# INVESTIGATING NEURAL AUDIO CODECS FOR SPEECH LANGUAGE MODEL-BASED SPEECH GENERATION

*Jiaqi Li[2],[⋆],[†] , Dongmei Wang[1],[⋆], Xiaofei Wang[1], Yao Qian[1], Long Zhou[1], Shujie Liu[1]*
*Midia Yousefi[1], Canrun Li[1] Chung-Hsien Tsai[1], Zhen Xiao[1], Yanqing Liu[1]*
*Junkun Chen[1], Sheng Zhao[1], Jinyu Li[1], Zhizheng Wu[2], Michael Zeng[1]*

[1] Microsoft, One Microsoft Way, Redmond, WA, USA
[2] The Chinese University of Hong Kong, Shenzhen, China

## ABSTRACT

Neural audio codec tokens serve as the fundamental building blocks for speech language model (SLM)-based speech generation. However, there is no systematic understanding on how the codec system affects the speech generation performance of the SLM. In this work, we examine codec tokens within SLM framework for speech generation to provide insights for effective codec design. We retrain existing high-performing neural codec models on the same data set and loss functions to compare their performance in a uniform setting. We integrate codec tokens into two SLM systems: masked-based parallel speech generation system and an auto-regressive (AR) plus non-auto-regressive (NAR) model-based system. Our findings indicate that better speech reconstruction in codec systems does not guarantee improved speech generation in SLM. A high-quality codec decoder is crucial for natural speech production in SLM, while speech intelligibility depends more on quantization mechanism.

*Index Terms*— neural audio codec, speech language model, speech generation, tokens, codec investigation

## 1. INTRODUCTION

The emergence and success of large language models, such as the GPT series of work [1, 2], have inspired research in the field of speech language model (SLM) within the speech generation community [3, 4, 5, 6]. Rather than synthesizing the speech in a sample-by-sample fashion [7] or by estimating the continuous features such as mel-spectrum, SLM directly predict the discrete speech tokens. These predicted tokens are then used by the pre-trained decoder module to reconstruct the waveform. By modeling discrete speech tokens within the language model framework, we can harness the advancements from large language models to enhance speech generation tasks. Another advantage of using discrete tokens is that it facilitates the construction of multi-modal model. These

tokens originating from various modalities (such as speech, text, video, etc.), can be seamlessly integrated and processed in a unified manner.

Discrete speech representation tokens primarily stem from two lines of work: self-supervised learning [8, 9, 10], resulting in semantic tokens, and the neural audio codec system [11, 12, 13], yielding codec tokens. Initially, semantic tokens were adopted by SLM-based speech generation [14]. While they produce intelligible speech, they fail to capture consistence acoustic traits like speaker identity. In contrast, the neural audio codec system, which is based on residual vector quantization (RVQ), generates hierarchical codec tokens that preserve rich acoustic information. These codec tokens are then employed, either alone [4] or alongside semantic tokens by SLM to generate speech [3, 5].

Neural audio codec systems were originally designed for communication purposes, whereas compression rate and reconstruction quality are the primary evaluation metrics. Extended from VQ-VAE [15] and introducing the RVQ module, SoundStream [11] stands as the pioneering neural audio codec model. It comprises three essential modules: encoder, RVQ-based quantizer, and decoder. These components are jointly trained using both reconstruction and adversarial losses. Subsequently, a series of neural audio codec models are proposed [12, 13, 16], expanding upon SoundStream. Encodec [12] incorporated LSTM layers and a small transformer language model over the quantized units to further reduce bandwidth. Vocos [16] predicted the short-time Fourier transform (STFT) coefficients instead of waveform in the decoder stage, which demonstrates improvement over Encodec. DAC [13] proposed two key improvements over Encodec. First, it addressed the codebook collapse problem by reducing the dimension of the latent vector to a small value for quantization. Second, DAC replaced the ReLU activation function with the snake activation function [17], offering benefits for reconstructing periodic signals such as speech and music. Lately, there has been a surge in research efforts focused on designing low bit-rate codec system [18, 19, 20, 21, 22, 23] that can be integrated with speech language models, aiming for

---
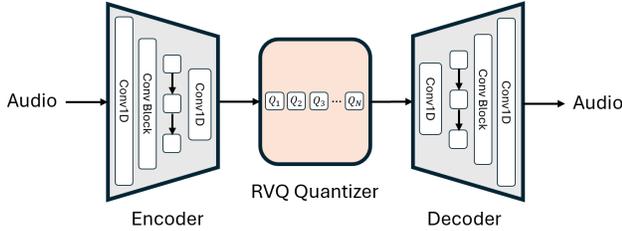
**Fig. 1**. High-level architecture of neural audio codecs

greater efficiency. A comprehensive review of existing neural codec models and the audio language models can be found in [24, 25].

