al. 2023. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *Proc. ICML*.

Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, et al. 2022. ProDiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proc. ACM MM*.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, et al. 2019. Comparison of diverse decoding methods from conditional language models. In *Proc. ACL*.

Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, et al. 2021. Diff-TTS: A denoising diffusion model for text-to-speech. In *Proc. Interspeech*.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *Proc. ICASSP*. IEEE.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, et al. 2023. Speak, Read and Prompt: High-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540*.

Heeseung Kim, Sungwon Kim, and Sungroh Yoon. 2022. Guided-TTS: A diffusion model for text-to-speech via classifier guidance. In *Proc. ICML*. PMLR.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. 2020. Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. *Proc. NeurIPS*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. In *Proc. NeurIPS*.

Yoonhyung Lee, Joongbo Shin, and Kyomin Jung. 2022. Bidirectional variational inference for non-autoregressive text-to-speech. In *Proc. ICLR*.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, et al. 2023. Flow matching for generative modeling. In *Proc. ICLR*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, et al. 2017. Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In *Proc. Interspeech*.

Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, et al. 2020. Flow-TTS: A non-autoregressive network for text to speech based on flow. In *Proc. ICASSP*. IEEE.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, et al. 2017. WaveNet: A generative model for raw audio. In *Proc. ICML*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. LibriSpeech: An ASR corpus based on public domain audio books. In *Proc. ICASSP*. IEEE.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, et al. 2021. Grad-TTS: A diffusion probabilistic model for text-to-speech. In *Proc. ICML*, pages 8599–8608. PMLR.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. WaveGlow: A flow-based generative network for speech synthesis. In *Proc. ICASSP*, pages 3617–3621. IEEE.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, et al. 2022. Hierarchical text-conditional image generation with CLIP latents. *arXiv e-prints*, pages arXiv–2204.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, et al. 2019. FastSpeech: Fast, robust and controllable text to speech. *Proc. NeurIPS*, 32.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, et al. 2023. AudioPaLM: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, et al. 2023. NaturalSpeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. ICML*. PMLR.

Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models. *Proc. NeurIPS*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, et al. 2022. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. 2017. Attention is all you need. *Proc. NeurIPS*, 30.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, et al. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Jiaming Wang, Zhihao Du, Qian Chen, Yunfei Chu, et al. 2023b. LauraGPT: Listen, attend, understand, and regenerate audio with GPT. *arXiv preprint arXiv:2310.04673*.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, et al. 2023c. SpeechX: Neural codec language model as a versatile speech transformer. *arXiv preprint arXiv:2308.06873*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, et al. 2017. Tacotron: Towards end-to-end speech synthesis. In *Proc. INTERSPEECH*.

Shoule Wu and Ziqiang Shi. 2022. ItôWave: Itô stochastic differential equation is all you need for wave generation. In *Proc. ICASSP*. IEEE.

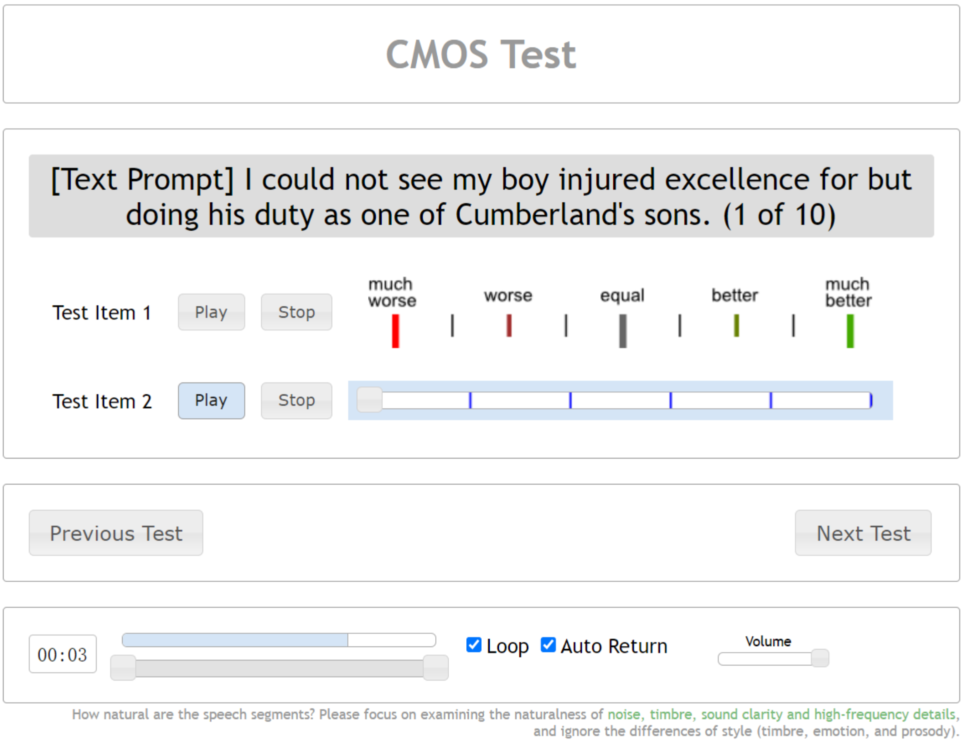
Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, et al. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Proc. NeurIPS*.

