

Improving Speech Generation via Multi-Neighbor Token Prediction for Neural Codec Language Models

Wenrui Liu¹, Qian Chen², Wen Wang², Yuxuan Wang¹, Haoxiang Wu¹,
Guanrou Yang³, Rui-Chen Zheng², Weiqin Li², Xiangang Li², Guo Tao⁴, Zemin Liu^{†1}

¹Zhejiang University, ²Alibaba Token Hub, Alibaba Group,

³Shanghai Jiao Tong University, ⁴Inspur Data Technology

{liuwenrui, liu.zemin}@zju.edu.cn

Abstract

Inspired by Large Language Models (LLMs), codec-based TTS systems have achieved remarkable success in high-fidelity speech generation by directly adopting the standard Cross-Entropy (CE) objective to predict sequences of discrete speech tokens from neural audio codecs. However, this practice overlooks the geometric structure of codec token spaces: tokens that are close in the codec codebook embedding space often correspond to acoustically similar speech realizations and can serve as perceptually plausible alternatives. We term this observation the **Neighboring Token Phenomenon**. The one-hot CE objective is agnostic to such structure, as it penalizes all non-target tokens equally, including these close neighbors, creating a mismatch between the structured codec latent space and the learning objective. In this paper, we quantitatively analyze the neighboring token phenomenon in both reconstruction and generation settings. To address the mismatch, we propose **Multi-Neighbor Cross-Entropy (MNCE)** loss function, which generalizes the one-hot target of CE loss into a positive set containing the ground-truth token and its nearest neighbors, encouraging the model to treat neighbor predictions as partially correct. As a result, even when the model misses the exact target token, it is more likely to predict a close neighbor rather than an unrelated token, reducing perceptually harmful errors in synthesized speech. Experiments demonstrate that MNCE is effective and generalizable across diverse codec-based TTS systems, improving quality and robustness. Moreover, scaling training to 20,000 hours further validates the scalability of our approach. The demo of audio samples is available online ¹.

1 Introduction

Large Language Models (LLMs) have recently propelled codec-based Text-to-Speech (TTS) (Chen et al., 2025a; Wang et al., 2024; Du et al., 2024; Ye et al., 2025) to generate remarkably human-like speech, establishing it as a key component in voice interaction (Zhang et al., 2023a; Défossez et al., 2024), omni-modal LLMs (Li et al., 2025; Xu et al., 2025), and spoken dialogue systems (Ju et al., 2025; Yu et al., 2025). The success of this paradigm hinges on two modules: a neural audio codec (Défossez et al., 2022; Ji et al., 2024; Jiang et al., 2025) encodes audio into discrete speech tokens, and a neural codec language model (Chen et al., 2025a; Wang et al., 2024; Ye et al., 2025; Du et al., 2024) initialized from a pretrained LLM (Yang et al., 2025; Dubey et al., 2024) is then trained to predict the speech tokens. The codec decoder then converts predicted speech tokens into a waveform. This representation of speech as a sequence of discrete indices relies on the assumption that speech tokens are analogous to text tokens (Chen et al., 2025a; Ye et al., 2025), which allows for the seamless integration with powerful text-based LLMs. Consequently, the training objective of the LLM, the typical Cross-Entropy (CE) loss, is directly applied in speech generation tasks.

However, this paradigm overlooks a vital distinction between the speech and text modalities. In text processing, a vocabulary maps text tokens into independent indices (Hu et al., 2025), which do not encode geometric relationships in the numerical values of the indices. In contrast, speech tokens are generated through vector quantization (Défossez et al., 2022; Zeghidour et al., 2021), a process that endows the tokens with a rich geometric relationship (Shuster and Nachmani, 2025). The index of each speech token corresponds to a specific vector within a learned codebook. Since this codebook resides in a continuous embedding space, every

[†]Corresponding author.

¹<https://multineighbortokenprediction.github.io/MultiNeighborTokenPrediction>

vector is surrounded by other vectors at varying Euclidean distances (Van Den Oord et al., 2017). This geometric structure indicates that every speech token has **neighboring tokens**: other tokens with the corresponding vectors that are close to the vector of the certain token in the embedding space, which in turn suggests feature similarity. Formally, we define the neighbors of a speech token as the K codebook entries with the top- K smallest Euclidean distance in the codebook embedding space.

Building on this insight, we investigate the impact of neighboring tokens across popular neural audio codecs. We first conduct a series of substitution experiments in the speech reconstruction task (Défossez et al., 2022; Liu et al., 2025b): in a sequence of speech tokens encoded from input speech, a given target token is replaced with one of its nearest neighbors via multinomial sampling, before being decoded back into audio. Our findings reveal that replacing a token with its nearest neighbors typically causes only minor degradation in reconstructed speech quality. We term this observation the **Neighboring Token Phenomenon**. This observation provides empirical evidence that codec neighbors are acoustically similar; moreover, ASR evaluation results also demonstrate that such neighbor substitutions often preserve the recognized content. Furthermore, our quantitative analyses confirm neighboring token phenomenon is a widespread phenomenon, evident not only in the neural audio codecs but also in the neural codec language models trained upon them.

Notably, the neighboring token phenomenon reveals a fundamental mismatch in how language models (Casanova et al., 2024; Du et al., 2025) are trained on speech tokens, a process that directly adopts the CE loss from text generation (Dubey et al., 2024; Yang et al., 2025). The CE loss enforces a one-hot objective, designating only the single ground-truth token as the target while treating acoustically similar neighbors and entirely unrelated tokens with the same degree of penalty, completely ignoring the inherent geometric structure of the speech token space. This mismatch between the learning objective and the geometry of speech token space hinders the model’s ability to learn more robust and accurate speech representations.

To alleviate this mismatch, we propose a novel training objective, **Multi-Neighbor Cross-Entropy (MNCE) loss**, for training the neural codec language models in codec-based TTS. Unlike the standard CE loss, MNCE expands the set

of positive targets to include both the ground-truth token and its nearest neighbors. By recognizing these neighbors as acceptable outcomes, MNCE allows the language model to explore auditorily valid alternatives, leading to improved intelligibility and speaker similarity without compromising the overall audio quality.

Our contributions can be summarized as follows:

- We analyze the Neighboring Token Phenomenon in codec-based TTS systems, and demonstrate the prevalence of this phenomenon across both codec and codec language models. Furthermore, we reveal the mismatch with the standard CE training objective.
- Based on this analysis, we propose the Multi-Neighbor Cross-Entropy (MNCE) loss function, a simple yet effective generalization of the standard CE loss. MNCE relaxes the one-hot target by treating neighboring tokens as valid targets.
- Experimental results validate the effectiveness of MNCE in various codec-based TTS systems. MNCE improves CosyVoice2 by 1.36% absolute reduction and 35.32% relative reduction in WER, and 0.9% absolute improvement in speaker similarity. Furthermore, expanding the training data to 20,000 hours validates the excellent scalability of our approach.

