Continuous Speech Tokenizer in Text To Speech

Anonymous ACL submission

Abstract

The fusion of speech and language in the era of large language models has garnered significant attention. Discrete speech token is often utilized in text-to-speech tasks for speech compression and portability, which is convenient for joint training with text and have good compression efficiency. However, we found that the discrete speech tokenizer still suffers from information loss. Therefore, we propose a simple yet effective continuous speech tokenizer and a text-to-speech model based on continuous speech tokens. Our results show that the speech language model based on the continuous speech tokenizer has better continuity and higher estimated Mean Opinion Scores (MoS). This enhancement is attributed to better information preservation rate of the continuous speech tokenizer across both low and high frequencies in the frequency domain.

1 Introduction

Autoregressive modeling is a common method for processing language sequences and is effective in token prediction (Vaswani, 2017; Radford, 2018; Yang et al., 2024). The success of large language models continues to inspire and develop this field (Tay et al., 2020; Dao et al., 2022; Zhuang et al., 2023). Transformer-based models are becoming more and more prominent due to the interest in speech language models utilized to understand and generate speech (Borsos et al., 2023; Wang et al., 2023a; Rubenstein et al., 2023; Wang et al., 2024). These models are designed to solve a range of speech-related tasks such as Text-to-Speech (TTS) (Wu et al., 2024) and Automatic-Speech-Recognition (ASR) (Kheddar et al., 2024). A significant difference between the text and speech domains is the discrete nature of text tokens versus the continuous nature of speech, whereas a commonality of language models is the utilization of discrete speech representations (Wang et al., 2023b; Yang et al., 2023b; Wu et al., 2024). The current discrete speech representation can be divided into two types: tokenizer-based compressed tokens such as VALL-E (Wang et al., 2023a), MusicLM (Agostinelli et al., 2023), CosyVoice (Du et al., 2024), and semantic tokens such as AudioLM (Borsos et al., 2023). The tokenizer-based model utilizes residual vector quantization (RVQ) and hierarchical quantizers to compress speech tokens, and semantic tokens are usually based on self-supervised pre-trained models. Many works have been studying the use of speech tokens, but the best speech representation remains unresolved (Défossez et al., 2022; Yang et al., 2023a; Zhang et al., 2023).

We observe that discrete speech tokenizers have potential information loss in text-to-speech tasks, which is reflected in different degrees of loss in low-frequency and high-frequency components. Therefore, we utilized a speech encoder that models **continuous speech vectors** for TTS. Based on this, we conducted experiments to demonstrate the advantages of continuous speech tokens over discrete versions. Continuous speech tokens have better information retention than discrete tokens in all frequency bands and better robustness to the sampling rate in text-to-speech tasks. Specifically, our contributions include:

1. We proposed a continuous speech tokenizer based text-to-speech model and provided a comprehensive training framework.

2. We conducted experiments to demonstrate the effectiveness of our approach in the text-to-speech task and showed the significance of the training pipeline.

3. We verified the advantages of the continuous speech tokenizer over the discrete version, including the retention of high-frequency information and the robustness of the sampling rate.