While the original goal of neural codec design is to achieve a high compression rate and superior signal level reconstruction quality, it remains unclear how the codec affects the SLM-based speech generation. In this work, we investigate the effectiveness of the established codec systems within the framework of SLM for zero-shot speech generation. Our goal is to identify key components for designing codec systems tailored to SLM. Regarding codec systems, we select three high-performing models: Encodec [12], Vocos [16] and DAC [13]. To ensure a fair comparison, we retrain these models and their variants by using the same data set and loss functions. We consider two SLM-based zero-shot speech generation models. First, we adopt the masked-based parallel speech generation model, originally proposed in the SoundStorm work [5]. We built upon the original design by incorporating enhancements such as mask span and classifier-free guidance (CFG) techniques [26, 27]. Second, we explore the AR + NAR models introduced in VALL-E work [4]. We systematically assess both the speech reconstruction quality of the codec systems and the speech generation quality of the speech generation systems. The main contributions of this work include:

- We leverage large-scale speech data to retrain high-performing codec models for a unified comparison.
- We integrate these codecs into two popular SLM-based speech generation systems for investigation.
- Thorough evaluation and analysis inform effective codec design for SLM-based speech generation.

## 2. NEURAL AUDIO CODEC SYSTEMS

Neural audio codecs can compress audio into discrete representations used by speech generation models. Figure 1 presents a high-level architecture of a neural audio codec. It has an encoder that downsamples waveforms to a much lower sampling rate (e.g., 50Hz), a residual vector quantization (RVQ) module that discretizes latent features, and a decoder that reconstructs audios from discrete tokens. In the following sections, we examine the design choices of Encodec [12], Vocos [16] and DAC [13]. We categorize two aspects of a

codec that affect speech generation: i) **vector quantization (VQ) scheme** which can affect the distribution of tokens and indirectly affect speech modeling complexity; ii) **decoder scheme** which affects the generated audio quality.

### 2.1. Encodec

For quantization, Encodec [12] uses Residual Vector Quantization (RVQ) similar to SoundStream [11]. The output of the $n$-th quantizer $Q_n$ is expressed as $Q_n = \text{VQ}(Q_0 - \sum_{i=1}^{n-1} Q_i)$ where $Q_0$ is the continuous latent vector. The VQ operation at each layer is to find the codebook vector that is closest to the residual embedding in Euclidean distance. During training, the codebook vectors are updated using Exponential Moving Average (EMA)[15]. To mitigate the codebook collapse problem, Encodec also applies a "restart" technique to replace unused codebook vectors with candidates sampled from the batch. Encodec utilizes a fully-convolutional decoder SEANet[28] same as SoundStream[11], with transposed convolutions to upsample the quantized features into waveform. Two small LSTM layers are added to improve sequence modeling.

### 2.2. Vocos

Vocos [16] is a GAN-based Vocoder trained to produce STFT coefficients. It can be integrated into any neural codec framework as a decoder [16], either by training from a frozen pre-trained encoder and quantizer, or by building it from scratch in an end-to-end manner. Vocos predicts STFT coefficients (logarithmic scale spectrum amplitude and the phase values) instead of raw waveforms, and upsampling to waveform is realized through inverse Fourier Transform [16]. This system has shown to produce higher quality audios than original Encodec [16], and is used in speech generation system VALL-E 2 [29].

### 2.3. DAC

Descript-audio-codec (DAC) [13] is a recent codec system that features several VQ improvements over Encodec. DAC mitigates codebook collapse by quantizing in a very low-dimensional latent space. Its updated RVQ is: $Q_n = \text{Proj\_Out}(\text{VQ}(\text{Proj\_In}(Q_0) - \sum_{i=1}^{n-1} \text{Proj\_In}(Q_i)))$ where Proj_In is a linear projection from the original latent space (1024 dimensional) to the low-dimensional quantization latent space (8 or 32 dimensional). It also changes the VQ lookup distance from Euclidean distance to cosine similarity for stability. DAC uses an explicit MSE codebook loss function instead of the EMA update scheme to learn the projection functions. This loss can be expressed as $L_{codebook} = ||q - z||_2^2$, where $z$ and $q$ denote the looked-up vectors and the quantized vectors, respectively.