2 Analysis of Neighboring Token Phenomenon in Codec-based TTS

In this section, we conduct **neighbor substitution** experiments in codec-based TTS systems to analyze the neighboring token phenomenon. First, we introduce the overall analysis method, including neighbor identification in the codec codebook and a sampling strategy for substitution with neighboring token. Then, we apply this method to the speech reconstruction task and speech generation task, to investigate the phenomenon for both codec and neural codec language model.

2.1 Design of Analysis Method

The codec-based TTS system typically involves three stages: (1) A neural audio codec quantizes a waveform X into a sequence of discrete speech tokens $S = (s_1, s_2, \dots, s_L)$. (2) A neural codec language model predicts a speech token sequence $\hat{S}$ from an input text T . (3) The codec’s decoder synthesizes an audio waveform $\tilde{X}$ from the predicted sequence $\hat{S}$.

To quantify the impact of the neighboring token phenomenon, our analysis method centers on a controlled token substitution process, which is designed to replace tokens within a ground-truth sequence S or a predicted sequence $\hat{S}$ with their neighboring tokens.

Neighbor Identification. The vocabulary $\mathcal{V}$ of a neural audio codec consists of V tokens, each corresponding to a D -dimensional codebook embedding. Let $C \in \mathbb{R}^{V \times D}$ denote the codebook, and let c_i denote the embedding of token s_i . We define the K nearest neighbors of s_i as

$$\mathcal{N}_K(s_i) = \text{TopK}_{s_j \in \mathcal{V} \setminus \{s_i\}} \left(-\|c_i - c_j\|_2 \right). \quad (1)$$

Neighbor Substitution. For the i -th speech token in the ground-truth sequence S or the predicted sequence $\hat{S}$, we keep the original token s_i with probability $1 - r$ and with probability r , we replace s_i by sampling a substitute token s'_i from its neighbor set $\mathcal{N}(s_i)$, according to a distance-based multinomial distribution:

$$p(s'_i = s_j \mid s_i) = \frac{\exp(-d(c_i, c_j))}{\sum_{s_k \in \mathcal{N}(s_i)} \exp(-d(c_i, c_k))}, \quad (2)$$

where $d(c_i, c_j)$ is the Euclidean distance between the token embeddings.

To analyze the impact of the neighboring token phenomenon on different stages of the codec-based TTS system, we apply the neighbor substitution method to two distinct tasks: speech reconstruction for analyzing the codec, and speech generation for analyzing the neural codec language model.

Speech Reconstruction. We first encode a ground-truth audio X into its corresponding token sequence S , then apply the neighbor substitution to each token in S to obtain a modified sequence S' . By comparing the auditory quality of the audio reconstructed from S' with that from the original S , we can measure the perceptual impact of the neighboring tokens at the codec level.

Speech Generation. The neural codec language model is used to predict a token sequence $\hat{S}$ from an input text T , then the same substitution process is applied to $\hat{S}$ to generate a modified sequence $\hat{S}'$. By comparing the final synthesized audio from $\hat{S}'$ with that from the original $\hat{S}$, we assess how these neighboring tokens affect the quality of the speech generated by the language model.

2.2 Analysis

For the speech reconstruction task, we investigate two popular neural audio codecs, X-codec2 (Ye et al., 2025), and S^3 tokenizer used in CosyVoice2 (Du et al., 2024). We choose these two codecs because (1) both deliver strong speech reconstruction performance and are widely adopted in multiple codec-based TTS systems (Tian et al., 2025; Wang et al., 2025); (2) they differ in codec design: X-codec2 follows the encoder-VQ-decoder architecture, whereas S^3 tokenizer uses an ASR model as the encoder and relies on an additional flow-matching decoder for waveform synthesis. These design differences allow us to verify that the neighboring token phenomenon is consistent across different codec implementations.

For the speech generation task, based on these two codecs, we examine two representative codec-based TTS systems: (1) Llasa (Ye et al., 2025), based on X-codec2 and Llama3.2 (Dubey et al., 2024); and (2) CosyVoice2 (Du et al., 2024), based on S^3 tokenizer and Qwen2 (Team et al., 2024).

We evaluate speech reconstruction and speech generation performance on LibriSpeech test-clean (Panayotov et al., 2015). We set $K = 10$, and $r = 0.25$. As detailed in Section 4.1, we use Word Error Rate (WER) for assessing content consistency and Speaker Similarity (SIM) for evaluating speaker identity. As shown in Section A.1, we study the sensitivity to the neighbor replacement probability r and find that increasing r from 0.15 to 0.25 leads to slightly larger but still limited degradation in both WER and SIM, indicating that the neighboring token phenomenon remains consistent across a range of sampling probabilities.

Speech Reconstruction. Figure 1 indicates that codec reconstruction is only mildly affected by neighbor substitution under a token substitution probability of $r = 0.25$. For both X-codec2 and S^3 tokenizer, WER gradually increases with minor fluctuations as K grows, since more distant neighbors can replace the original tokens and slightly degrade audio quality. SIM follows a similar trend: a slight decrease at small K , a modest recovery, and then a slow drift towards a stable and slightly worse SIM. Overall, enlarging the neighbor set leads to progressive but small degradations in WER and SIM, suggesting that neighboring tokens are perceptually valid substitutes for reconstruction.

Speech Generation. As shown in Figure 2, speech generation exhibits similarly small changes

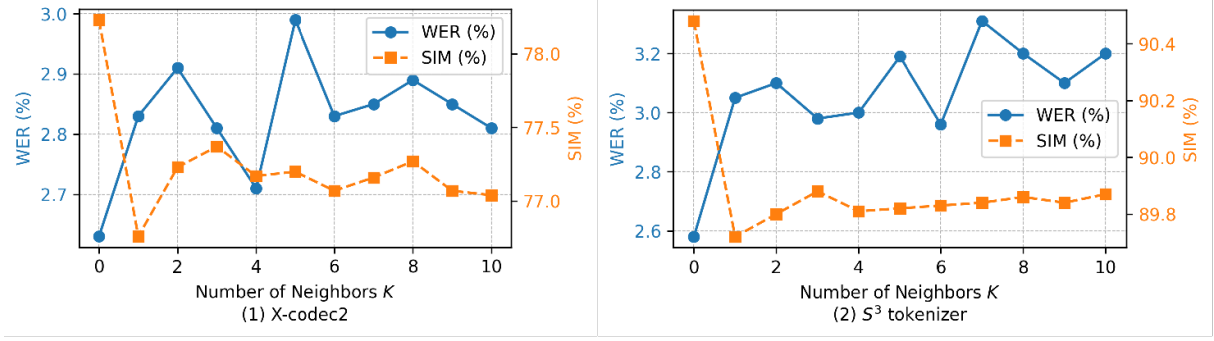


Figure 1: Impact of Neighbor Substitution with Probability $r = 0.25$ on the Speech Reconstruction Task. $K = 0$ indicates that the original token is kept unchanged. As K increases, the original token can be replaced by more distant neighbors, which moderately increases WER while keeping SIM changes within a narrow range.

