Hierarchical Representation Alignment Learning of Diffusion Transformers for Neural Audio Codec

Sang-Hoon Lee[1,2], Ha-Yeong Choi[3]

[1]Department of Software and Computer Engineering, Ajou University, Suwon, Korea
[2]Department of Artificial Intelligence, Ajou University, Suwon, Korea
[3]KT Corp., Seoul, Korea

Correspondence: sanghoonlee@ajou.ac.kr

Abstract

Despite recent progress in diffusion and conditional flow matching (CFM) models for low-resolution domains such as latent representations, their application to high-resolution data like raw waveform signals remains underexplored. Generative adversarial networks (GANs) have been the dominant approach in neural vocoder and neural audio codecs for realistic waveform generation. However, under low-bitrate conditions, these models suffer from degraded performance due to information loss caused by heavy compression and quantization, often resulting in mispronunciations. To address the aforementioned problem, we first leverage CFM to iteratively generate raw waveform in an extremely low-bitrate scenario. We then introduce hierarchical representation alignment learning (REPA-H) to enable efficient and robust CFM training. Furthermore, we propose dense vector quantization (DVQ), a novel factorized quantization method using a single quantizer. Our model, FlowTokenizer, outperforms state-of-the-art neural audio codecs in audio quality and semantic intelligibility under low-bitrate conditions, using only 25 tokens per second for 24 kHz waveform generation. Audio samples are available at `https://flowtokenizer.github.io/demo`.

1 Introduction

Recent advances in diffusion (Ho et al., 2020; Dhariwal and Nichol, 2021; Song et al., 2021) and conditional flow matching (CFM) (Lipman et al., 2022) models have demonstrated strong performance in generating data from low-resolution domains, such as latent representations (Yang et al., 2023b; Huang et al., 2023; Ju et al., 2024; Li et al., 2024; Lee et al., 2025a; Jiang et al., 2025b) and Mel-spectrograms (Kim et al., 2024; Le et al., 2024; Eskimez et al., 2024; Chen et al., 2025; Du et al., 2024). While end-to-end latent diffusion models (Vahdat et al., 2021) have also been investigated, the application to high-resolution signals, particularly raw waveforms, remains underexplored, especially in the context of end-to-end audio token training for neural audio codec using vector quantization (VQ) (Gray, 1984).

In contrast, generative adversarial networks (GANs) (Goodfellow et al., 2014; Kong et al., 2020; Lee et al., 2023) have been the dominant approach for neural audio codecs (Zeghidour et al., 2021), due to their ability to synthesize high-quality waveforms. However, GAN-based models still struggle in extremely low-bitrate scenarios (e.g., 12.5 Hz and 25 Hz), where reconstructing high-resolution waveform (e.g., 24,000 Hz) from heavily compressed tokens leads to degraded perceptual quality and intelligibility, often resulting in mispronunciations. Residual vector quantization (RVQ) (Zeghidour et al., 2021; Défossez et al., 2023; Kumar et al., 2023; Yang et al., 2023a; Défossez et al., 2024) has been introduced to enhance the representational capacity of neural audio codecs. While RVQ improves reconstruction quality by increasing codebook expressiveness, it also introduces additional complexity when applied to systems such as speech large language models.

To address the limitations of RVQ, recent neural audio codecs have explored the use of a single quantizer, enabling more natural integration with large language models (LLMs). BigCodec (Xin et al., 2024) and StableCodec (Parker et al., 2025) scale up the model size for a low-bitrate neural speech codec. WavTokenizer (Ji et al., 2025) compresses raw waveform into 40 or 75 tokens using only a single quantizer, while preserving perceptual quality. UniCodec (Jiang et al., 2025a) adopts a domain-adaptive codebook and MoE. X-codec2 further incorporates semantic information by concatenating semantic and acoustic representations before quantization. However, when operating at extremely low bitrates, such as 25 tokens per second (TPS), these models suffer from significant

Findings of the Association for Computational Linguistics: ACL 2026, pages 32410–32426