Compared with the decoder in Encodec, DAC replaces the ReLU activation with Snake activation, which is shown to benefit periodic signal reconstruction quality [17].
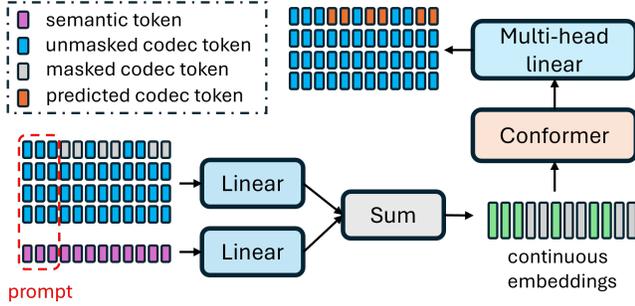
**Fig. 2**. Model architecture of masked-based parallel speech generation.

## 3. SPEECH GENERATION WITH CODEC TOKENS

For the speech generation task, we selected two types of SLM-based systems. The first one is a masked-based parallel speech generation model [5] conditioned on oracle semantic tokens. The second is an AR + NAR models-based text-to-speech system [4]. Both systems predict the codec tokens for the target speech which are then used to reconstruct the final waveform via a pre-trained codec decoder.

### 3.1. Masked-based parallel speech generation

The masked-based parallel speech generation is proposed in [5] which is inspired by maskGIT in image generation field [30]. Unlike the AR-based method, masked-based parallel speech generation produces codec tokens for the entire speech sample in a batch-style manner, iterating through multiple rounds based on confidence scores. Fig. 2 illustrates the model architecture for masked-based parallel speech generation. The backbone employs a bidirectional self-attention-based Conformer, which predicts masked codec tokens using summed embeddings of codec tokens and semantic tokens. Due to the hierarchical structure of the RVQ-based codec tokens, masked-based parallel generation occurs layer by layer, advancing to the next layer only when all tokens from the current layer have been estimated.

We adopted the span-based masking strategy [26] where a sequence of block-wise (block size is set as 5 in our case) masks is applied instead of individually masking each token in every iteration. To further enhance the quality of the generated speech, we integrated the annealing-based CFG mechanism, as proposed in the same work [26]. During training, the model is trained both conditionally and unconditionally with a certain probability. During inference, the generated signal is sampled from a linear combination of the predicted conditional and unconditional probabilities with ratio controlled by the masking rate. This mechanism gradually steers the generation process from being solely guided by semantic tokens to incorporating contextual infilling. Ultimately, the predicted codec tokens are transformed into speech waveforms using the decoder module of the codec system.
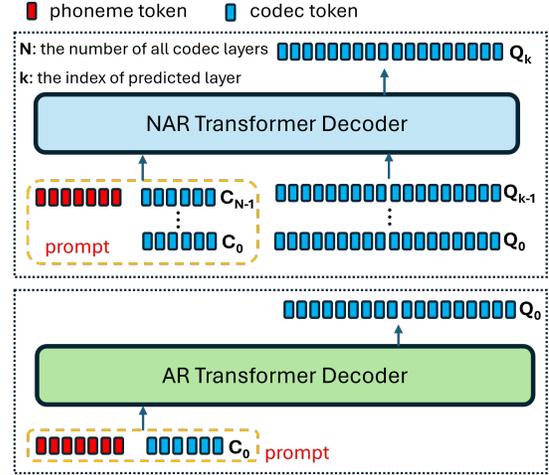


**Fig. 3**. AR + NAR speech generation.

Note that in this experiment, our focus is solely on investigating the framework of masked-based parallel generation across various codec systems. We achieve this by utilizing oracle semantic tokens as input, without the need for text-to-semantic-tokens mapping as in [5].

### 3.2. Speech generation with AR and NAR models

The alternative SLM-based speech generation system we've experimented with employs AR and NAR models, as proposed in the VALL-E work [4]. Fig. 3 illustrates the model architectures of the method. The overall system involves two stages of generation. First, the AR model takes the phoneme sequences derived from the text and the prompt codec tokens from the first quantization layer as input, predicting the first layer of the codec tokens for the target speech in an AR manner. Subsequently, the NAR model predicts the remaining codec tokens layer-by-layer, based on all the already predicted layers of codec tokens combining with the phoneme sequence and the prompt codec tokens of all quantization layers, in a parallel fashion. Both the AR and NAR models utilize the same Transformer model architecture. However, the AR model operates causally, whereas the NAR model operates in parallel.