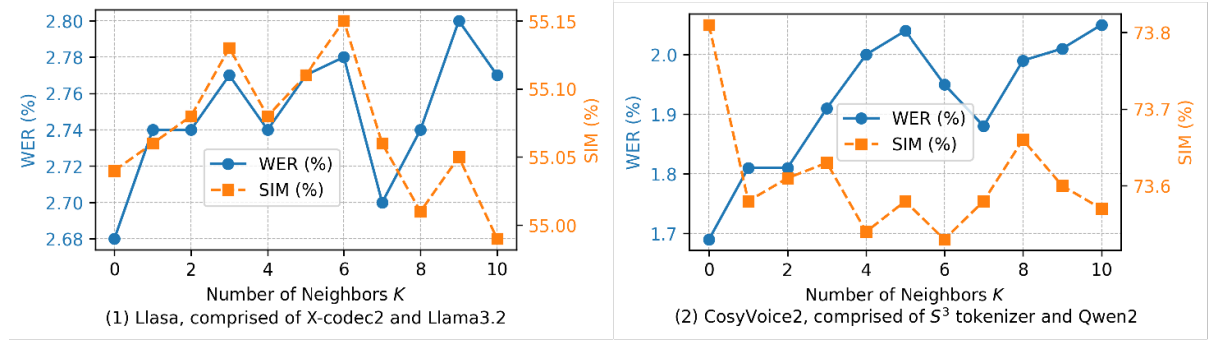


Figure 2: Impact of Neighbor Substitution with Probability $r = 0.25$ on the Speech Generation Task. Across codec-based TTS systems, substituting tokens with their neighbors leads to small changes in WER and SIM, suggesting that neighboring tokens are perceptually plausible alternatives, validating the neighboring token phenomenon.

with increasing K under the substitution probability $r = 0.25$. For both Llasa and CosyVoice2, WER and SIM vary only slightly up to $K = 10$, supporting the conclusion that neighboring tokens are acoustically plausible alternatives for language-model-predicted sequences. Interestingly, the degradation in speech generation is smaller than that in speech reconstruction. For example, with S³ tokenizer, increasing K from 0 to 10 raises reconstruction WER from 2.58 to 3.20 (absolute +0.62) and decreases SIM from 90.48 to 89.87 (absolute −0.61). In contrast, for speech generation, WER increases from 1.69 to 2.05 (absolute +0.36) and SIM decreases from 73.81 to 73.57 (absolute −0.24). This may be due to higher tolerance in generation, since swapping a token to an acoustically similar neighbor introduces only a small change and the inference stage already uses stochastic sampling that is robust to such variations.

These results confirm: **(1) neighboring token phenomenon exists not only in the codec but also in the neural codec language model trained upon it, spanning the entire codec-based TTS system;** **(2) tokens can be substituted by near neighbors**

with negligible perceptual impact. These observations motivate our new training objective that accounts for neighboring tokens, rather than treating all non-target tokens as equally incorrect.

3 Multi-Neighbor Token Prediction

In this section, we first revisit the use of the cross-entropy loss in codec-based TTS systems, including how the ground-truth speech tokens are used as training targets. Based on the analysis in Section 2.2, we propose **multi-neighbor token prediction**, a novel training paradigm that treats neighboring tokens as partially correct targets, and describe how it can be incorporated into codec-based TTS.

3.1 Preliminaries

In codec-based TTS, a neural codec language model parameterized by θ is trained on pairs (T, S^{gt}) , where T is the text transcription and S^{gt} is the ground-truth speech token sequence. Following the standard zero-shot TTS setup, we apply prompt conditioning during training by using the first N^{ref} tokens of S^{gt} as a speech prompt $\tilde{S}$. The speech prompt conveys speaker timbre and other

paralinguistic information. Conditioned on T and $\tilde{S}$, the model predicts the remaining tokens of S^{gt} after the prompt. We denote this continuation sequence by $S = (s_1, \dots, s_L)$ for simplicity, where s_1 corresponds to the $(N^{\text{ref}} + 1)$ -th token of S^{gt} . The model learns the autoregressive conditional distribution

$$P_\theta(S | T, \tilde{S}) = \prod_{i=1}^L P_\theta(s_i | T, \tilde{S}, s_{<i}), \quad (3)$$

where $s_{<i} = (s_1, \dots, s_{i-1})$ and L is the length of S .

The language model is trained by maximizing $P_\theta(S | T, \tilde{S})$, which is equivalent to minimizing the Cross-Entropy (CE) loss:

$$\mathcal{L}_{\text{CE}}(\theta) = - \sum_{i=1}^L \log P_\theta(s_i | T, \tilde{S}, s_{<i}). \quad (4)$$

3.2 Multi-Neighbor Token Prediction

Given the speech-token vocabulary $\mathcal{V}$, at step i , the model outputs logits $\{z_{i,v}\}_{v \in \mathcal{V}}$ and computes

$$P_\theta(v | T, \tilde{S}, s_{<i}) = \frac{\exp(z_{i,v})}{\sum_{u \in \mathcal{V}} \exp(z_{i,u})}. \quad (5)$$

From a pairwise comparison perspective, similar to Circle Loss (Sun et al., 2020), the CE loss in (4) can be written as

$$\mathcal{L}_{\text{CE}}(\theta) = - \sum_{i=1}^L \log \frac{\exp(z_{i,s_i})}{\sum_{u \in \mathcal{V}} \exp(z_{i,u})} \quad (6)$$

$$= \sum_{i=1}^L \log \sum_{u \in \mathcal{V}} \exp(z_{i,u} - z_{i,s_i}) \quad (7)$$

$$= \sum_{i=1}^L \log \left(1 + \sum_{u \in \mathcal{V} \setminus \{s_i\}} \exp(z_{i,u} - z_{i,s_i}) \right). \quad (8)$$

Eq. (8) shows that CE loss encourages the positive logit z_{i,s_i} to exceed each negative logit $z_{i,u}$. In other words, CE loss treats the ground-truth token s_i as the only positive sample, while every other token $u \in \mathcal{V} \setminus \{s_i\}$ is a negative sample.

Section 2.2 shows that, for codec tokens, nearest-neighbor tokens often correspond to perceptually similar audio and can serve as reasonable substitutes. Therefore, using a strict one-hot target in CE loss may be unnecessarily harsh for codec tokens: besides the ground-truth token s_i , its neighbor tokens can be viewed as partially correct targets.