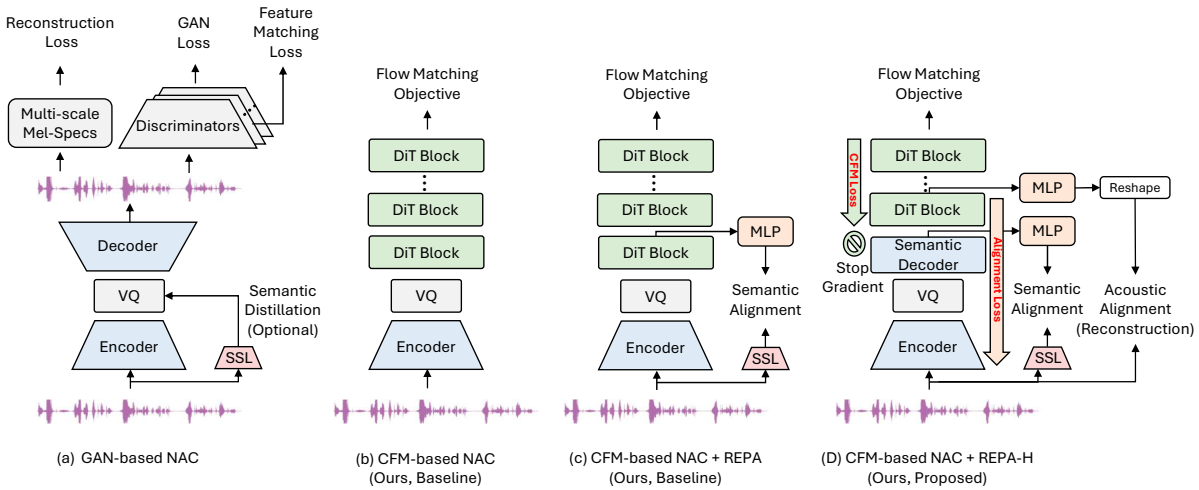

Figure 1: Comparison of neural audio codec (NAC) frameworks. (a) Conventional GAN-based NAC. (b) Our baseline CFM-based NAC. (c) CFM-based NAC with REPA. (d) The proposed REPA-H, which hierarchically aligns semantic and acoustic representations.

loss of semantic information, leading to high word error rate.

Meanwhile, diffusion-based waveform generation models have been investigated (Kong et al., 2021; Chen et al., 2021; Lee et al., 2022; Huang et al., 2022; Roman et al., 2023). More recently, CFM has been applied to raw waveform generation from pre-defined acoustic representation such as Mel-spectrogram or RVQ tokens (Lee et al., 2025b; Liu et al., 2025; Welker et al., 2025). (Lee et al., 2024) further accelerated CFM models via adversarial fine-tuning. Following the success of CFM in waveform synthesis, we shift our focus from GAN to CFM for training low-bitrate neural audio codec. We design a CFM-based codec that operates at extremely low-bitrate using a single quantizer and 25 TPS, and explore efficient modeling using only low-resolution features (<50 Hz) for 24 kHz waveform reconstruction. Furthermore, we leverage Diffusion Transformers (DiT) for raw waveform modeling, and incorporate representation alignment (REPA) (Yu et al., 2025) to distill self-supervised speech representations within the DiT layers, thereby enhancing the semantic capability of our neural audio codec.

Building on these design choices and findings, we introduce **FlowTokenizer**, a CFM-based neural audio codec that employs a single quantizer and operates at 25 TPS for waveform generation at a sampling rate of 24 kHz. To achieve this, we carefully design the encoder using causal Transformer and decoder using DiT. We further propose **hierarchical representation alignment learning (REPA-H)** to enhance training stability and efficiency by hierarchically aligning semantic to acoustic features while estimating vector fields. In addition, we introduce **dense vector quantization (DVQ)**, a novel factorized quantization method that jointly compresses semantic and acoustic representations using a single quantizer. Experimental results show that FlowTokenizer achieves superior performance in audio quality, reconstruction fidelity, and semantic intelligibility in low-bitrate scenarios.