## 4. EXPERIMENT

### 4.1. Experiment of codec reconstruction

#### 4.1.1. Model configuration

We evaluated official 24kHz pretrained models of Encodec [12], Vocos (with Encodec features)[16], and DAC [13]. We also evaluated these reproduced codec models. Since the official Encodec does not provide training codes, we reproduced a baseline codec with encoder and decoder architecture from Encodec. For the quantizer, we used an EMA quantiza-

**Table 1**. Training configuration comparison between official pre-trained models and our reproduced models.

| Model name | Quantization method | Sampling rate (kHz) | Token rate (Hz) | Bitrate (kbps) |
|---|---|---|---|---|
| Encodec-official | EMA w/ restart | 24 | 75 | 1.5-24 |
| Vocos-official | - | 24 | 75 | 1.5-12 |
| DAC-official | Projection | 24 | 75 | 0.75-24 |
| Baseline-16kHz | EMA | 16 | 50 | 4 |
| Baseline-Vocos-16kHz | - | 16 | 50 | 4 |
| DAC-16kHz | Projection | 16 | 50 | 4 |
| DAC-Vocos-16kHz | - | 16 | 50 | 4 |

**Table 2**. Evaluation results of both official codec and retrained codec models

| Codec models | PESQ | STOI | VISQOL | MCD($\downarrow$) | SIM | WER (%) |
|---|---|---|---|---|---|---|
| Encodec-official | 3.12 | 0.94 | 4.37 | 2.60 | 0.89 | 1.31 |
| Vocos-official | 3.57 | 0.95 | 4.41 | 2.50 | 0.90 | 1.31 |
| DAC-official | 3.77 | 0.95 | 4.36 | 2.34 | 0.90 | 1.27 |
| Baseline-16kHz | 3.63 | 0.95 | 4.44 | 2.32 | 0.90 | 1.29 |
| Baseline-Vocos-16kHz | 3.62 | 0.95 | 4.47 | 2.58 | 0.92 | 1.20 |
| DAC-16kHz | **3.99** | **0.97** | 4.53 | **1.95** | 0.94 | 1.12 |
| DAC-Vocos-16kHz | 3.98 | **0.97** | **4.54** | 2.00 | **0.95** | **1.06** |

tion module with no restart technique. After obtaining the baseline codec system, we trained a Vocos decoder with the encoder and quantizer frozen. We reproduced DAC using its official repository[1]. Then, we trained a Vocos decoder for the reproduced DAC with frozen encoder and quantizer.

We retrained all the codec models with 54k hours of the Librilight-Large dataset [31]. Table 1 shows a comparison between official pre-trained models and our reproduced models. We trained our models using 16kHz sampling rates to align with our training set. We also used a fixed bitrate for training reproduced models.

### 4.1.2. Training Details

For discriminators, we used a combination of a multi-scale STFT (MS-STFT) discriminator from Encodec [12], and a multi-period discriminator (MPD) from HiFi-GAN [32]. We used the same discriminator implementation for all reproduced codecs. For the loss formulation, we followed an effective configuration in [13] and used a combination of reconstruction loss, adversarial loss and commitment loss.

Each reproduced model was trained on 8 V100 GPU for 200k steps. We used a segment length of 1 second, and a batch size of 22 per GPU. We used AdamW optimizer with learning rate 1e-4, $\beta_1 = 0.8$, $\beta_2 = 0.9$, and an exponential learning rate decay with $\gamma = 0.999996$.

### 4.1.3. Evaluation Metrics

We used the short-split of Librispeech-test-clean [33] as the test set with a duration range spanning from 4 seconds to 10 seconds. To evaluate the speech quality of the codecs, we used the Perceptual Evaluation of Speech Quality (PESQ) [34], Short Term Objective Intelligibility (STOI) [35], Virtual Speech Quality Objective Listener(ViSQOL) [36], Mel Cepstral Distortion (MCD) [37] as indicators. We also measured the speaker similarity (SIM) between the original speech and reconstructed speech using the WavLM-TDNN model [38]. For Word Error Rate (WER) evaluation, we used a market-leading ASR API to get the transcripts and calculate word error rates with ground-truth transcripts.