For each position i , $\mathcal{N}(s_i)$ is the set of K nearest neighbor tokens of s_i computed by Eq. (1). We define the target set $\Omega_{\text{tgt}}^{(i)} = \{s_i\}$, the positive set $\Omega_{\text{pos}}^{(i)} = \{s_i\} \cup \mathcal{N}(s_i)$, and the negative set $\Omega_{\text{neg}}^{(i)} = \mathcal{V} \setminus \Omega_{\text{pos}}^{(i)}$. Ideally, the model should assign the highest probability to the ground-truth token, high probabilities to its neighbors, and low probabilities to all other tokens. Accordingly, our objectives are: (1) tokens in $\Omega_{\text{tgt}}^{(i)}$ receive the highest logit score; and (2) tokens in $\Omega_{\text{pos}}^{(i)}$ score higher than those in $\Omega_{\text{neg}}^{(i)}$. This is also beneficial at inference time: even if the model does not predict the exact token s_i , preferring $\Omega_{\text{pos}}^{(i)}$ over $\Omega_{\text{neg}}^{(i)}$ increases the chance of generating a neighbor token rather than an unrelated one, reducing audible degradation. The standard CE loss already enforces the first objective, while we focus on the second objective.

To achieve the second objective, we propose **Multi-Neighbor Cross-Entropy (MNCE)** loss, which extends the CE loss by treating neighbor tokens as positives. Importantly, in autoregressive generation, early tokens have a significant impact on the remainder of the sequence. Errors at the beginning can propagate and be amplified in subsequent predictions. In addition, the initial segment of the speech token sequence is critical for establishing speaker characteristics and prosody, making it more sensitive to token perturbations. Hence, we compute the MNCE loss only from the p_{start} -th position onward, as follows:

$$\mathcal{L}_{\text{MNCE}}(\theta) = \sum_{i=p_{\text{start}}}^L \log \left(1 + \sum_{u \in \Omega_{\text{neg}}^{(i)}} \sum_{v \in \Omega_{\text{pos}}^{(i)}} \exp(z_{i,u} - z_{i,v}) \right). \quad (9)$$

Notably, when $K = 0$ and $p_{\text{start}} = 1$, $\mathcal{N}(s_i) = \emptyset$ and $\Omega_{\text{pos}}^{(i)} = \{s_i\}$, then $\mathcal{L}_{\text{MNCE}}$ reduces to $\mathcal{L}_{\text{CE}}$.

MNCE loss encourages all positives including neighbors to outrank negatives, but it does not explicitly enforce the ground-truth token to dominate its neighbors. Thus, we use it as a regularizer on top of the CE loss to satisfy both objectives. The overall training objective is:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{CE}}(\theta) + \lambda \mathcal{L}_{\text{MNCE}}(\theta), \quad (10)$$

λ controls the multi-neighbor regularization.

Discussion: Why not weight positive logits based on distance. In practice, to avoid degrading reconstruction or generation quality, we use a

Table 1: Hyper-parameters for MNCE. Frame rate denotes the number of frames produced by the codec per second. p_{start} is the token index in the speech token sequence S from which MNCE is applied, and K is the number of neighbors in the neighbor set $\mathcal{N}$.

Neural Audio Codec	Frame Rate	p_{start}	K
WavTokenizer	75	150	6
S^3 tokenizer	25	50	3
X-codec2	50	100	3

small neighbor set so that selected neighbors remain acoustically similar. As a result, the distance distribution within the neighbor set is negligible. For example, with X-codec2 at $K = 8$, the 1st and 8th neighbors of a given s_i have average Euclidean distances 2.07 and 2.11, respectively. Therefore, we treat all neighbor tokens in $\Omega_{\text{pos}}^{(i)}$ equally and do not weight positives by their Euclidean distance to s_i , for the simplicity of the algorithmic design.

We precompute neighbor sets offline, so MNCE loss adds negligible training overhead. The pseudocode is provided in Algorithm 1 in Appendix.

4 Experiments

4.1 Experimental Setup

Datasets. We train neural codec language models on LibriTTS (Zen et al., 2019) (960 hours). For the speech generation task, we conduct in-domain evaluation on LibriSpeech-PC *test-clean* following the configuration of F5TTS (Chen et al., 2025b). We additionally evaluate out-of-domain generalization on SEED *test-en* (Anastassiou et al., 2024). For data scaling experiments, we train on Emilia (He et al., 2024) using 20,000 hours of speech.

Baseline Models. We evaluate our method on zero-shot text-to-speech, which is a representative speech generation task. We compare against open-source TTS systems, including MaskGCT (Wang et al., 2024), Llasa (Ye et al., 2025), CosyVoice2 (Du et al., 2024), CosyVoice3 (Du et al., 2025), F5-TTS (Chen et al., 2025b) and VibeVoice (Peng et al., 2025). In addition, for each codec-based TTS setup considered in our experiments, we train two codec language models, one using the standard CE objective only and the other using the proposed MNCE objective. We also include label smoothing (Müller et al., 2019) in our comparisons as the baseline.

Training Details. We adopt Qwen-2.5-0.5B as the backbone of the neural codec language model and consider three types of neural audio

Table 2: **Objective speech generation** results on SEED *test-en*, compared with SOTA and open-source TTS systems. The best result is in **bold** and the second-best is underlined. ♣ denotes the model reproduced and trained with our *MNCE loss*, ◇ denotes results inferred from the official checkpoints, and † denotes results cited from JoyVoice (Yu et al., 2025) or the original papers.

TTS model	Data Scale	WER (%)↓	SIM(%)↑
MaskGCT†	100Kh	2.62	73.0
LLasa◇	250Kh	3.34	60.6
CosyVoice2◇	170Kh	2.35	73.9
CosyVoice3-0.5B†	-	2.02	79.0
F5-TTS†	95Kh	<u>1.83</u>	<u>74.2</u>
VibeVoice-7B†	-	4.42	59.4
CosyVoice2♣	20Kh	1.69	72.8

codecs: an acoustic codec WavTokenizer (Ji et al., 2024), a semantic codec S^3 tokenizer (Du et al., 2024), and a mixed-semantic-acoustic codec X-codec2 (Ye et al., 2025). We use ttsfrd² to transcribe the input text as phonemes, and use a speech prompt of approximately 3 seconds, corresponding to N^{ref} speech tokens depending on the codec frame rate. We train the language model with the Adam optimizer (Diederik, 2014) for 100,000 iterations, using a learning rate of 9.5×10^{-4} and $\beta = (0.9, 0.99)$.

When applying the MNCE loss in Eq. (10), we set $\lambda = 0.1$ and compute MNCE only from the token position corresponding to 2 seconds of target speech. Since different codecs have different frame rates, this corresponds to different values of p_{start} , as summarized in Table 1. Other details are also listed in Table 1. The inference hyper-parameters of the codec-based TTS systems are provided in Table 6 in the appendix.

Evaluation Metrics. (1) Objective evaluation. We use Whisper (Radford et al., 2023) to transcribe the generated speech and compute the word error rate (WER). For speaker similarity (SIM), we extract speaker embeddings from the generated speech and the prompt audio using the ERes2Net speaker verification model (Chen et al., 2024), and compute the cosine similarity between the ℓ_2 -normalized embeddings. To assess naturalness, we report the score of UTMOS (Saeki et al., 2022), an automatic predictor of MOS for generated speech. **(2) Subjective evaluation.** We randomly select 50 utterances from the LibriSpeech-PC *test-clean* and conduct mean opinion score (MOS) and similarity MOS (SMOS) evaluations (Chu and Peng, 2006).

