DualCodec: A Low-Frame-Rate, Semantically-Enhanced Neural Audio Codec for Speech Generation

Anonymous submission to Interspeech 2025

Abstract

Neural audio codecs form the foundational building blocks for language model (LM)-based speech generation. Typically, there is a trade-off between frame rate and audio quality. This study introduces a low-frame-rate, semantically enhanced codec model. Existing approaches distill semantically rich self-supervised (SSL) representations into the first-layer codec tokens. This work proposes ***DualCodec***, a dual-stream encoding approach that integrates SSL and waveform representations within an end-to-end codec framework. In this setting, DualCodec enhances the semantic information in the first-layer codec and enables the codec system to maintain high audio quality while operating at a low frame rate. Note that a low-frame-rate codec improves the efficiency of speech generation. Experimental results on audio codec and speech generation tasks confirm the effectiveness of the proposed DualCodec compared to state-of-the-art codec systems, such as Mimi Codec, DAC, Encodec, and SpeechTokenizer. Demos are available at: `https://dualcodec.github.io`.

Index Terms: Neural Audio Codec, Speech Generation, Self-Supervised Feature, Low Frame Rate

1. Introduction

Neural audio codec is a technique to compress audio signals into a series of discrete codes for efficient data storage and transmission [1, 2, 3]. Recently, they are more frequently utilized as the tokenizers and de-tokenizers of speech language models (SLMs). These SLMs are inspired by the success of large language models and have shown impressive results in text-to-speech (TTS). In a typical SLM framework like VALL-E [4], a neural audio codec such as Encodec [2] encodes waveform signal into hierarchical discrete speech tokens with multiple layers of codebook tokens. The first-layer codebook tokens are predicted by an autoregressive (AR) model conditioned on text, and the remaining codebook layers are predicted via a non-autoregressive (NAR) model conditioned on the first-layer codebook tokens. Then, the codec decoder converts the speech tokens into audio.

Although this SLM framework has impressive zero-shot TTS capabilities, it still suffers from problems like inaccurate speech *content*, limited speech generation *quality*, and slow inference *speed* [5, 6, 7]. These three problems are closely related to the speech tokens. Motivated by recent works on improving each of these aspects [5, 7, 8], we summarize important design principles for a practical speech generation-oriented neural audio codec:

- Semantic enhancement: Self-supervised (SSL) speech features have shown to benefit various downstream tasks [9, 10]. Previous codec work SpeechTokenizer [5] has incorporated SSL feature in neural audio codec via semantic distillation.
- Low frame rate: A low token rate decreases the sequence length, reducing the speed and resources to train and inference SLMs. Both single-codebook [11, 1] and low frame-rate [7] codecs serve this purpose, but low-frame-rate codecs deliver higher speedup.
- Audio quality: A high codec reconstruction quality is essential for SLM's generation quality [6]. This becomes challenging for low-token-rate audio codecs.

Table 1: *A high-level comparison between codec systems.*

	Semantic Enhancement	Audio Quality	Frame Rate
Encodec	✗	Good	75Hz
SpeechTokenizer	✓ (distill)	Good	50Hz
DAC	✗	**Great**	75Hz/50Hz
Mimi	✓ (distill)	Good	**12.5Hz**
DualCodec	**✓ (dual encoding)**	**Great**	**12.5Hz/25Hz**

A comparison of the relevant existing codec models is presented in Table 1. Some of the existing models attempt to model semantic information in the codec explicitly by distilling SSL representation to codec. Also, existing models have a trade-off between audio quality and frame rate.

We argue that the three design principles can be integrated into a single framework: incorporating an explicit semantic-related codec layer, maintaining a low frame rate, and preserving high audio quality. To achieve this, we propose **DualCodec**, which unifies SSL and waveform representations in a single framework using dual encoding. In this framework, the code from the first layer is semantically enhanced directly from SSL features. Rather than making a trade-off between frame rate and audio quality, DualCodec enables the model to achieve a low frame rate while maintaining high audio quality. Additionally, we release the training and inference code for a 12.5 Hz codec. To the best of our knowledge, this is the first open-source 12.5 Hz low bit-rate codec[1].