2 FlowTokenizer

In this section, we present **FlowTokenizer**, a CFM-based neural audio codec tailored for low-bitrate scenarios, as illustrated in Figure 2. We first investigate the CFM-based neural audio codec including advantages, limitations (§2.1), and introduce novel techniques for efficient and robust training under low-bitrate scenarios (§2.2). Specifically, we adopt a representation alignment (REPA) learning strategy to efficiently train diffusion transformers, and extend it to high-frequency raw waveform modeling. The architecture details are described in (§2.3). Furthermore, we propose a novel vector quantization method, dense vector quantization (DVQ), for a single-layer factorized representation (§2.4). To accelerate sampling speed, we also apply adversarial fine-tuning with a fixed-step modification (§2.5).

2.1 CFM-based Neural Audio Codec

Previously, (Lee et al., 2025b; Liu et al., 2025; Welker et al., 2025; Choi and Lee, 2025) utilized CFM for raw waveform generation from pre-defined acoustic representation such as Mel-spectrogram or pre-trained RVQ tokens. They

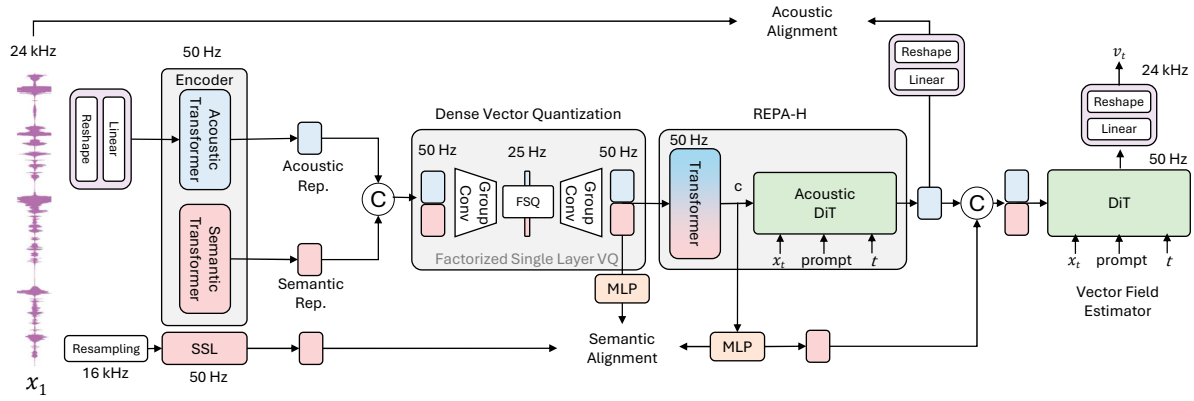

Figure 2: FlowTokenizer Architecture

demonstrated that CFM could refine the waveform signal with iterative sampling, resulting in high-fidelity reproduction, and using only CFM objective could accelerate the entire training speed compared to GAN-based models due to discriminator-free training. However, CFM has not yet been explored for training neural audio codec, where efficient and accurate end-to-end CFM training under low-bitrate constraints presents new challenges and opportunities.

Conditional Flow Matching CFM (Lipman et al., 2022) trains a neural network to approximate the transformation from a simple prior distribution to a complex target distribution via a time-dependent vector field. Specifically, it models a conditional vector field $v_\theta(t, x_t)$ to match the ideal target field $u_t(x_t|x_1)$, conditioned on data samples x_1. Formally, this objective can be expressed as:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t\sim U[0,1],x_t} \, ||u_t(x_t|x_1) - v_\theta(t, x)||_2^2 . \quad (1)$$

DiT-based Structure Following the success of diffusion transformer (DiT) in image and Mel-spectrogram generation tasks, we design a CFM-based neural audio codec using a DiT decoder structure as illustrated in Figure 1. Inspired by VoiceBox (Le et al., 2024), we adopt a prompting strategy that concatenates the noisy input x_t, and prompt x_1 with masking, and condition c (the output of VQ), and the diffusion time step t. However, we observe that the vanilla DiT (Peebles and Xie, 2023) fails to preserve semantic consistency in low-bitrate scenarios, resulting in high word error rate (WER) and poor reconstruction of high-frequency components.