---

[1]https://github.com/descriptinc/descript-audio-codec

### 4.1.4. Results

As shown in Table 2, for official models, we observed that the official DAC model performs the best among most metrics. We also observed that using the Vocos-official decoder for Encodec improves its codec quality. Our reproduced models share similar trends, where the reproduced DAC-16kHz model performed better than the baseline codec. Also, adding Vocos can improve some aspects of sound quality, in particular, we found this gives better speaker similarity and ViSQOL scores, and lower word error rates. We observed that our reproduced models generally outperform official models in the evaluation metrics. This superiority may stem from both our training set and test set being within the audiobook domain, ensuring better alignment between train-test distributions compared to official models, which incorporate crowd-sourced data like Common Voice [39] in training.

## 4.2. Experiment of masked-based parallel generation

### 4.2.1. Dataset

We trained the masked-based parallel speech generation model with 54k hours of Librilight-Large dataset [31]. The data was chunked into a maximum 30s for each training sample. For inference, we utilized the short-split of LibriSpeech-test-clean [33] data set.

### 4.2.2. Model configuration

For semantic token extraction, we used the Hubert-base model [9]. The semantic and retrained codec tokens were generated at 50 tokens per second, while the official Encodec tokens were at a rate of 75 tokens per second. To align the timing between official Encodec tokens and semantic tokens, we up-sampled the embeddings of the semantic tokens to match the official Encodec token rate. The model has 12 layers of Conformer, each layer has 16 attention heads, 1024 dimension of embeddings, 4096 feedforward dimensions, with a convolution kernel size of 5 and rotary positional embeddings, similar as [5]. During decoding, we applied 5 iterations for the first codec layer. For annealing CFG configuration, we set the initial and final guidance coefficients to 0 and 2, respectively. For the rest of the RVQ layers, we used

**Table 3**. Results of masked-based parallel speech generation

| ID | Codecs | Token rate | Continuation generation | | | Cross-speaker generation | | | | |
|----|--------|-----------|------|-------|--------|-------|-------|--------|------|------|
| | | | SIM-O | NISQA | WER (%) | SIM-O | NISQA | WER (%) | CMOS | SMOS |
| GT | GroundTruth | - | 0.67 | 3.87 | 0.96 | 0.70 | 3.87 | 0.96 | 0.29 | 4.75 |
| S1 | Encodec-official | 75 | 0.50 | 3.17 | 1.54 | 0.55 | 3.31 | 1.79 | -0.79 | 4.29 |
| S2 | Baseline-16kHz | 50 | 0.59 | 3.33 | 1.27 | 0.58 | 3.52 | 1.85 | -0.38 | **4.69** |
| S3 | Baseline-Vocos-16kHz | 50 | **0.61** | 3.47 | **1.22** | 0.58 | 3.76 | 1.86 | -0.31 | 4.52 |
| S4 | DAC-16kHz | 50 | 0.54 | **3.80** | 1.37 | 0.58 | **3.99** | **1.78** | -0.09 | 4.43 |
| S5 | DAC-Vocos-16kHz | 50 | 0.54 | 3.74 | 1.28 | **0.59** | 3.97 | 1.79 | **0.00** | 4.57 |

the greedily decoding without iterations. Overall, it requires 12 forward passes to predict the tokens for all 8 RVQ layers.

The model was trained using Adam optimization, with a batch size of 32. We employ the linear decay learning rate scheduler with 10k steps of warmup and a peak learning rate of 1e-4. The model trained with official Encodec tokens used 10 epochs, while other codec tokens utilized 5 epochs.

### 4.2.3. Results

Table 3 presents the evaluation results. In continuation generation, we used the first 3 seconds of speech from the same utterance as the prompt. For cross-speaker generation, we employed 3 seconds of speech from a different utterance as the prompt. Our evaluation metrics for continuation generation include SIM-O [40], NISQA score [41] and WER. Additionally, for cross-speaker generation, we included CMOS (comparative mean option score) and SMOS (similarity mean option score) to assess human perception of speech naturalness and speaker similarity in the generated speech. The testing procedure of CMOS and SMOS followed the same protocol as described in [4], with 10 subjects participating each test and 10 randomly selected samples for each condition.