2. Related Works

Vanilla neural audio codecs [2, 8] consist of an encoder, a residual vector quantization (RVQ) module, and a decoder. In this framework, only waveform is utilized as input in both training and inference. To serve codec systems better for SLMs, the following three important design decisions, including semantic enhancement, low token rate, and audio quality have been investigated in previous works.

[1]The Mimi codec is the first open-weight 12.5Hz codec, but its data and training codes are not available. We train on a public dataset and release our models and training codes.

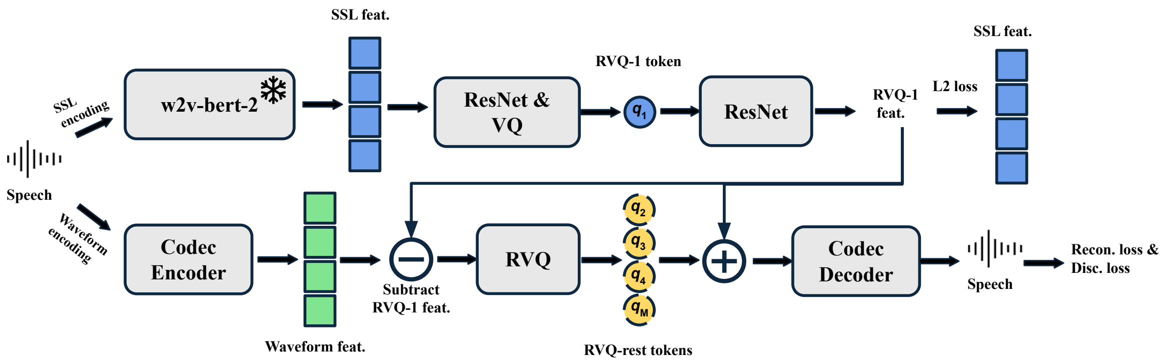

Figure 1: *The dual encoding method for neural audio codecs. The upper stream is SSL encoding, and the lower stream is waveform encoding. Given a speech input, the SSL feature is obtained from a pretrained w2v-bert-2 model and then encoded as the codec's first-layer token (RVQ-1). The remaining RVQ layers encodes the residual between the waveform feature and the RVQ-1 feature, and outputs audio. The framework is trained end-to-end requiring an additional L2 SSL feature loss in addition to codec training losses.*

2.1. Semantic Enhancement

The discrete tokens extracted from self-supervised (SSL) speech representations are commonly referred to as semantic tokens. These semantic tokens are extracted by k-means or vector quantization (VQ) on self-supervised (SSL) representations [12, 13]. Previous studies reveal that they possess rich phonetic and semantic information, reduces the model prediction complexity, but cannot accurately reconstruct audios due to a lack of acoustic traits like speaker identity [14, 12]. Audio codec tokens, on the other hand, contain more complex information supporting waveform reconstruction. Because of this information complexity, SLMs that predict vanilla audio codec tokens are known to be more unstable in their intelligibility than semantic token-based systems [5, 15, 12]. This happens primarily in the AR model because the AR generation can accumulate prediction errors.

Previous work SpeechTokenizer [5] proposed to unify the two types of tokens by enhancing the first-layer audio codec token (RVQ-1) through semantic distillation. Specifically, building upon the Encodec model [2], the approach introduces a semantic distillation loss between the RVQ-1 codebook vector and a certain layer HuBERT [16] hidden feature extracted from the same speech input. However, we find that the distilled tokens still lack semantic content accuracy, and have not been extensively verified in SLMs, especially multilingual SLMs.