Representation Alignment Recently, REPA (Yu et al., 2025) introduces representation alignment to guide diffusion training. In the original REPA, semantic representations from DINOv2 (Oquab et al., 2023) are used to supervise intermediate DiT layers, thereby improving both convergence speed and final performance. We also apply REPA to the training of proposed DiT-based neural audio codec. Specifically, we extract semantic representation from a middle layer of wav2vec 2.0, and guide early DiT layers to align with them, while the full DiT stack concurrently estimates the vector fields.

To further strengthen semantic fidelity, we combine an L_1 loss on the semantic embeddings with a cosine similarity loss:

$$\mathcal{L}_{\text{semantic}} = \mathbb{E}\|\mathbf{h}_t - \hat{\mathbf{h}}_t\|_1 + \mathbb{E}\big[-\log\sigma\big(\cos(\mathbf{h}_t, \hat{\mathbf{h}}_t)\big)\big], \quad (2)$$

where $\mathbf{h}_t$ is the teacher embedding from wav2vec 2.0, $\hat{\mathbf{h}}_t$ is the embedding predicted by the DiT. While REPA improves semantic reproduction under low-bitrate conditions, we found that the REPA loss could interfere with accurate vector field estimation. This necessitates careful tuning of the weight of REPA loss. Moreover, even with REPA, the model still fails to effectively reconstruct high-frequency components of the waveform.

2.2 Hierarchical Representation Alignment Learning (REPA-H)

To address these limitations, we propose hierarchical representation alignment learning (REPA-H), for effective CFM-based neural audio codec training by aligning from semantic to acoustic features hierarchically. With semantic REPA, we additionally use acoustic REPA as below.

Acoustic Alignment In a low-bitrate scenario, each token must encode both semantic and acoustic information, making it challenging to maintain fidelity across all aspects. However, the CFM objective (Eq. 1) and semantic loss (Eq. 2) are limited to guide acoustic details. To address this, we introduce a second stage of representation alignment

focused on acoustic representation. Specifically, we reconstruct the waveform via linear-reshape transformation from the output of the acoustic DiT, and apply multi-resolution Mel-spectrogram losses of DAC (Kumar et al., 2023) using multi-resolution windows of [32, 64, 128, 256, 512, 1024, 2048]. For efficient training, we segment the raw waveform signal using a sliding window of 65,536 frames for STFT, which effectively capture frequency information across the multi-resolution. Formally, it is defined as:

$$\mathcal{L}_{\text{acoustic}} = \sum_{i=1}^{N} \left[\left\| \log\left(\mathbf{M}_i(\hat{y})\right) - \log\left(\mathbf{M}_i(y)\right) \right\|_1 \right] \tag{3}$$

where $\hat{y}$ and y denote the predicted and target waveform segments respectively, $\mathbf{M}_i(\cdot)$ denotes the Mel-spectrogram computed with the i-th resolution STFT window, N is the number of multi-resolution levels (e.g., $N = 7$).

Disentangled Alignment Training However, we observe instability during the early stages of training: applying conditioning dropout to noisy x_t impairs the model's ability to accurately estimate acoustic features. Furthermore, fp16 training often yields NaN losses, destabilizing the overall acoustic loss computation. To mitigate these issues, we separately perform a forward pass of the semantic and acoustic DiT from the DiT blocks, and do not use conditioning dropout for this stage. Then, we additionally perform a second forward for entire DiT blocks with random condition dropout. Furthermore, CFM loss is only used for acoustic DiT and DiT blocks using stop gradient. This enhances the stability of training and improves the semantic and acoustic capability on the quantized layer and DiT blocks, leading to higher-fidelity and more intelligible speech generation.