²<https://huggingface.co/FunAudioLLM/CosyVoice-ttsfrd>

Table 3: **Objective speech generation** results on LibriSpeech-PC *test-clean* and SEED *test-en*. The best result is in **bold** and the second-best is underlined. ♣ denotes the model trained with *MNCE loss*, LS denotes the model trained with *Label Smoothing*, ◇ denotes results inferred from the official checkpoints, and † denotes results cited from the original papers.

Neural Audio Codec	Neural Codec Language Model	Data Scale	LibriSpeech-PC <i>test-clean</i>			SEED <i>test-en</i>		
			WER (%)↓	SIM(%)↑	UTMOS↑	WER (%)↓	SIM(%)↑	UTMOS↑
Ground Truth	/	/	2.13	69.7	4.04	2.14	74.2	3.51
X-codec2	LLasa [◇]	250,000h	2.97	55.4	4.27	3.34	60.6	3.95
S^3 tokenizer	CosyVoice2 [◇]	170,000h	<u>1.69</u>	73.7	4.34	2.35	73.9	4.16
WavTokenizer	Qwen2.5	960h	6.61	45.7	4.04	10.21	44.6	3.76
	Qwen2.5♣	960h	4.81	46.9	4.07	7.39	45.1	3.80
X-codec2	Qwen2.5	960h	4.13	49.5	4.25	4.82	49.4	3.97
	Qwen2.5♣	960h	2.89	50.1	4.25	3.51	50.2	3.97
	Qwen2.5 ^{LS}	960h	3.82	49.6	4.23	3.98	49.2	3.95
S^3 tokenizer	Qwen2.5	960h	3.85	71.9	4.30	4.57	72.8	<u>4.13</u>
	Qwen2.5♣	960h	2.49	72.8	4.32	3.45	<u>73.3</u>	<u>4.13</u>
	Qwen2.5 ^{LS}	960h	3.38	72.7	4.31	4.08	72.8	4.12
S^3 tokenizer	Qwen2.5	20,000h	2.38	72.1	4.32	<u>2.27</u>	71.7	4.11
	Qwen2.5♣	20,000h	1.67	<u>72.1</u>	<u>4.33</u>	1.69	72.8	4.12

Table 4: **Subjective speech generation evaluation** results on LibriSpeech-PC *test-clean*. The best result is in **bold** and the second-best is underlined. ♣ denotes the model trained with *MNCE loss*, ◇ denotes results inferred from the official checkpoints. The subscript (e.g., 960h) denotes the training data scale.

Neural Audio Codec	Neural Codec Language Model	Subjective	
		MOS↑	SMOS↑
X-codec2	LLasa [◇] _{250Kh}	4.16	4.07
S^3 tokenizer	CosyVoice2 [◇] _{170Kh}	4.36	4.34
X-codec2	Qwen2.5 _{960h}	3.76	3.82
	Qwen2.5♣ _{960h}	3.84	3.89
S^3 tokenizer	Qwen2.5 _{960h}	4.13	4.22
	Qwen2.5♣ _{960h}	4.21	4.25
S^3 tokenizer	Qwen2.5 _{20Kh}	4.23	4.26
	Qwen2.5♣ _{20Kh}	<u>4.25</u>	<u>4.29</u>

MOS measures the naturalness of the generated speech. SMOS measures timbre similarity between the generated speech and the prompt speech.

4.2 Speech Generation Results

Comparison with SoTA. Table 2 compares MNCE with SOTA and open-source TTS systems on SEED *test-en*. The reproduced CosyVoice2 system is built on the S^3 tokenizer and a Qwen2.5-based neural codec language model, and is trained with the proposed MNCE loss function. With 20K hours of training data, this MNCE-trained system achieves the best WER of 1.69, outperforming the previous best result from F5-TTS (WER 1.83) and improving over the official CosyVoice2 checkpoint (WER 2.35). These results indicate that MNCE yields con-

sistent gains when applied to a strong codec-based TTS system.

Objective evaluation. Table 3 reports speech generation results on LibriSpeech-PC *test-clean* and SEED *test-en*. In addition to open-source base-lines evaluated from their released checkpoints, we train our own codec language model based on Qwen2.5 with three neural audio codecs (WavTokenizer, X-codec2, and S^3 tokenizer), and compare two training objectives: the standard CE loss and CE augmented with MNCE. We find that:

(1) Across different codecs, MNCE consistently improves WER, SIM, and UTMOS, indicating higher-quality generated speech. These gains hold for acoustic codecs (e.g., WavTokenizer), semantic-acoustic mixed codecs (e.g., X-codec2), and semantic codecs (e.g., S^3 tokenizer). MNCE substantially reduces WER, indicating better intelligibility and content accuracy, and it also increases SIM and slightly improves UTMOS, suggesting improved speaker consistency and speech naturalness. For instance, with 960 hours of training data, MNCE improves S^3 tokenizer + Qwen2.5 by a 1.36% absolute and a 35.32% relative reduction in WER and a 0.9% absolute increase in SIM on LibriSpeech-PC *test-clean*. These gains are also consistent on the out-of-domain SEED *test-en* set, indicating improved generalization. For example, with 960 hours, MNCE reduces WER for S^3 tokenizer + Qwen2.5 from 4.57 to 3.45.

(2) MNCE remains effective under data scaling. For S^3 tokenizer with Qwen2.5, scaling the

training data from 960 hours to 20,000 hours still preserves clear gains from MNCE. In particular, on LibriSpeech-PC *test-clean*, MNCE reduces WER from 2.38 to 1.67, which corresponds to a 0.71% absolute and 29.83% relative reduction.

(3) The model trained with MNCE loss achieves competitive performance with substantially less training data. With only 20,000 hours of training data, our S^3 tokenizer + Qwen2.5 model trained with MNCE achieves a WER of 1.67 on LibriSpeech-PC *test-clean*, which is comparable to, and even slightly better than, CosyVoice2 (WER 1.69) trained on 170,000 hours. This indicates that MNCE can narrow the performance gap to large-scale TTS systems using much less training data.

(4) MNCE also outperforms label smoothing, suggesting that uniformly distributing probability mass over the whole vocabulary is less effective than exploiting the neighborhood structure of the codec token space.

Subjective evaluation. As shown in Table 4, MNCE generally improves MOS and SMOS across different codecs, indicating better perceived naturalness and speaker similarity. These subjective gains are consistent with the objective improvements in WER and SIM.

4.3 Ablation Study

Table 5 in the appendix studies the effect of the MNCE start index p_{start} and the neighbor set size K . We make the following findings.