2.2. Token Rate

Vanilla neural audio codecs operate at more than 4kbps bitrate and above 50Hz frame rates [2, 8, 3]. Lately, there has been a surge in research efforts focused on designing low bit-rate codec systems [7, 17, 1, 18]. These low-bitrate codec systems benefit the SLM efficiency by reducing the speech token length. In particular, the recent neural audio codec Mimi [7] operates at only 12.5Hz, a 6x reduction to the original 75Hz Encodec. Mimi uses SpeechTokenizer's semantic distillation with increased stride sizes and codebook sizes based on Encodec. We still find it has speech reconstruction artifacts especially at low bitrates.

2.3. Audio Quality

There has been several attempts to improve the audio reconstruction quality over vanilla systems like Encodec. Descript-Audio-Codec (DAC) [8] addressed codebook collapse problem by reducing the codebook latent dimensions to a very small value for quantization, and applied cosine similarity matching on L2 normalized codebooks. It also replaces the ReLU activation function with the snake activation function [19], offering benefits for reconstructing periodic signals. Some recent works explored using Transformer as replacements for CNN modules [1, 7]. We incorporate the DAC architecture in this work, and leave Transformer codecs as future investigations.

3. Method: Dual Encoding

As shown in Figure 1, our system consists of two encoding streams: an SSL encoding stream and a waveform encoding stream.

- The SSL encoding stream captures *semantic-rich* information to the first-layer codec tokens by directly encoding from SSL feature.
- The waveform encoding stream encodes and decodes *high-quality* audio with the proven DAC framework.
- We apply downsampling to both streams to achieve a *low frame rate*.

By using the two encoding streams, we obtain semantic-rich RVQ-1 tokens, with remaining layers (RVQ-rest) focused on the *remaining* acoustic aspects in the waveform feature. This "disentanglement" is achieved by subtracting RVQ-1 feature from waveform feature, before obtaining RVQ-rest tokens. Finally to decode audio, the RVQ-1 feature is re-summed to the codebook vectors of RVQ-rest. For SLM training, both encoding streams are required to obtain training tokens. During SLM inference, only the codec decoder is used to produce audio.

3.1. SSL Encoding

The SSL encoding stream contains a pretrained SSL model, a ResNet encoder, a vector quantization (VQ) module and a ResNet decoder. This architecture is analogous to a VQ-VAE [20], inspired by the RepCodec tokenizer [13] which first applied VQ-VAE to discretizing SSL features.

SSL Model. The SSL model is used here to obtain semantic-rich representations. We use normalized 16th layer w2v-BERT-2.0 [21] feature following [22]. The model outputs 50Hz feature from 16kHz waveforms with a 600M-parameter Transformer [23] network. The SSL model is frozen during training and inference.

ResNet Encoder and Decoder. These networks are used to process the SSL feature before and after the VQ module. This allows the VQ tokens to capture more complex semantic patterns. The decoder mirrors the encoder. Both models contain stacked ConvNeXt [24] blocks, which are the latest ResNet [25] variants. There's no down-sampling or up-sampling operation in these ResNet modules.

VQ Module. This module discretizes latent represetation $\mathbf{Z} \in \mathbb{R}^{H \times T}$ into a 1D token sequence $RVQ\_1 \in \mathbb{Z}^{1 \times T}$, where H is the hidden dimension and T is the feature length. We use the VQ formulation in DAC. Formally, RVQ_1 is computed by finding the closest codebook vector to the projected input: $RVQ\_1 = \arg\min_k ||\ell_2(W_{\text{in}}\mathbf{Z}) - \ell_2(e_k)||_2$. Here, $W_{\text{in}} \in \mathbb{R}^{D \times H}$ is the input projection matrix with $D = 8, H = 1024$, ℓ_2 is the L2-normalizaton, $e_1, e_2, ..., e_k$ are codebook vectors, $e_k \in \mathbb{R}^{H \times T}$. The continuous feature is $RVQ\_1\_feat = \text{ResNet}(e_k)$.

3.2. Waveform Encoding

The waveform encoding stream is inspired by existing neural audio codecs. We adopt the DAC [8] architecture, comprising a codec encoder, an RVQ module, and a codec decoder.