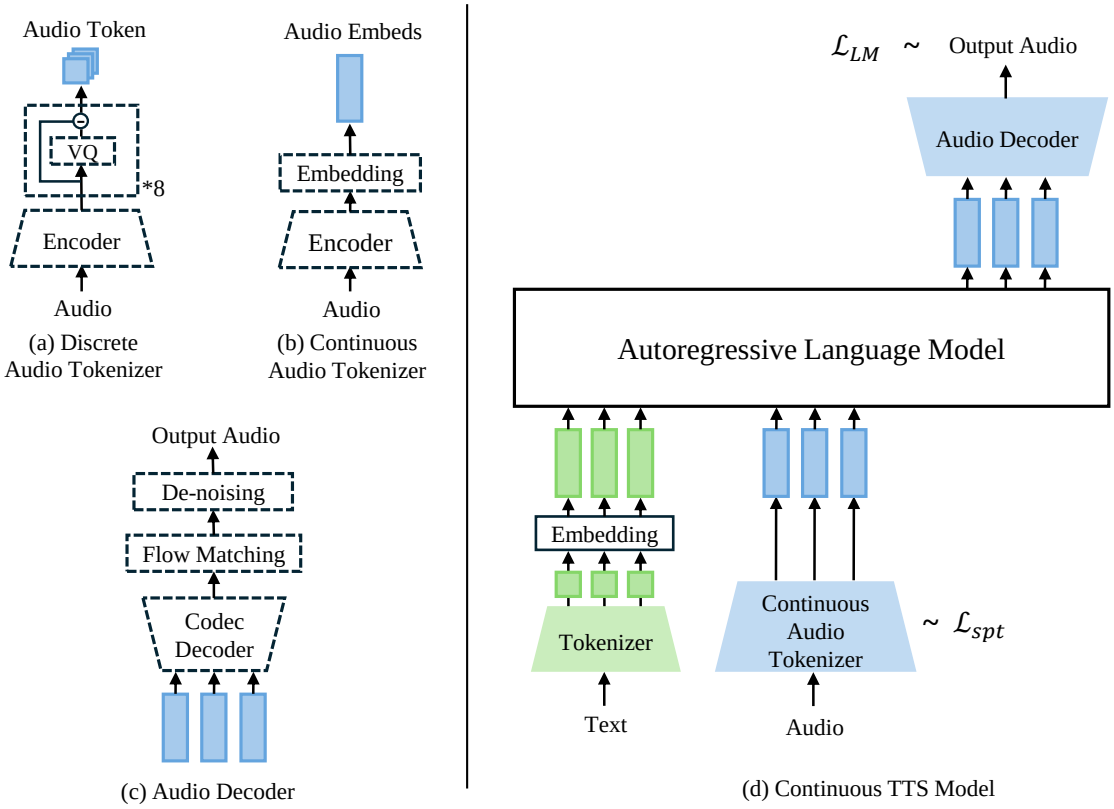

Figure 1: Continuous Text-to-speech Model. (a) Traditional audio tokenizer structure, using preset or trainable code table for RVQ. (b) Continuous audio tokenizer, replace RVQ with an embedding module to generate continuous speech token representations. (c) Audio decoder, consisting of codec decoder, flow-matching and de-noising modules. (d) Overall continuous TTS model structure.

2 Method

2.1 Continuous Speech Tokenizer

Most contemporary speech language models utilize the RVQ quantizer (Défossez et al., 2022; Wang et al., 2023a; Agostinelli et al., 2023) as a basic building block to accomplish the process of speech discretization. As illustrated in Figure 1a, the quantizer usually contains multiple codebooks that convert speech features into discrete tokens. The quantization process usually involves distance grounding based on the content within the codebook, which suffers from great information loss.

We propose to adopt a **continuous speech tokenizer** for text-to-speech tasks. After the input audio is resampled at 24 kHz, it is passed through the encoder to obtain the speech feature vector. In the discrete version, this step is typically followed by RVQ to generate discrete tokens. In contrast, our continuous speech tokenizer retains the audio resampling step and integrates the encoder layer to obtain continuous speech tokens, and then utilizes the output continuous speech tokens directly as the input for the language model.

Figure 1b shows the continuous speech tokenizer. The decoder corresponds to the encoder, which is utilized to decode the output of the autoregressive language model. The two can be pre-trained together in a VAE-like form, and our experiments show that this is beneficial for TTS tasks.

2.2 Continuous Speech Tokenizer Based TTS

We propose a TTS model based on the continuous speech tokenizer and regards the TTS task as an autoregressive token generation task. Different from predicting discrete speech codes, our method predicts continuous speech tokens. Given the input speech $\mathbb{A}$ and text $\mathbb{T}$, we calculate the continuous speech token and text embedding respectively,

$$\mathcal{A} = \text{Cont-SpeechTokenizer}\left(\mathbb{A}\right). \quad (1)$$

where $\mathcal{A}$ denotes for the continuous speech token. Text processing includes text tokenizer and language model embedding layer,

$$x = \text{Text Tokenizer}\left(\mathbb{T}\right), \mathcal{T} = \text{Embeds}\left(x\right). \quad (2)$$

The speech token and text embedding are concatenated together as the input for the language model during autoregressive generation. After generating the predicted continuous speech token, we utilize the audio decoding layer and flow matching to convert the audio.

2.3 Flow Matching

We utilize the optimal-transport conditional flow matching (OT-CFM) following the codec decoder to generate a speech mel-spectrogram based on the generated continuous speech features. By leveraging the idea of optimal transport, CFM can be constructed to generate an ordinary differential equation (ODE) with simple vector fields.