In general, generated speech using the retrained codec tokens outperformed the official Encodec tokens across most evaluation metrics. The WER results of generated speech were similar across all codec tokens, likely because we used oracle semantic tokens as conditions. From NISQA scores, we observed the following trends: i) The Vocos decoding outperformed the waveform-based decoding in the baseline codec model; ii) DAC decoding exhibited similar performance to Vocos decoding, with both surpassing the baseline decoding. Moreover, the CMOS score in cross-speaker generation showed a robust correlation with the NISQA score. These results suggested that employing the snake activation function for waveform decoding had a similar impact on speech naturalness as predicting the STFT coefficients. Comparing Baseline-Vocos-16kHz and DAC-Vocos-16kHz, we observed that the quantization of DAC yields better speech naturalness.

In terms of SIM-O results, there was a slight discrepancy between continuation generation and cross-speaker generation when comparing the baseline-16kHz and DAC-16kHz.

However, considering that all SIM-O scores fell within a narrow range, the discrepancy can be ignored. The SMOS scores indicated that baseline-16kHz has the best speaker similarity performance, followed by DAC-Vocos-16kHz.

### 4.3. Experiment of AR + NAR generation models

#### 4.3.1. Dataset

We trained both the AR and NAR models with 54k hours of Librilight-Large dataset [31]. For the AR model, data was chunked into samples between 10 and 20 seconds, while for NAR model we used samples of up to 30 seconds. The phoneme data was obtained from the ASR model and the phoneme alignment tool presented in [4] with a frame size of 30ms, whereas the consecutive repetitions of the phonemes were removed. For inference, we also used the short-split of LibriSpeech-test-clean [33] data set.

#### 4.3.2. Model configuration

We adopted the same Transformer architecture as described in the VALL-E work [4] for both AR and NAR models. This architecture consisted of 12 layers, with each layer containing 16 attention heads. The embedding dimension is 1024, and the feed-forward layer dimension is 4096.

All models were trained for 800k steps with a 6k codec token batch size across 16 GPUs, except the NAR model with Encodec tokens, trained for 560k steps. Using the AdamW optimizer, AR and NAR models underwent optimization with a linear decay learning rate scheduler, 32k warmup steps, and a peak rate of 5e-4.

#### 4.3.3. Results

We present the evaluation results in Table 4. For both continuation and cross-speaker generation, we experimented with three different temperature settings during AR model inference. Notably, different temperatures yielded optimal results for different systems.. For each temperature setting, both DAC-16kHz and DAC-Vocos-16kHz demonstrated equally outstanding performance across all evaluation metrics. An unexpected observation from Table 4 was that the Baseline-16kHz and Baseline-Vocos-16kHz exhibited significantly

**Table 4**. Results of AR + NAR models-based speech generation

| ID | Codecs | Token rate | Continuation generation | | | Cross-speaker generation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | SIM-O | NISQA | WER (%) | SIM-O | NISQA | WER (%) | CMOS | SMOS |
| GT | GroundTruth | - | 0.67 | 3.87 | 0.96 | 0.70 | 3.87 | 0.96 | 0.61 | 4.77 |
| | | | | AR model sampling temperature: 1.0 | | | | | | |
| S1 | Encodec-official | 75 | 0.43 | 3.21 | **4.73** | 0.47 | 3.25 | 3.53 | -1.03 | 4.15 |
| S2 | Baseline-16kHz | 50 | 0.46 | 3.35 | 10.20 | 0.48 | 3.32 | 11.25 | -0.73 | **4.39** |
| S3 | Baseline-Vocos-16kHz | 50 | 0.46 | 3.52 | 10.10 | 0.48 | 3.53 | 11.31 | -0.70 | 4.37 |
| S4 | DAC-16kHz | 50 | **0.48** | **3.78** | 4.97 | **0.49** | **3.72** | 3.31 | **0.01** | 4.33 |
| S5 | DAC-Vocos-16kHz | 50 | **0.48** | 3.76 | 4.95 | **0.49** | **3.72** | **3.28** | 0.00 | 4.33 |
| | | | | AR model sampling temperature: 0.9 | | | | | | |
| S1a | Encodec-official | 75 | 0.44 | 3.23 | 3.88 | 0.47 | 3.22 | 3.38 | - | - |
| S2a | Baseline-16kHz | 50 | 0.46 | 3.35 | 6.94 | 0.48 | 3.31 | 8.52 | - | - |
| S3a | Baseline-Vocos-16kHz | 50 | 0.47 | 3.52 | 6.97 | 0.49 | 3.52 | 8.63 | - | - |
| S4a | DAC-16kHz | 50 | **0.48** | **3.76** | **3.08** | 0.49 | **3.75** | 3.03 | - | - |
| S5a | DAC-Vocos-16kHz | 50 | **0.48** | 3.74 | 3.09 | **0.50** | 3.73 | **2.91** | - | - |
| | | | | AR model sampling temperature: 0.8 | | | | | | |
| S1b | Encodec-official | 75 | 0.42 | 3.23 | 4.59 | 0.45 | 3.16 | 9.80 | - | - |
| S2b | Baseline-16kHz | 50 | 0.46 | 3.35 | 5.39 | 0.48 | 3.23 | 8.40 | - | - |
| S3b | Baseline-Vocos-16kHz | 50 | 0.47 | 3.54 | 5.30 | 0.48 | 3.46 | 8.55 | - | - |
| S4b | DAC-16kHz | 50 | 0.48 | **3.74** | **3.16** | 0.49 | 3.69 | 3.82 | - | - |
| S5b | DAC-Vocos-16kHz | 50 | **0.49** | 3.73 | 3.17 | 0.49 | 3.68 | 3.89 | - | - |