Codec encoder and decoder. The codec encoder and decoder are CNN networks with snake activation function [19]. The encoder contains a series of strided convolution layers to downsample the waveform into feature resolution. The decoder mirrors the encoder's structure, replacing strided convolutions with upsampling transposed convolutions to produce waveform.

RVQ module. This module has $N - 1$ layers of VQ. Each VQ layer quantizes the residual error of the previous layer [3]. The input to this module is the residual between the waveform feature and the RVQ_1_feat. It discretizes into RVQ_rest $\in \mathbb{Z}^{(N-1) \times T}$. After obtaining the RVQ_rest tokens, their codebook vectors e_k are added together with RVQ_1_feat. This continuous feature summarizes SSL encoding and waveform encoding, and is used as input to the codec decoder. We employ RVQ dropout [2] during training. That is, we only use the first q quantizers each time, where $q \in [0, N - 1]$ is randomly chosen. When $q = 0$, only the SSL encoding stream is used, allowing the model to vocode the RVQ-1 tokens.

Frame rate. Our framework operates at low frame rate options of 25Hz and 12.5Hz, with 24kHz audio input. We release models of both frame rates to support diverse application demands. To output a 25Hz frame rate, the encoder uses 4 CNN blocks with strides $[4, 5, 6, 8]$, giving 24000Hz $\div (4 \times 5 \times 6 \times 8) =$ 25Hz. The 12.5Hz version uses strides $[4, 5, 6, 8, 2]$. We also downsample the 50Hz SSL feature into our frame rates using simple 1D average pooling with $kenel\_size = stride\_size = downsampling\_factor$. The downsampling_factor is 2 for 25Hz, is 4 for 12.5Hz.

3.3. Training objective

The dual encoding framework is trained end to end. It is trained on an added SSL reconstruction loss [13] on top of the GAN training objective from DAC [8]: spectrogram reconstruction loss, quantization loss, and adversarial loss.

SSL reconstruction loss. This is an MSE losss between the reconstructed SSL feature and the input SSL feature. Both features are either the 12.5Hz or 25Hz downsampled version.

Spectrogram reconstruction loss. This is a multi-scale Mel Spectrogram loss between the input and reconstructed audio.

Quantization loss The codebooks are trained with an L1 loss between features before and after quantization. There's also a commitment loss with a weight of 0.25. They both employ the stop-gradient technique [3].

Adversarial loss We use Multi-Period Discriminator (MPD) and Multi-Scale STFT Discriminator (MS-STFTD) [8, 2]. A L1 feature matching loss is employed in all intermediate layers between generated and ground truth samples [8].

4. Experiments

4.1. Model training setup

We use the 100K-hour multilingual, 24kHz speech dataset Emilia [26, 27], and 8 A100 GPUs for training. Each codec model is trained for 500K steps. Each TTS model is trained for 600K steps. There are $N = 8$ codebook layers in DualCodec.

4.2. Semantic content analysis

Table 2: *WER Results of codec-reconstructed RVQ-1 audio.*

ID	Rate	Method	RVQ-1 Config	EN WER(%)↓	ZH WER(%)↓
A1	-	Ground-Truth	-	2.13	1.25
A2	50Hz	SpeechTokenizer	1024 EMA	14.9	83.2
B1	25Hz	DAC	1024 Proj	55.4	46.4
B2	25Hz	w/ Distill	1024 Proj	28.4	26.4
B3	25Hz	w/ Dual encoding	1024 Proj	**5.59**	**6.52**
C1	25Hz	DAC	16384 Proj	31.8	21.0
C2	25Hz	w/ Distill	16384 Proj	17.8	14.4
C3	25Hz	w/ Dual encoding	16384 Proj	**2.98**	**2.91**
D1	12.5Hz	w/ Dual encoding	16384 Proj	6.94	6.36

Metrics. We evaluate the semantic preservation of the RVQ-1 tokens by reporting the ASR Word Error Rate (WER) on the codec-reconstructed audio, using only RVQ-1. The evaluations leverage Whisper-large-v3 for English (EN) and Paraformer-zh for Chinese (ZH), tested on the Seed-TTS-Eval [28] dataset. The results are summarized in Table 2.