(1) **Impact of p_{start} .** Fixing $K = 3$, applying MNCE from a moderate position yields the best results (WER 2.89 at $p_{\text{start}} = 100$), while starting too early ($p_{\text{start}} = 0$) or too late ($p_{\text{start}} = 150$) brings smaller gains. We attribute this to the sensitivity of autoregressive generation to early tokens: when p_{start} is too small, relaxing the one-hot targets at the beginning may introduce errors that propagate to subsequent predictions; when p_{start} is too large, the regularization signal becomes too weak. Overall, a moderate p_{start} provides a better trade-off.

(2) **Impact of K .** Fixing $p_{\text{start}} = 100$, a moderate neighbor set size performs best (WER 2.89 at $K = 3$). Increasing K does not monotonically improve performance and can even hurt WER (e.g., WER 3.86 at $K = 5$). We also observe that the MNCE loss tends to increase with larger K , suggesting that the model struggles to simultaneously account for too many neighbors. Moreover, Section 2.2 shows that substituting a token with a small number of nearest neighbors already causes lim-

ited quality degradation, implying that selecting an appropriate, rather than the largest, neighbor set is sufficient in practice.

5 Related Work

Neural Audio Codecs. Discrete speech tokens are usually extracted using self-supervised learning (SSL) speech models or neural audio codecs, which can be grouped into three categories. **Semantic codecs** focus on linguistic content, such as S^3 tokenizer (An et al., 2024; Du et al., 2024), which incorporates an ASR auxiliary task during training to encourage the tokenizer to produce semantic tokens. **Acoustic codecs** aim for high-fidelity reconstruction and preserve rich acoustic details, including EnCodec (Défossez et al., 2022; Liu et al., 2025b), HiFiCodec (Yang et al., 2023), Wav-Tokenizer (Ji et al., 2024), and UniCodec (Jiang et al., 2025). **Semantic-acoustic-mixed codecs** seek high-quality audio reconstruction while retaining rich semantic information; they typically leverage a pretrained SSL speech encoder (Baevski et al., 2020; Hsu et al., 2021) to extract semantic representations, as in SpeechTokenizer (Zhang et al., 2023b) and X-codec2 (Ye et al., 2025).

Neural Codec Language Models. Neural codec language models (Chen et al., 2025a; Yang et al., 2024; Liu et al., 2025a) predict speech tokens, which are then decoded into waveforms by codecs. CosyVoice2 (Du et al., 2024) uses S^3 tokenizer to extract semantic tokens and synthesizes speech with a flow-matching decoder. MaskGCT (Wang et al., 2024) predicts discrete semantic and acoustic tokens using a mask-and-predict paradigm. Llasa (Ye et al., 2025) adopts X-codec2, which encodes semantics and acoustics simultaneously. These neural codec language models are typically trained with the standard cross-entropy loss, which treats the target token as the only correct label, ignoring the neighborhood structure of codec token spaces. While label smoothing (Müller et al., 2019) also relaxes one-hot supervision, it distributes probability mass uniformly over the whole vocabulary and does not leverage neighbor tokens, and thus is less targeted for codec tokens. To address this mismatch, we propose MNCE loss and improve the performance of codec-based TTS systems.

6 Conclusions

In this paper, we identify a mismatch in codec-based TTS between the one-hot CE objective and

the geometry of codec token spaces. Neighbor-substitution analyses in both speech reconstruction and generation verify the **Neighboring Token Phenomenon**, where nearest-neighbor tokens are often perceptually plausible substitutes. To address this, we propose **Multi-Neighbor Cross-Entropy (MNCE)** loss, which extends CE by treating the ground-truth token and its nearest neighbors as a positive set. Experiments on diverse codec-based TTS systems show that MNCE consistently improves speech generation quality and robustness, with gains persisting under data scaling.

Limitations

This paper explores the neighboring-token phenomenon in speech reconstruction and speech generation, and proposes the MNCE loss. Nevertheless, several limitations remain.

(1) The neighbor set is typically precomputed globally, whereas token substitutability can be context-dependent and may vary with phonetic position, speaker characteristics, and prosody. As a result, a fixed neighbor set may include suboptimal positives in certain contexts.

(2) When K is set too large, MNCE may over-relax one-hot supervision and distribute probability mass across multiple neighbors, potentially leading to slightly duller articulation or blurred fine-grained acoustic details.

(3) MNCE introduces additional hyperparameters (e.g., K , λ , and p_{start}) and requires neighbor computation, which can increase tuning effort and training overhead.

Ethics Statement

This work should be deployed in a responsible manner, with due respect for speaker consent, data privacy, and relevant synthetic-media policies.

References

- Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, et al. 2024. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*.
- Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.
- Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Gökner, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*.
- Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2025a. [Neural codec language models are zero-shot text to speech synthesizers](#). *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718.
- Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, et al. 2024. [3d-speaker-toolkit: An open source toolkit for multi-modal speaker verification and diarization](#).
- Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. 2025b. [F5-TTS: A fairytale that fakes fluent and faithful speech with flow matching](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271, Vienna, Austria. Association for Computational Linguistics.
- Min Chu and Hu Peng. 2006. Objective measure for estimating mean opinion score of synthesized speech. US Patent 7,024,362.
- Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.
- Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.
- P Kingma Diederik. 2014. Adam: A method for stochastic optimization. (*No Title*).
- Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. 2025. [Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training](#). *arXiv preprint arXiv:2505.17589*.
- Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. 2024. [Cosyvoice 2: Scalable streaming speech synthesis with large language models](#). *arXiv preprint arXiv:2412.10117*.

- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- 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. In *Proc. of SLT*.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.
- Teng Hu, Jiangning Zhang, Ran Yi, Jieyu Weng, Yabiao Wang, Xianfang Zeng, Zhucun Xue, and Lizhuang Ma. 2025. Improving autoregressive visual generation with cluster-oriented token prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9351–9360.
- Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, et al. 2024. Wavtok-enizier: an efficient acoustic discrete codec tokenizer for audio language modeling. *arXiv preprint arXiv:2408.16532*.
- Yidi Jiang, Qian Chen, Shengpeng Ji, Yu Xi, Wen Wang, Chong Zhang, Xianghu Yue, ShiLiang Zhang, and Haizhou Li. 2025. Unicoec: Unified audio codec with single domain-adaptive codebook. *arXiv preprint arXiv:2502.20067*.
- Zeqian Ju, Dongchao Yang, Jianwei Yu, Kai Shen, Yichong Leng, Zhengtao Wang, Xu Tan, Xinyu Zhou, Tao Qin, and Xiangyang Li. 2025. Mooncast: High-quality zero-shot podcast generation. *arXiv preprint arXiv:2503.14345*.
- Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, et al. 2025. Baichuan-omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*.
- Wenrui Liu, Qian Chen, Wen Wang, Guanrou Yang, Weiqin Li, Minghui Fang, Jialong Zuo, Xiaoda Yang, Tao Jin, Jin Xu, et al. 2025a. Speech token prediction via compressed-to-fine language modeling for speech generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10632–10641.
- Wenrui Liu, Zhifang Guo, Jin Xu, Yuanjun Lv, Yunfei Chu, Zemin Liu, and Junyang Lin. 2025b. Analyzing and mitigating inconsistency in discrete speech tokens for neural codec language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31035–31046.
- Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.
- 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.
- Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, et al. 2025. Vibevoice technical report. *arXiv preprint arXiv:2508.19205*.
- 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.
- Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*.
- Tal Shuster and Eliya Nachmani. 2025. Q2d2: A geometry-aware audio codec leveraging two-dimensional quantization. *arXiv preprint arXiv:2512.01537*.
- Yifan Sun, Changmao Cheng, Yuhang Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6397–6406. IEEE.
- Qwen Team et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).
- Wenjie Tian, Xinfa Zhu, Hanke Xie, Zhen Ye, Wei Xue, and Lei Xie. 2025. Llasa+: Free lunch for accelerated and streaming llama-based speech synthesis. *arXiv preprint arXiv:2508.06262*.
- Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.
- Tianrui Wang, Haoyu Wang, Meng Ge, Cheng Gong, Chunyu Qiang, Ziyang Ma, Zikang Huang, Guanrou Yang, Xiaobao Wang, Eng Siong Chng, et al. 2025. Word-level emotional expression control in zero-shot text-to-speech synthesis. *arXiv preprint arXiv:2509.24629*.
- Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2024. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*.

- Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. 2023. 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, Haohan Guo, Xuankai Chang, Jiaotong Shi, Jiang Bian, Zhou Zhao, et al. 2024. Uniaudio: Towards universal audio generation with large language models. In *Forty-first International Conference on Machine Learning*.
- Zhen Ye, Xinfu Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. 2025. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*.
- Fan Yu, Tao Wang, You Wu, Lin Zhu, Wei Deng, Weisheng Han, Wenchao Wang, Lin Hu, Xiangyu Liang, Xiaodong He, et al. 2025. Joyvoice: Long-context conditioning for anthropomorphic multi-speaker conversational synthesis. *arXiv preprint arXiv:2512.19090*.
- 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.
- Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. [Libritts: A corpus derived from librispeech for text-to-speech](#). *Preprint*, arXiv:1904.02882.
- Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023a. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.
- Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023b. Speectokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

A Appendix

A.1 More Results of Neighbor Token Phenomenon

Figure 3, 4 and 5 demonstrate the speech generation results with neighbor replacement, and we can draw the following conclusions.

(1) The Neighboring Token Phenomenon holds in the generation setting and is relatively robust: replacing tokens with their nearest neighbors causes only limited changes in WER and SIM for both Llasa and CosyVoice2, indicating that neighbor tokens are often perceptually plausible alternatives.

(2) Increasing the number of neighbors K generally leads to larger degradation, especially in WER. As the neighborhood expands to include more distant tokens, replacements are more likely to introduce content errors and reduce speaker consistency.

(3) The system becomes more sensitive to K when the replacement probability r increases. As r grows from 0.15 to 0.25, the degradation becomes more noticeable, but it remains relatively small overall, suggesting that neighbor substitutions are still mostly safe even at higher replacement rates.

(4) The same trends are observed across different codec designs, indicating that the phenomenon is consistent across implementations. Meanwhile, the degradation patterns are not identical across systems, implying that different tokenizers may have different tolerance to neighborhood expansion.

A.2 Pseudo-code for MNCE loss

As shown in Algorithm 1, the pseudocode computes MNCE at each position by forming a positive set (the ground-truth token and its nearest neighbors), computing log-sum-exp terms for the positives and for the full vocabulary, deriving the corresponding negative term via log-sub-exp, and then applying a softplus to obtain the per-position loss. An attention mask (set to 1 only after N^{ref}) is used to average the loss over valid positions only.

Key numerical-stability techniques to prevent overflow and instability:

(1) Log-sum-exp (LSE) for $\log \sum \exp(\cdot)$, avoiding overflow from computing $\exp$ and summing in linear space.

(2) Stable log-sub-exp: $\log(\exp(a) - \exp(b)) = a + \log(1 - \exp(b - a))$, avoiding catastrophic cancellation when subtracting large numbers.

(3) Softplus for $\log(1 + \exp(x))$, avoiding overflow when x is large.

A.3 Results of Ablation Study

Table 5 studies the effect of the MNCE start index p_{start} and the neighbor set size K . We draw the following conclusions:

(1) **Impact of p_{start} .** With $K = 3$, a moderate start index performs best (WER 2.89 at $p_{\text{start}} = 100$), while starting too early ($p_{\text{start}} = 0$) or too late ($p_{\text{start}} = 150$) yields smaller gains. This is consistent with the sensitivity of autoregressive generation to early tokens: a small p_{start} may introduce errors that propagate, whereas a large p_{start} weakens the regularization signal.

(2) **Impact of K .** With $p_{\text{start}} = 100$, a moderate neighbor set performs best (WER 2.89 at $K = 3$). Increasing K does not monotonically help and can even degrade WER (e.g., 3.86 at $K = 5$); correspondingly, the MNCE loss tends to increase with larger K , suggesting optimization becomes harder when too many neighbors are treated as positives. Together with the neighbor-substitution results in Section 2.2, this indicates that an appropriate (rather than the largest) K is sufficient in practice.

A.4 Detailed hyper-parameters of Neural Codec Language Models

Table 6 demonstrates the hyper-parameters of inference for various codec-based TTS systems.

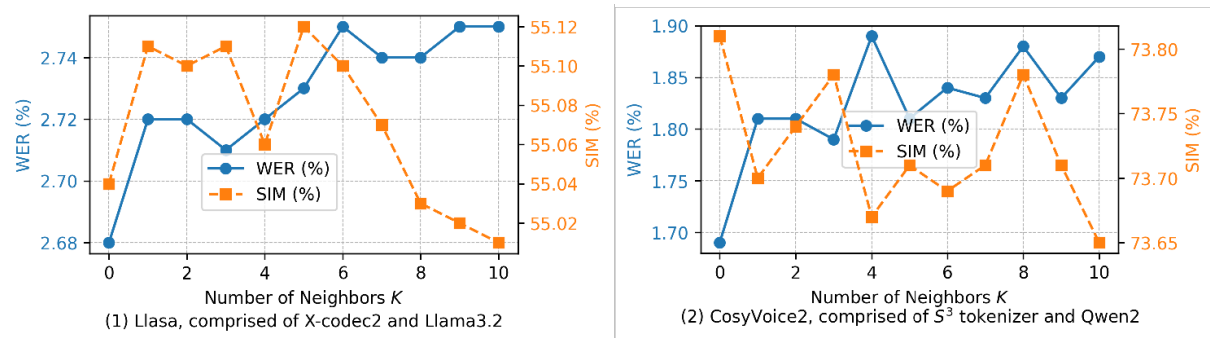


Figure 3: Impact of Neighbor Substitution with Probability $r = 0.15$ on the Speech Generation Task.