Causal Semantic Transformer Structure We compare the types of semantic decoders including causal Transformer and DiT. While the DiT could improve the generative performance in terms of audio quality, it is difficult to optimize the model by guaranteeing both semantic and acoustic capability within DiT blocks. To address this, we employ a causal Transformer that directly predicts continuous semantic features, and then concatenate semantic features with hidden representation before fed to the DiT, as illustrated in Figure 2. This shows better semantic preservation in low-bitrate scenarios, resulting in lower WER while it slightly degrades the reproduction quality of acoustic domains.

2.3 Architecture Details

Input Transformation We adopt an efficient WaveNeXt-style (Okamoto et al., 2023; Wu et al., 2024) of waveform transformation to reshape the raw waveform into a low-resolution feature space. Specifically, we reduce the temporal resolution from 24 kHz to 50 Hz using a reshape-based transformation, where a 1D waveform signal of shape [B, 24000] is directly converted into 2D representation of shape [B, 50, 480]. Importantly, this transformation do not require any past or future context, unlike STFT which requires larger window size than the current frame. This property makes our approach suited for left-to-right context prediction, enabling real-time and streaming scenarios aligned with LLM.

Encoder We utilize eight causal Transformers as the acoustic encoder. The reshaped 2D features are fed to the acoustic encoder to extract the acoustic representations. For semantic encoder, we additionally use Transformers. Specifically, we utilize a massively multilingual speech (MMS) (Pratap et al., 2024) which is a large-scale pre-trained Wav2Vec 2.0[1] as the target semantic representation for multilingual extension. The acoustic features and semantic features are concatenated before quantization. A causal convolutional layer is applied to downsample feature into VQ space.

Decoder We utilize 6 blocks of causal semantic Transformers, and 6 blocks of DiT for acoustic DiT, and 12 blocks for vector field estimation. For REPA and final projection, we use AdaLN-zero followed by a linear projection layer to map the shape of representations.

Output Transformation We also utilize a WaveNeXt-style of waveform transformation for waveform-level vector field estimation. Specifically, the 2D features of shape [B, 50, 480] is converted into 1D waveform signal of shape [B, 24000] directly. Note that our model executes the ODE sampling on the 50 Hz resolution, and the quantization on the 25 Hz resolution with a single layer VQ without any additional upsampling layer for waveform generation.

[1]We leverage full-GPU processing for Wav2Vec feature extraction in an on-the-fly manner.