Group A baselines. Group A (A1 and A2) models are ground-truth and official SpeechTokenizer checkpoint, respectively. The A2 model has semantic distillation and 1024 EMA codebooks. While it is trained on English-only data, we still report its Chinese performance and find that its RVQ-1 has extremely high Chinese WER. Our listening test suggests that its RVQ-1 lacks pitch information, which explains because pitch information is critical for Chinese understanding.

Effect of Dual Encoding. The DAC framework at 25Hz with a 1024 projection codebook (B1) yields WERs of 55.4 (English) and 46.4 (Chinese). We then add a semantic distillation loss from [5], this (B2) reduces WERs to 28.4 and 26.4. Dual encoding (B3) further improves performance, achieving WERs of 5.59 (English) and 6.52 (Chinese). These results highlight the effectiveness of dual encoding in significantly enhancing semantic preservation.

Effect of Larger Codebooks. Increasing the RVQ-1 codebook size to 16384 brings additional improvements. With dual encoding (C3), the WERs drop to 2.98 (English) and 2.91 (Chinese), closely approaching the ground truth (A1). Meanwhile, at a reduced frame rate of 12.5Hz, the dual encoding configuration (D1) achieves competitive results, with WERs of 6.94 (English) and 6.36 (Chinese).

4.3. Audio quality analysis

Metrics. In this section, we report the audio reconstruction quality of DualCodec. We use the full Librispeech-test-clean [29] data. Metrics include the Perceptual Evaluation of Speech Quality (PESQ) [30] (both the 8kHz narrow-band PESQ_nb, and 16kHz wide-band PESQ_wb), Short Term Objective Intelligibility (STOI) [31], Mel Cepstral Distortion (MCD) [32]. We also evaluate on the reference-free neural MOS predictor UT-MOS [33], a metric that highly correlates with human preferences [34, 1]. The subjective test is the MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [35]. We conduct the test with 8 participants rating 15 sets of audios reconstructions sampled from the same test set. We use open-source baselines DAC [8], Encodec [2], SpeechTokenizer[5], WavTokenizer [11], and Mimi [7]. We compare under a consistent setup

Table 3: *Audio reconstruction performance of neural audio codecs around 75token/s and 0.75kbps bitrate on LibriSpeech-test-clean.*

ID	System *(RVQ-1 size, RVQ-rest size)*	Bit(kbps)	Tok/s	#VQ	PESQ_nb↑	PESQ_wb↑	STOI↑	MCD↓	UTMOS↑	MUSHRA↑
E1	DAC-official 75Hz	0.75	75	1	1.46	1.18	0.75	6.00	1.32	26.0
E1	Encodec 75Hz	1.5	150	2	1.92	1.54	0.84	4.30	1.55	36.2
E3	SpeechTokenizer 50Hz	1.0	100	2	1.42	1.15	0.70	6.94	1.81	35.9
E4	WavTokenizer-large 75Hz	0.90	75	1	**2.54**	**2.05**	**0.89**	**3.99**	**3.87**	**81.0**
E5	Mimi 12.5Hz	0.83	75	6	2.51	1.99	0.89	4.13	3.43	72.8
F1	DAC-repro 25Hz *(1024+1024)*	0.75	75	3	2.58	2.06	0.89	3.93	3.29	68.8
F2	DAC-repro 12.5Hz *(1024+1024)*	0.75	75	6	**2.88**	**2.33**	**0.91**	**3.70**	**3.87**	**81.8**
G1	DualCodec 25Hz *(1024+1024)*	0.75	75	3	2.64	2.07	0.90	3.99	3.86	79.5
G2	DualCodec 25Hz *(16384+1024)*	0.85	75	3	2.92	2.32	0.91	3.61	4.08	86.2
G3	DualCodec 12.5Hz *(1024+1024)*	0.75	75	6	2.89	2.30	0.91	3.61	3.94	83.5
G4	DualCodec 12.5Hz *(16384+1024)*	0.80	75	6	2.94	2.33	0.91	3.65	4.04	85.2
G5	DualCodec 12.5Hz *(16384+4096)*	0.93	75	6	**3.11**	**2.54**	**0.92**	**3.33**	**4.11**	**88.2**

of 75 tokens per second and around 0.75kbps low bitrate [2]. Table 3 presents the results.