worse WER performance compared to other codec settings, despite having similar WER for the reconstruction in Table 2.

To investigate this issue, we analyzed the logarithmic distribution of the first-layer codec token utilization rate using LibriSpeech-test-clean, as depicted in Fig. 4. The Baseline-16kHz curve sharply declined at around 700, indicating the lowest code utilization rate among the three codec systems (Encodec-official and DAC-16kHz used 850 and 1024 codes, respectively). This low code utilization rate likely contributed to the poor WER performance. We infer that this low code utilization is related to the underlying VQ scheme: our baseline model used standard EMA update, while the rest applied techniques to benefit codebook utilization. Using a lower temperature during AR model inference significantly improved the WER performance for Baseline-16kHz. The sharper sampling distribution resulting from a lower temperature may mitigate hallucination issues, particularly when the number of used codes is much smaller than the number of classes in the AR model's classification layer. However, we found that the sampling temperature must be within a specific range; an excessively small value could worsen the WER.

We reported subjective evaluation results when AR model sampling temperature was at 1.0. The same 10 participants from the masked-based parallel speech generation experiments were invited for this test, following the same protocol. The results indicated a positive correlation between CMOS scores and NISQA scores, while SMOS scores fall within a tight range, with Baseline-16kHz achieving the highest score.
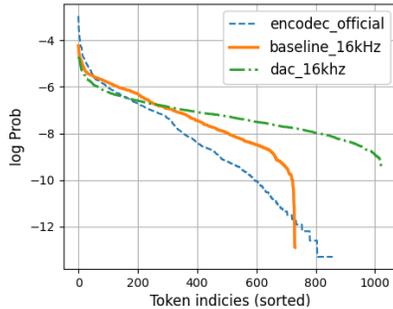


**Fig. 4**. The log-scale distribution of 1st layer codec tokens.

## 5. CONCLUSION

We explored various neural codec systems for the SLM-driven speech generation task. Our investigation included training our own baseline codec systems, as well as variants based on existing high-performing codec systems, including Encodec, Vocos, and DAC. We integrated the codec tokens from these codec systems into two types of SLM-based speech generation systems: masked-based parallel speech generation and AR + NAR models-based text to speech generation systems. Our experiment results revealed that DAC models perform exceptionally well overall for SLM-based speech generation. Vocos served as a competitive vocoder, demonstrating similar performance as the DAC decoder. Interestingly, we observed that the speech reconstruction quality was highly correlated with the naturalness of the generated speech, but this correlation did not hold true for speaker similarity and speech intelligibility.

# 6. REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, et al., "Language models are few-shot learners," in *arXiv: 2005.14165*, 2020.

[2] OpenAI, "GPT-4 technical report," in *arXiv: 2303.08774*, 2024.

[3] Z. Borsos, Raphael Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, et al., "AudioLM: a language modeling approach to audio generation," in *arXiv: 2209.03143*, 2023.

[4] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, et al., "Neural codec language models are zero-shot text to speech synthesizers," in *arXiv: 2301.02111*, 2023.

[5] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," in *arXiv: 2305.09636*, 2023.