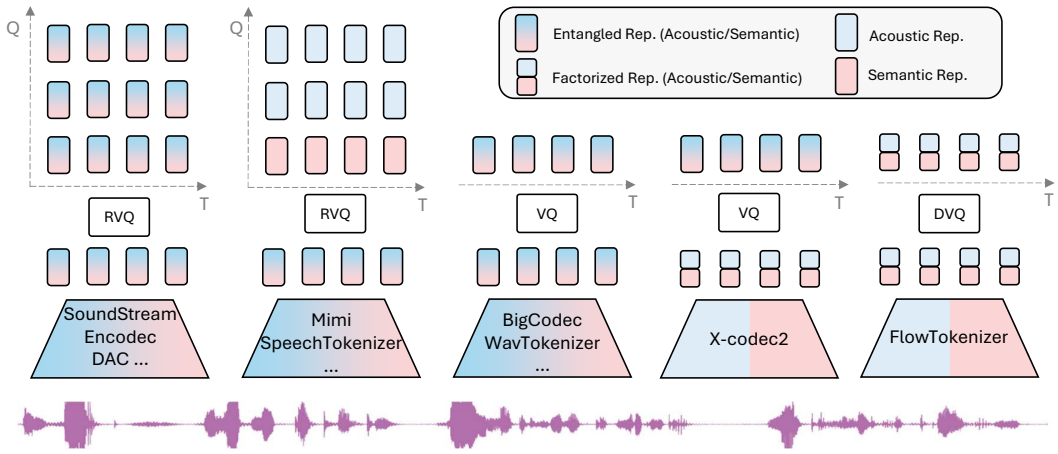

Figure 3: Comparison of representation structures in recent neural audio codecs

2.4 Dense Vector Quantization

We propose dense vector quantization (DVQ), a novel quantization method that factorizes semantic and acoustic components into a single quantized representation without relying on RVQ. While RVQ effectively encodes the information, it increases token lengths, leading to greater design complexity and inefficiencies in speech LLM applications. To reduce this limitation, BigCodec (Xin et al., 2024) and WavTokenizer (Ji et al., 2025) have explored single-layer VQ for efficient waveform reconstruction. X-codec2 (Ye et al., 2025) concatenated the semantic representation with the acoustic representation without residual modeling at a single layer for low-bitrate neural audio codec modeling. Following the philosophy of X-codec2, we also concatenate the semantic representation with the acoustic representation. Although this significantly improves the semantic capability, however, we found that X-codec2 uses a linear projection of the concatenated representation to match the latent dimension of finite scalar quantization (FSQ), which loses their intrinsic integrity of each representation.

In this regard, we carefully design the latent dimension of the codebook using group convolutional layers to ensure the cardinality of latent representation for each acoustic and semantic representation, and apply FSQ to this factorized representation as illustrated in Figure 3. This has the same effect as a dense connection, so we call this new VQ method as ***dense vector quantization***. Thanks to DVQ, the quantized representations could preserve both acoustic and semantic representations. Specifically, we use eight dimensions and four levels for FSQ, which can be expressed by [4, 4, 4, 4, 4, 4, 4, 4] yielding 65,536 tokens. Furthermore, by isolating the scalar corresponding to the semantic dimension [4, 4, 4, 4], we can extract 256 semantic tokens. This auxiliary semantic token can be used to compute an additional semantic loss, serving as a semantic guidance for neural codec language models.

2.5 Adversarial Fine-tuning

After training the model with CFM and REPA-H, we fine-tune the model with adversarial feedback and REPA-H losses for accelerating sampling speed as illustrated in Figure 7. While the pre-trained model could generate waveform signal in low-bitrate scenarios, it requires enough sampling steps and classifier-free guidance, which increases inference time. Following (Lee et al., 2024), we fix the sampling steps, and generate raw waveform signal from x_0 with ODE sampling using four step fixed generation. Then, we fine-tune the model with adversarial feedback, multi-resolution Mel-spectrogram reconstruction losses on the generated waveform $\mathcal{L}_{mel}$, and semantic REPA loss $\mathcal{L}_{semantic}$. Specifically, we utilized multi-period discriminator (MPD), multi-scale sub-band Constant-Q Transform discriminator (MS-SB-CQTD), and multi-scale STFT discriminator (MS-STFTD). For adversarial feedback, we use least-squares GAN (LS-GAN) loss $\mathcal{L}_{adv}$ and feature matching loss $\mathcal{L}_{fm}$ together.

2.6 Training Objective

Total Loss for CFM pre-training The total loss for pre-training can be expressed as follows:

$$\mathcal{L}_{pretrain} = \mathcal{L}_{CFM} + \lambda_{semantic}\mathcal{L}_{semantic} + \lambda_{acoustic}\mathcal{L}_{acoustic} \quad (4)$$

Table 1: Objective evaluation results using LibriTTS Benchmark Dataset. Note that some models have a target sampling rate of 16 kHz, which is a significant lower sampling rate compared to 24 kHz (24 kHz models have $\times 1.5$ higher compression rate than 16 kHz models). StableCodec do not release streaming models so we use a parallel implementation trained with CTC loss.