Baseline Systems. Among the baselines (group E), Encodec achieved the highest reference-based scores but has a low UTMOS of 2.34 which indicates low perceptual quality, and it operates at a higher bitrate of 1.5 kbps. In contrast, WavTokenizer-large, despite its lower bitrate of 0.9 kbps, performed competitively with similar reference-based scores, and the highest UTMOS = 3.87 among the baselines.

Reproduced DAC. Group F focuses on our retrained DAC codec modified for 25Hz and 12.5Hz frame rates. At the same 0.75 kbps bitrate, the 12.5 Hz DAC obtain much higher performance than 25Hz in every metric. This suggests that operating at a lower frame rate with more quantization layers is more effective than a larger frame rate with less quantization layers. Interestingly, the 12.5Hz DAC model outperforms all baseline models, suggesting *the effectiveness of the DAC framework especially at lower frame rates*.

DualCodec. Group G highlights the performance of DualCodec under various configurations of its RVQ codebooks.

First, models G1 and G3 utilize a codebook size of 1024 at each RVQ layer, enabling direct comparison with Group F models. Comparing G1 (DualCodec 25Hz) with F1 (DAC-repro 25Hz), G1 achieves similar performance across most objective metrics but demonstrates a significant improvement in UTMOS (3.86 vs. 3.29), indicating a noticeable enhancement in perceptual audio quality. Similarly, comparing G3 (DualCodec 12.5Hz) with F2 (DAC-repro 12.5Hz) shows that while objective metrics like PESQ and MCD are comparable, *DualCodec consistently delivers better perceptual quality as evidenced by its higher UTMOS scores*. These comparisons highlight the benefits of DualCodec's additional semantic encoding stream in enhancing perceptual audio quality.

We further examine the impact of increasing the RVQ codebook sizes in DualCodec, which slightly increases the bitrate while maintaining a consistent token rate of 75 tokens/s. Model G2, with a configuration of 16384 codebooks in RVQ-1 and 1024 in RVQ-rest, shows marked improvements over G1 in every metric. The trend continues with models G4 and G5, which explore configurations with 12.5Hz frame rates and larger codebooks in the waveform encoding stream. Model G5, with a configuration of 16384+4096 codebooks, achieves the best overall performance among all systems, with PESQ_nb=3.11, STOI=0.92, and UTMOS=4.11. *This result highlights that increasing the codebook size while leveraging lower frame rates can significantly enhance low-bitrate audio quality.*

[2] Bitrate quantifies how much data is used to represent audio signals. Codec model usually support multiple bitrate settings by RVQ dropout. Bitrate is calculated by ($\log_2$ codebook_size) $\times$ Tok/s $\times$ Num_vq.

4.4. SLM analysis

Table 4: *Codec-based SLM performance on Seed-TTS-Eval.*

SLM	Codec	EN WER↓	EN SIM↑	ZH WER↓	ZH SIM↑	RTF↓
GT	-	2.13	0.73	1.25	0.75	-
VALL-E	SpeechTokenizer	15.4	0.47	21.5	0.55	0.76
	Mimi	8.16	0.48	10.5	0.55	**0.16**
	DualCodec 25Hz	**3.40**	**0.57**	**2.49**	**0.67**	0.30
	DualCodec 12.5Hz	4.40	0.54	4.90	0.65	**0.16**
AR + SoundStorm	SpeechTokenizer	11.3	0.50	46.3	0.57	1.18
	Mimi	9.09	0.50	39.4	0.57	**0.34**
	DualCodec 25Hz	**3.56**	**0.67**	**2.93**	**0.75**	0.66
	DualCodec 12.5Hz	4.93	0.59	4.72	0.69	**0.34**