Heiga Zen, Keiichi Tokuda, and Alan W Black. 2009. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064.

A Evaluation

A.1 Subjective Evaluation

For CMOS, listeners are instructed to compare a pair of speech segments and indicate which of the two sentences is more natural and of higher quality. The relative scoring range is from -1 to 1. For SMOS, listeners are required to assess the similarity between the synthesized speech segment and the speaker of the audio prompt. The scoring range is from 1 to 5. The screenshots of instructions for testers are shown in 6 and 7, respectively. We paid $8 to participants hourly and totally spent about $600 on participant compensation.



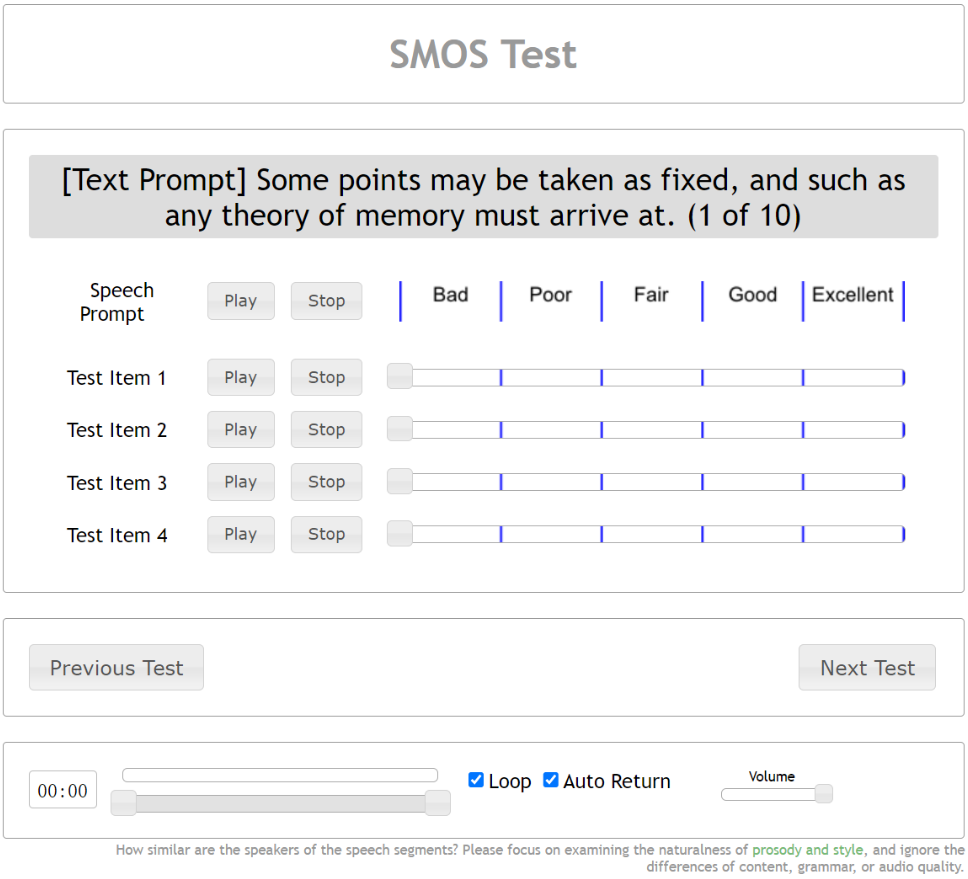
CMOS Test

[Text Prompt] I could not see my boy injured excellence for but doing his duty as one of Cumberland's sons. (1 of 10)

00:03 ☑ Loop ☑ Auto Return

How natural are the speech segments? Please focus on examining the naturalness of noise, timbre, sound clarity and high-frequency details, and ignore the differences of style (timbre, emotion, and prosody).

Figure 6: Screenshot of CMOS testing.



SMOS Test

[Text Prompt] Some points may be taken as fixed, and such as any theory of memory must arrive at. (1 of 10)

00:00 ☑ Loop ☑ Auto Return

How similar are the speakers of the speech segments? Please focus on examining the naturalness of prosody and style, and ignore the differences of content, grammar, or audio quality.

Figure 7: Screenshot of SMOS testing.