Method	Hz	TPS	N_q	Stream.	CER (↓)	WER (↓)	M-STFT (↓)	PESQ (↑)	Period. (↓)	V/UV (↑)	Pitch (↓)	UTMOS (↑)	MOS (↑)
GT	24k	-	-	-	1.12	3.06	-	-	-	-	-	3.862	3.72±0.02
SpeechTokenizer (Zhang et al., 2024)	16k	400	8	✓	2.06	4.46	-	2.468	0.129	0.934	41.631	3.585	3.64±0.02
SpeechTokenizer (Zhang et al., 2024)	16k	50	1	✓	6.71	12.65	-	1.118	0.412	0.616	1565.082	1.259	-
BigCodec (Xin et al., 2024)	16k	80	1	✗	2.61	5.44	-	2.607	0.145	0.927	33.757	3.889	3.66±0.02
X-codec2 (Ye et al., 2025)	16k	50	1	✗	2.45	5.20	-	2.259	0.230	0.857	53.052	3.869	3.68±0.02
StableCodec (Parker et al., 2025)	16k	50	2	✗	6.85	12.42	-	2.072	0.184	0.899	32.002	4.134	3.55±0.03
StableCodec (Parker et al., 2025)	16k	25	1	✗	8.62	16.19	-	1.893	0.194	0.892	48.680	4.131	3.58±0.03
EnCodec (Défossez et al., 2023)	24k	600	8	✓	1.19	3.55	1.163	2.771	0.113	0.941	32.147	2.969	3.30±0.03
DAC (Kumar et al., 2023)	24k	600	8	✗	1.09	2.91	1.012	3.505	0.075	0.962	22.290	3.546	3.76±0.02
DAC (Kumar et al., 2023)	24k	100	2	✗	9.74	16.21	1.386	1.632	0.190	0.888	81.032	2.050	-
Mimi (Défossez et al., 2024)	24k	100	8	✓	3.07	6.91	1.352	2.266	0.165	0.910	50.686	3.506	3.74±0.02
Mimi (Défossez et al., 2024)	24k	50	4	✓	7.42	12.72	1.552	1.657	0.210	0.880	77.575	3.019	-
Mimi (Défossez et al., 2024)	24k	25	2	✓	12.60	21.41	1.916	1.239	0.272	0.824	234.489	2.413	-
WavTokenizer (Ji et al., 2025)	24k	75	1	✗	4.87	9.42	1.220	2.247	0.157	0.918	41.123	3.821	3.75±0.03
WavTokenizer (Ji et al., 2025)	24k	40	1	✗	13.83	23.53	1.627	1.604	0.196	0.891	69.903	3.467	3.44±0.03
FlowTokenizer w/o Prompt	24k	25	1	✓	2.57	5.87	1.621	1.556	0.178	0.907	69.368	4.000	3.80±0.02
FlowTokenizer w/ 3s Prompt	24k	25	1	✓	1.86	4.68	1.482	1.643	0.171	0.908	77.421	3.937	-

where $\lambda_{semantic}$ and $\lambda_{acoustic}$ are set to 0.05 and 0.005, respectively. The detailed training pipeline is illustrated in Appendix A.

Total Loss for Adversarial fine-tuning The total loss for fine-tuning can be expressed as follows:

$$\mathcal{L}_{finetune} = \mathcal{L}_{adv} + \lambda_{fm}\mathcal{L}_{fm} + \lambda_{mel}\mathcal{L}_{mel} + \lambda_{semantic}\mathcal{L}_{semantic} \quad (5)$$

where λ_{fm}, λ_{mel} and $\lambda_{semantic}$ are set to 2, 15, and 100, respectively. We utilize the Euler method, and use [0, 0.25, 0.5, 0.75] for fixed time t. All parameters are jointly trained with $\mathcal{L}_{finetune}$.

3 Experiment and Result

3.1 Experimental Setting

Dataset For reproducible experiment in low-resource settings, we use an open-source, small-scale dataset, LibriTTS (Zen et al., 2019), a high-quality multi-speaker dataset with a sampling rate of 24 kHz. We use all train subsets including *train-clean-100*, *train-clean-360*, and *train-other-500*, which consists of approximately 500 hours of speech. Raw waveforms are used directly as an input for all experiments.

Training We train FlowTokenizer using AdamW optimizer with a learning rate of 2×10^{-4}, batch size of 256 for 1M steps on eight NVIDIA H100 GPUs. For adversarial fine-tuning, we use a learning rate of 2×10^{-5}, batch size of 128 for 0.1M steps on eight NVIDIA H100 GPUs. For efficient training, we use the segments of 96,000 frames (4s) and use mixed precision training. For acoustic REPA and adversarial fine-tuning, the generated waveform is segmented into 65,536 before fed to STFT function and discriminators. It only takes 4 days to train the model (3 days for pre-training and 1 days for adversarial fine-tuning). Note that CFM pre-training can decrease entire GAN training which was demonstrated in (Lee et al., 2024).