Metrics. We adopt VALL-E [4] and AR+SoundStorm [36] as SLM systems and train each SLM with different codec systems. VALL-E has 270M parameters in AR, 400M in NAR. AR+SoundStorm has 800M in AR, 300M in NAR. For DualCodec, we use models G2 and G5 in Table 3 for 25Hz and 12.5Hz, respectively. We report the WER and speaker similarity SIM-O (SIM) on Seed-TTS-Eval benchmark [28]. We report the real-time-factor (RTF) tested on an A100 GPU which correlates to the inference speed. Results are shown in Table 4.

Performance Comparisons. Table 4 demonstrates that DualCodec outperforms SpeechTokenizer and Mimi baselines in both SLMs performance. We attribute this to DualCodec's more accurate semantic content and better codec reconstruction quality. The AR+SoundStorm SLM paired with DualCodec 25Hz achieves the best performance, followed by DualCodec 12.5Hz. The comparison between DualCodec 25Hz and 12.5Hz reveals a clear tradeoff between quality and inference speed. DualCodec 25Hz consistently achieves lower WER and higher SIM scores, making it the ideal choice for tasks prioritizing accuracy and similarity. On the other hand, DualCodec 12.5Hz provides faster inference at the cost of different degrees of performance decrease. We also notice that Mimi and SpeechTokenizer have excessively large Chinese WERs in AR+SoundStorm. We suggest this is due to a lack of RVQ-1 semantic pitch information, which becomes more notable in SoundStorm because its NAR does not have text prompting.

5. Conclusion

We introduced DualCodec, a low-frame-rate, semantically-enhanced neural audio codec designed for efficient speech generation. By leveraging dual encoding, low frame rates and larger codebooks, DualCodec significantly improves semantic accuracy, audio reconstruction quality, and SLM efficiency. Future work will investigate methods to further increase the 12.5Hz semantic accuracy, scaling up the model and data, and exploring Transformer architecture. DualCodec also has the potential to be used in real-time multimodal LLM applications.

6. References

[1] H. Wu, N. Kanda, S. E. Eskimez, and J. Li, “Ts3-codec: Transformer-based simple streaming single codec,” *arXiv preprint arxiv:2411.18803*, 2024.

[2] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Transactions on Machine Learning Research*, 2023.

[3] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[4] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” *arXiv preprint arXiv:2301.02111*, 2023.

[5] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, “Speechtokenizer: Unified speech tokenizer for speech language models,” in *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

[6] J. Li, D. Wang, X. Wang, Y. Qian, L. Zhou, S. Liu, M. Yousefi, C. Li, C.-H. Tsai, Z. Xiao, Y. Liu, J. Chen, S. Zhao, J. Li, Z. Wu, and M. Zeng, “Investigating neural audio codecs for speech language model-based speech generation,” in *SLT*, 2024.

[7] A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour, “Moshi: a speech-text foundation model for real-time dialogue,” Kyutai, Tech. Rep., September 2024. [Online]. Available: http://kyutai.org/Moshi.pdf

[8] 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.

[9] M. Cui, D. Tan, Y. Yang, D. Wang, H. Wang, X. Chen, X. Chen, and X. Liu, “Exploring ssl discrete tokens for multilingual asr,” *CoRR*, vol. abs/2409.08805, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2409.08805

[10] J. Y. Lee, M. Jeong, M. Kim, J.-H. Lee, H.-Y. Cho, and N. S. Kim, “High fidelity text-to-speech via discrete tokens using token transducer and group masked language model,” in *Interspeech 2024*, 2024, pp. 3445–3449.

[11] S. Ji, Z. Jiang, X. Cheng, Y. Chen, M. Fang, J. Zuo, Q. Yang, R. Li, Z. Zhang, X. Yang, R. Huang, Y. Jiang, Q. Chen, S. Zheng, W. Wang, and Z. Zhao, “Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling,” *CoRR*, vol. abs/2408.16532, 2024.