[6] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, et al., "Audiopalm: A large language model that can speak and listen," in *arXiv: 2306.12925*, 2023.

[7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv: 1609.03499*, 2016.

[8] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervisedlearning of discrete speech representations," in *International Conferenceon Learning Representations (ICLR)*, 2020.

[9] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learningby masked prediction of hidden units," in *arXiv:2106.07447*, 2021.

[10] Y. Chung, Y. Zhang, W. Han, C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-bert: Combining contrastive learning and masked language modelingfor self-supervised speech pre-training," in *IEEE Automatic SpeechRecognition and Understanding Workshop, ASRU*, 2021.

[11] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," in *arXiv:2107.03312*, 2021.

[12] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," in *arXiv:2210.13438*, 2022.

[13] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

[14] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, et al., "On generative spoken language modeling from raw audio," *Transactions of the Association forComputational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[15] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems, NeurIPS*, 2017.

[16] H. Siuzdak, "Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis," in *International Conference on Learning representations, ICLR*, 2024.

[17] Z. Liu, H. Tilman, and U. Masahito, "Neural networks fail to learn periodic functions and how to fix it," in *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[18] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, et al., "Naturalspeech 3: Zero-shot speech synthesiswith factorized codec and diffusion models," in *arXiv:2403.03100*, 2024.

[19] H. Liu, X. Xu, Y. Yuan, M. Wu, W. Wang, and M. D. Plumbley, "SemantiCodec: An ultra low bitrate semanticaudio codec for general sound," in *arXiv: 2405.00233*, 2024.

[20] Y. Pan, L. Ma, and J. Zhao, "Promptcodec: High-fidelity neural speech codec using disentangled representation learningbased adaptive feature-aware prompt encoders," in *arXiv:2404.02702*, 2024.

[21] Y. Ren, T. Wang, J. Yi, L. Xu, J. Tao, C. Y. Zhang, and J. Zhou, "Fewer-token neural speech codec with time-invariant codes," in *arXiv:2310.00014*, 2024.

[22] S. Ahn, B. J. Woo, M. H. Han, C. Moon, and N. S. Kim, "Hilcodec: High fidelity and lightweight neuralaudio codec," in *arXiv:2405.04752*, 2024.

[23] H. Li, L. Xue, H. Guo, X. Zhu, Y. Lv, et al., "Singlecodec: Single-codebook speech codec towards high-performance speech generation," in *arXiv:2406.07422*, 2024.

[24] Haibin Wu, Ho-Lam Chung, Yi-Cheng Lin, Yuan-Kuei Wu, Xuanjun Chen, et al., "Codec-SUPERB: An in-depth analysis of sound codec models," in *arXiv:2402.13071*, 2024.

[25] H. Wu, X. Chen, Y. Lin, K. Chang, H. Chung, et al., "Towards audio language modeling - an overview," in *arXiv:2402.13236*, 2024.

[26] A. Ziv, I. Gat, G. L. Lan, T. Remez1, F. Kreuk1, et al., "Masked audio generation using a single non-autoregressive transformer," in *International conference on learning represenations (ICLR)*, 2024.

[27] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *arXiv:2207.12598*, 2024.

[28] Qitong Wang and Themis Palpanas, "Seanet: A deep learning architecture for data series similarity search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12972–12986, 2023.

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

[30] H. Chang, H. Zhang, L. Jiang, C. Liu, and B. Freeman, "Maskgit: Masked generative image transformer," in *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[31] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Librilight: A benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673, `https://github.com/facebookresearch/libri-light`.

[32] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifigan: generative adversarial networks for efficient and high fidelity speech synthesis," in *Proceedings of the 34th International Conference on Neural Information Processing Systems, NeurIPS*, 2020.

[33] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[34] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 2001, vol. 2, pp. 749–752 vol.2.

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

[36] Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines, "Visqol v3: An open source production ready objective speech and audio metric," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[37] Robert Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE pacific rim conference on communications computers and signal processing*. IEEE, 1993, vol. 1, pp. 125–128.

[38] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," in *arXiv:2110.13900*, 2022.

[39] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2020.

[40] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, et al., "Voicebox: Text-guided multilingual universal speech generation at scale," in *arXiv:2306.15687*, 2023.

[41] G. Mittag, B. Naderi, A. Chehadi, and S. Moller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," in *arXiv:2104.09494*, 2021.