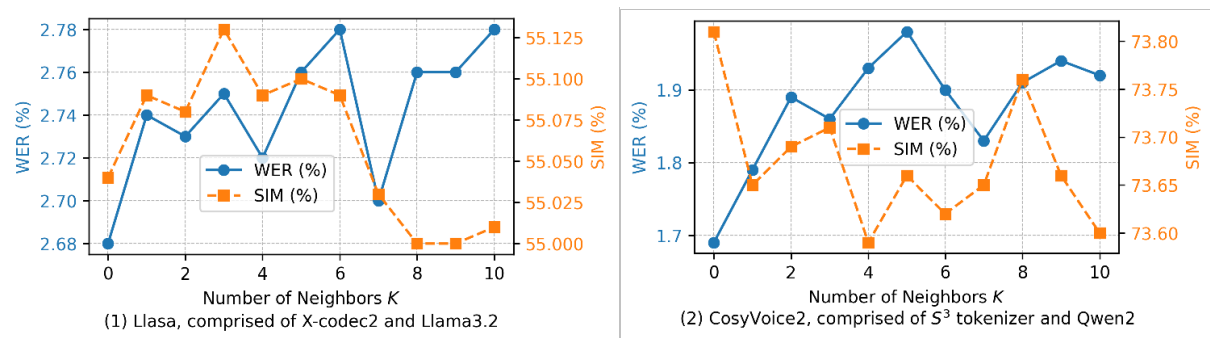


Figure 4: Impact of Neighbor Substitution with Probability $r = 0.20$ on the Speech Generation Task.

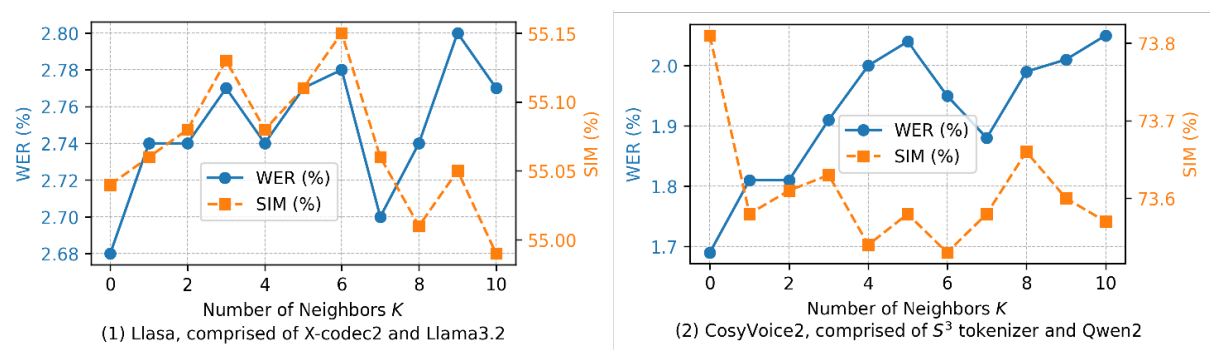


Figure 5: Impact of Neighbor Substitution with Probability $r = 0.25$ on the Speech Generation Task.

Algorithm 1: Multi-Neighbor Cross-Entropy (MNCE) Loss

Input: $\mathbf{Z} \in \mathbb{R}^{B \times S \times V}$: logits over the full vocabulary at each position;
 $\mathbf{Z}^+ \in \mathbb{R}^{B \times S \times K}$: logits of the positive set (ground-truth token and its K nearest neighbors) at each position;
 $\mathbf{M} \in \{0, 1\}^{B \times S}$ (optional): attention mask where $M_{b,s} = 1$ indicates the position is included in the loss (e.g., $s \geq N^{\text{ref}}$), otherwise ignored.
Output: $\mathcal{L}$: scalar MNCE loss averaged over valid positions.

for $b = 1$ **to** B **do**

for $s = 1$ **to** S **do**

$\ell_{b,s}^+ \leftarrow \text{LSE}(-\mathbf{Z}_{b,s,:}^+)$;
 // $\log \sum_{v \in \Omega_{\text{pos}}} \exp(-z_v)$
 $\ell_{\text{all},b,s} \leftarrow \text{LSE}(\mathbf{Z}_{b,s,:})$;
 // $\log \sum_{u \in \mathcal{V}} \exp(z_u)$
 $\ell_{\text{pos},b,s} \leftarrow \text{LSE}(\mathbf{Z}_{b,s,:}^+)$;
 // $\log \sum_{v \in \Omega_{\text{pos}}} \exp(z_v)$
 $\ell_{b,s}^- \leftarrow \text{LogSubExp}(\ell_{\text{all},b,s}, \ell_{\text{pos},b,s})$;
 // $\log \sum_{u \in \Omega_{\text{neg}}} \exp(z_u)$
 $L_{b,s} \leftarrow \text{Softplus}(\ell_{b,s}^- + \ell_{b,s}^+)$;
 // $\log(1 + \sum_{\Omega_{\text{neg}}} e^z \sum_{\Omega_{\text{pos}}} e^{-z})$

if $\mathbf{M}$ is provided **then**

$\mathcal{L} \leftarrow \frac{\sum_{b,s} M_{b,s} L_{b,s}}{\sum_{b,s} M_{b,s}}$

else

$\mathcal{L} \leftarrow \frac{1}{BS} \sum_{b,s} L_{b,s}$

return $\mathcal{L}$

Definitions:

$\text{LSE}(\mathbf{x}) = \log \sum_i \exp(x_i)$ (log-sum-exp).

$\text{Softplus}(x) = \log(1 + \exp(x))$.

$\text{LogSubExp}(a, b) =$

$a + \log(1 - \exp(b - a))$ (stable

$\log(\exp(a) - \exp(b))$, assuming $a \geq b$).

Table 5: Ablation study on X-codec2 with Qwen2.5, which is trained on LibriTTS 960h and evaluated on the LibriSpeech *test-clean* subset. p_{start} is the token index in the speech token sequence S from which MNCE is applied, and K is the number of neighbors in the neighbor set $\mathcal{N}$.

p_{start}	K	WER (%)↓	SIM(%)↑	UTMOS↑
0	0	4.13	49.5	4.25
0	3	3.76	49.9	4.23
50	3	3.13	49.2	4.25
100	3	2.89	50.1	4.25
150	3	4.05	49.4	4.24
100	1	3.94	49.8	4.21
100	2	3.37	50.0	4.23
100	3	2.89	50.1	4.25
100	4	3.16	49.8	4.25
100	5	3.86	49.7	4.24
100	6	3.23	49.9	4.24

Table 6: Inference hyper-parameters for different codec-based TTS systems. All systems use a Qwen2.5-based codec language model but differ in the neural codec and decoding configuration. The decoding settings for the X-codec2-based system follow the official implementation (<https://huggingface.co/HKUSTAudio/Llasa-1B>), and the S^3 tokenizer-based system follow the official implementation (<https://github.com/FunAudioLLM/CosyVoice/tree/main/cosyvoice>).

Codec	Top- p	Top- k	Temperature
WavTokenizer	1.0	40	0.9
X-codec2	1.0	50	0.8
S^3 tokenizer	0.8	25	1.0