[12] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin *et al.*, “AudioLM: a language modeling approach to audio generation,” in *arXiv: 2209.03143*, 2023.

[13] Z. Huang, C. Meng, and T. Ko, “Repcodec: A speech representation codec for speech tokenization,” 2024. [Online]. Available: https://arxiv.org/abs/2309.00169

[14] P. Mousavi, J. Duret, S. Zaiem, L. D. Libera, A. Ploujnikov, C. Subakan, and M. Ravanelli, “How should we extract discrete audio tokens from self-supervised models?” in *Interspeech 2024*, 2024.

[15] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma, Z. Gao, and Z. Yan, “Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens,” *ArXiv preprint arXiv:2407.05407*, 2024.

[16] W.-N. Hsu, B. Bolte, Y.-H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[17] 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 *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 2024.

[18] H. Li, L. Xue, H. Guo, X. Zhu, Y. Lv *et al.*, “Single-codec: Single-codebook speech codec towards high-performance speech generation,” in *Interspeech*, 2024.

[19] 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.

[20] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *NeurIPS*, 2017, pp. 6306–6315.

[21] S. Communication, “Seamless: Multilingual expressive and streaming speech translation,” *arXiv preprint arXiv:2312.05187*, 2023.

[22] Y. Wang, H. Zhan, L. Liu, R. Zeng, H. Guo, J. Zheng, Q. Zhang, X. Zhang, S. Zhang, and Z. Wu, “Maskgct: Zero-shot text-to-speech with masked generative codec transformer,” *arXiv preprint arXiv:2409.00750*, 2024.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17, 2017, p. 6000–6010.

[24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594692

[26] H. He, Z. Shang, C. Wang, X. Li, Y. Gu, H. Hua, L. Liu, C. Yang, J. Li, P. Shi, Y. Wang, K. Chen, P. Zhang, and Z. Wu, “Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation,” *SLT*, 2024.

[27] X. Zhang, L. Xue, Y. Gu, Y. Wang, J. Li, H. He, C. Wang, T. Song, X. Chen, Z. Fang, H. Chen, J. Zhang, T. Y. Tang, L. Zou, M. Wang, J. Han, K. Chen, H. Li, and Z. Wu, “Amphion: An open-source audio, music and speech generation toolkit,” in *IEEE Spoken Language Technology Workshop, SLT 2024*, 2024.

[28] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao, M. Gong, P. Huang, Q. Huang, Z. Huang, Y. Huo, D. Jia, C. Li, F. Li, H. Li, J. Li, X. Li, X. Li, L. Liu, S. Liu, S. Liu, X. Liu, Y. Liu, Z. Liu, L. Lu, J. Pan, X. Wang, Y. Wang, Y. Wang, Z. Wei, J. Wu, C. Yao, Y. Yang, Y. Yi, J. Zhang, Q. Zhang, S. Zhang, W. Zhang, Y. Zhang, Z. Zhao, D. Zhong, and X. Zhuang, “Seed-tts: A family of high-quality versatile speech generation models,” in *arXiv preprint arxiv:2406.02430*, 2024.

[29] V. Panayotov, G. Chen, D. Povey, and S. 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.

[30] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *ICASSP 2001*, 2001, pp. 749–752 vol.2.

[31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. 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.

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

[33] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, “Utmos: Utokyo-sarulab system for voicemos challenge 2022,” *Interspeech 2022*, 2022.

[34] J. Shi, J. Tian, Y. Wu, J. weon Jung, J. Q. Yip, Y. Masuyama, W. Chen, Y. Wu, Y. Tang, M. Baali, D. Alharhi, D. Zhang, R. Deng, T. Srivastava, H. Wu, A. H. Liu, B. Raj, Q. Jin, R. Song, and S. Watanabe, “Espnet-codec: Comprehensive training and evaluation of neural codecs for audio, music, and speech,” *SLT*, 2024.

[35] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webmushra — a comprehensive framework for web-based listening tests," *Journal of Open Research Software*, Feb 2018.

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