Moreover, de-noising module is added for final output adjustment, consisting of frequency limiting and off-the-shelf denoising model.

2.4 Training Objective

2.4.1 Speech Tokenizer Reconstruction Loss

For the pre-training of the speech tokenizer, we use a training method similar to VAE, connecting the continuous speech tokenizer with the speech decoder for overall training. Given the training audio $\mathbb{A}$ and the corresponding text label $\mathbb{Y}$, the calculation process is as follows,

$$\widehat{\mathbb{A}} = \text{AD}\left(\text{Cont-SpeechTokenizer}\left(\mathbb{A}\right)\right). \quad (3)$$

where AD refers to AudioDecoder. And the estimated audio-to-text is calculated by the ASR decoder:

$$\widehat{\mathbb{Y}} = \text{ASR-Decoder}\left(\widehat{\mathbb{A}}\right). \quad (4)$$

The reconstruction loss contains two parts, the similarity loss between the regenerated audio $\widehat{\mathbb{A}}$ and the original signal $\mathbb{A}$, and the CTC loss between the decoded ASR results $\widehat{\mathbb{Y}}$ and the text label $\mathbb{Y}$.

$$\begin{aligned}\mathcal{L}_{spt} &= \mathcal{L}_{rec} + \mathcal{L}_{ASR} \\ &= \text{SIM}\left(\mathbb{A}, \widehat{\mathbb{A}}\right) + \text{CTC}\left(\mathbb{Y}, \widehat{\mathbb{Y}}\right).\end{aligned} \quad (5)$$

2.4.2 Language Modeling Loss

In the language model training part, we also activate the speech continuous tokenizer as a training state and train it together with the language model, while these two components are updated with different learning rates. Given the model output $\mathcal{O}$ and the speech label $\mathbb{M}$, the speech label is first encoded into continuous speech tokens, and the loss function is calculated by the distance between the continuous speech token of the label and the model output.

$$\mathcal{M} = \text{Cont-SpeechTokenizer}\left(\mathbb{M}\right). \quad (6)$$

$$\mathcal{L}_{LM} = \text{MSE}\left(\mathcal{O}, \mathcal{M}\right). \quad (7)$$

2.4.3 Training Pipeline

The training consists of two steps, the pre-training of the tokenizer and the joint training of the entire model. First, the tokenizer encoder is connected to the decoder and pre-trained as described in 2.4.1. In the second step, the decoder is frozen, and the entire model is jointly trained as outlined in 2.4.2. During this step, the learning rate of the tokenizer is set to 0.05 of that of the language model.

3 Experiments

3.1 Experiment Setup

Datasets. For training, we utilize the LibriSpeech (Panayotov et al., 2015) dataset, which contains a corpus of approximately 1,000 hours of English speech. We use the dev-clean subset for validation and the test-clean subset for evaluation.

Model Settings. Following VALL-E (Wang et al., 2023a), we construct the continuous speech tokenizer based TTS model, while the number of model blocks are set to the default values.

Baselines. For the speech tokenizer experiments, we use the commonly used discrete speech tokenizer Encoder as the baseline. For the TTS model experiment, we use the following models as the baseline: VALL-E (Wang et al., 2023a), MELLE (Meng et al., 2024).

Evaluation Metrics. For the speech tokenizer, we utilize an estimated transfer function to show the ability of the speech tokenizer in preserving information. For the TTS model, we employ the following metrics: (1) WER (Word Error Rate), (2) SIM (Speaker Similarity), (3) EMoS (Estimated Mean Opinion Score), (4) CLVP Score, (5) STOI, (6) Noisiness, Continuity, Loudness Quality, and Naturalness. Details of the metrics are shown in the Appendix A.1.

3.2 TTS Evaluation

In this section we evaluate and compare the results of the continuous speech tokenizer based TTS model. Our text-to-speech model based on continuous speech tokens achieves better performance in various aspects and shows its value in TTS tasks.

Model \ Metrics	WER ↓	SIM ↑
Origin	2.61	0.86
VALL-E	12.73	0.53
MELLE	7.26	0.67
Ours	**6.59**	**0.73**

Table 1: Evaluation of WER (%) and SIM on the LibriTTS.

Metrics	VALL-E	Cont-SPT
EMoS ↑	0.83	**1.32**
CLVP ↓	3.52	**2.94**
STOI ↑	0.20	**0.42**
Noisiness Quality ↑	1.27	**1.82**
Continuity Quality ↑	1.80	**3.61**
Loudness Quality ↑	1.56	**1.65**
Naturalness ↑	1.31	**1.45**

Table 2: Evaluation of speech generation quality. Cont-SPT that based on continuous speech tokens achieved higher speech quality than baseline model based on discrete tokens.

SPT \ Freq	2k	5k	8k
Discrete	0.95	0.78	0.34
Continuous	0.94	0.81	0.55

Table 3: The retention rate of audio after the tokenizer at different frequencies. The continuous speech tokenizer has high information retention in all frequency bands, especially in the high frequency part.

Table 1 shows the comparison between our method and the baseline, and our method outperforms the baseline in most cases. More detailed evaluation indicators shown in Table 2 elucidate the specific advantages of our method. In the evaluation of relative indicators including CLVP Score, STOI, and absolute indicators such as noisiness quality and continuity quality, the model based on continuous speech tokenizer exceeds the baseline. The ablation study is shown in Appendix A.2.

3.3 Analysis on Different Frequencies

We are interested in how continuous speech tokenizers can help TTS models improve their performance. As mentioned above, the number of tokens in the codebook of discrete tokenizers is limited, which constrains the space that discrete tokenizers can represent to be limited and increases the information loss. In this section, we demonstrate the advantages of continuous speech tokenizers in retaining information by studying the transfer function of speech tokenizers.

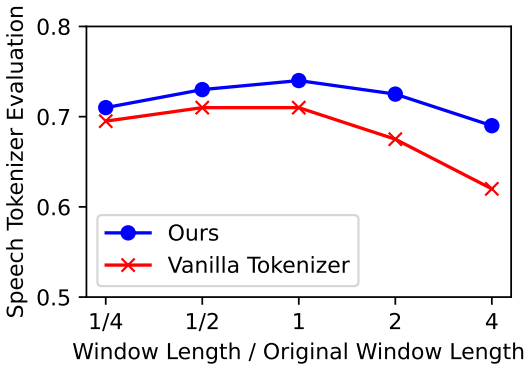

Figure 2: Evaluation of modifying window length and sampling rate utilizing the average attention rate after the tokenizer.

Table 3 presents the estimated transfer functions of the continuous and discrete speech tokenizers across various frequency ranges. It can be seen that in the high-frequency range, the transmission effect of the discrete speech tokenizer drops sharply, while the continuous speech tokenizer maintains a good effect in this range. In fact, the continuous speech tokenizer has a great information retention ability in the entire frequency range.

3.4 Sampling Rate and Window Length Robustness

We further study the robustness to different sampling rates. Due to the presence of narrowband data, robustness to sampling rate is also worth noting. To ensure the consistency of input dimensions, we modify the sampling rate and window length synchronously. As shown in Figure 2, the continuous speech tokenizer is more robust than the discrete tokenizer, especially when the window length ratio is larger. This is due to the better information preservation of continuous speech tokenizer, and this conclusion is consistent with Section 3.3.

4 Conclusion

In this paper, we propose a method based on continuous speech tokenizer in TTS and demonstrate its effectiveness through experiments. In addition, we conduct a series of experiments on continuous and discrete speech tokenizers to illustrate the advantages of continuous tokenizer in terms of information retention and robustness. Finally, the TTS model based on continuous speech token proposed in this paper provides a complete framework and a foundation for subsequent research.

Limitations

Although we proposed a novel continuous speech token based approach in the TTS field and compared it with discrete methods, there are limitations for future studies. Our work focuses on TTS tasks as well as joint training of speech and text, and has not yet been verified and evaluated on the Multimodal Large Language Model (MLLM) (Zhang et al., 2024). The addition of multiple modalities brings challenges. The continuous speech tokenizer proposed in our work could improve the information carrying capacity, but it may also bring difficulties of training. Future research will continue to explore these unexplored aspects.

References

A wrapper around speech quality metrics mosnet, bsseval, stoi, pesq, srmr, sisdr. [Online]. https://github.com/aliutkus/speechmetrics.git.

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

J ames Betker. 2022. TTS-scores.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Beskrovnyi Ivan. 2024. nisqa-s. https://github.com/deepvk/nisqa-s.

Hamza Kheddar, Mustapha Hemis, and Yassine Himeur. 2024. Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*, page 102422.

Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. 2019. Mosnet: Deep learning based objective assessment for voice conversion. *arXiv preprint arXiv:1904.08352*.

Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. 2024. Autoregressive speech synthesis without vector quantization. *arXiv preprint arXiv:2407.08551*.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. 2021. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *Interspeech 2021*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4214–4217. IEEE.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *ACM Computing Surveys*, 55:1 – 28.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

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.

Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023b. Viola: Unified codec language models for speech recognition, synthesis, and translation. *arXiv preprint arXiv:2305.16107*.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. 2024. Speechx: Neural codec language model as a versatile speech transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Haibin Wu, Xuanjun Chen, Yi-Cheng Lin, Kai-wei Chang, Ho-Lam Chung, Alexander H Liu, and Hung-yi Lee. 2024. Towards audio language modeling-an overview. *arXiv preprint arXiv:2402.13236*.

Yi-Chiao Wu, Israel D Gebru, Dejan Marković, and Alexander Richard. 2023. Audiodec: An open-source streaming high-fidelity neural audio codec. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. 2023a. Hifi-codec: Group-residual vector quantization for high fidelity audio codec. *arXiv preprint arXiv:2305.02765*.

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

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. 2023. A survey on efficient training of transformers. *arXiv preprint arXiv:2302.01107*.

A Experiments Illustration

A.1 Metrics

In this section, we elucidate the details of the metrics used in this paper.

WER The Word Error Rate is caculated by performing ASR using Whisper Large (Radford et al., 2023) on the generated speech.

SIM The Speaker Similarity is obtained by first calculating the speaker embeddings of both the generated speech and the ground truth speech using WavLM-TDNN (Chen et al., 2022), and then calculating the cosine similarity between the embeddings.

EMoS The Mean Opinion Score is a commonly used indicator for evaluating TTS quality. We use model-estimated EMoS (Lo et al., 2019) to provide a more general evaluation, and the open source library we utilize is located at (SPE).

CLVP Score The CLVP Score measures the distance predicted by Contrastive Language-Voice Pretrained model between text and an audio clip where that text is spoken, and the open source library we utilize is located at (Betker, 2022)

STOI The Short-Time Objective Intelligibility (Taal et al., 2010) is used to reflect the clarity of speech, ranging from 0 to 1. The open source library we utilize is located at (SPE).

Noisiness, Continuity, Loudness Quality, and Naturalness The Noisiness, Continuity, Loudness, and Naturalness are calculated by NISQA (Mittag et al., 2021) model, and the open source library we utilize is located at (Ivan, 2024)

A.2 Ablation Study

Model \ Metrics	WER ↓	SIM ↑
Ours	**6.59**	**0.73**
w/o SPT Joint Training	6.83	0.71
w/o $\mathcal{L}_{spt}$ Pre-train	8.42	0.64

Table 4: Ablation study of our model.

We conduct the ablation study on our method. Table 4 shows the effect of continuous speech tokenizer pre-training, the impact of speech tokenizer joint training. Our method provides a complete and usable framework for continuous speech tokenizer and continuous speech token based TTS.

B Supplement to Related Work

In this section, we briefly supplement the related work on speech tokenizers in the TTS field. Since

the currently commonly used Encodec (Défossez et al., 2022) was proposed, many works have studied the form and properties of tokenizers such as Audiodec (Wu et al., 2023). AudioLM (Borsos et al., 2023) propose a hybrid tokenization scheme to achieve the combination of reconstruction quality and long-term structure. Hifi-codec (Yang et al., 2023a) propose the group-residual vector quantization (GRVQ) technique. SpeechTokenizer (Zhang et al., 2023) proposed to unify semantic and acoustic tokens by combining semantic distillation in the RVQ process. Concurrent work MELLE (Meng et al., 2024) proposed a TTS model based on mel-spectrogram feature extraction and vocoder. It is worth noting that MELLE also questioned the concept of discrete speech tokens. Unlike us, they removed the speech tokenizer. We retained this concept and conducted pre-training and ablation experiments on the continuous speech tokenizer.