3.2 LibriTTS benchmark: Multi-speaker Dataset with 24,000 Hz

We compared the model with various neural audio codecs including EnCodec, DAC, Mimi, and WavTokenizer which are trained for high-resolution waveform with sampling rate of 24 kHz. We compare the model with SpeechTokenizer, BigCodec, X-codec2, and StableCodec. Note that these models are trained with sampling rate of 16 kHz so it is much easier than 24 kHz reproduction.

The results show that our model preserves semantic information using only 25 token per second in terms of CER and WER. Furthermore, FlowTokenizer outperformed other low-bitrate models including Mimi and WavTokenizer in terms of perceptual quality and semantic consistency. It is worth noting that our model can decode the tokens in a streaming manner, while current low-bitrate models encode and decode in parallel by using convolutional layer with large receptive fields. Also, our model achieve better performance in subjective evaluation. We used the target sampling rate of each model for subjective evaluation to demonstrate the importance of high-resolution waveform

Table 2: Objective evaluation results using Librispeech benchmark dataset. For reference, we cite the reported performance of models† from X-codec2 (Ye et al., 2025), and use the same evaluation methods used in X-codec2.

Method	Hz	TPS	N_q	Bitrate	Stream.	WER (↓)	STOI (↑)	PESQ -WB (↑)	PESQ -NB (↑)	SPK-SIM (↑)	UTMOS (↑)
GT	16k	-	-	-	-	1.96	1.00	4.64	4.55	1.00	4.09
SpeechTokenizer† (Zhang et al., 2024)	16k	100	2	1000	✓	3.92	0.77	1.25	1.59	0.36	2.28
SpeechTokenizer† (Zhang et al., 2024)	16k	50	1	500	✓	5.01	0.64	1.14	1.30	0.17	1.27
BigCodec† (Xin et al., 2024)	16k	80	1	1040	✗	2.76	0.93	2.68	3.27	0.84	4.11
X-codec2† (Ye et al., 2025)	16k	50	1	800	✗	2.47	0.92	2.43	3.04	0.82	4.13
StableCodec† (Parker et al., 2025)	16k	50	2	700	✗	5.12	0.91	2.24	2.91	0.62	4.23
EnCodec† (Défossez et al., 2023)	24k	600	8	6000	✓	2.15	0.94	2.77	3.18	0.89	3.09
EnCodec† (Défossez et al., 2023)	24k	150	2	1500	✓	4.90	0.85	1.56	1.94	0.60	1.58
WavTokenizer†	24k	75	1	900	✗	3.98	0.90	2.13	2.63	0.65	3.79
WavTokenizer†	24k	40	1	480	✗	11.20	0.85	1.62	2.06	0.48	3.57
Mimi† (Défossez et al., 2024)	24k	100	8	1100	✓	2.96	0.90	2.25	2.80	0.73	3.56
Mimi (Défossez et al., 2024)	24k	100	8	1100	✓	2.92	0.90	2.27	2.80	0.73	3.63
Mimi† (Défossez et al., 2024)	24k	50	4	550	✓	4.89	0.85	1.64	2.09	0.50	3.03
Mimi (Défossez et al., 2024)	24k	50	4	550	✓	4.84	0.85	1.65	2.09	0.50	3.10
Mimi (Défossez et al., 2024)	24k	25	2	225	✓	8.35	0.76	1.26	1.52	0.27	2.51
FlowTokenizer w/o Prompt	24k	25	1	400	✓	3.38	0.86	1.56	1.94	0.41	4.09
FlowTokenizer w/ 3s Prompt	24k	25	1	400	✓	2.91	0.86	1.61	2.05	0.60	4.01

Table 3: Ablation Study

Method	CER (↓)	WER (↓)	M-STFT (↓)	PESQ (↑)	Period. (↓)	V/UV (↑)	Pitch (↓)	UTMOS (↑)
FlowTokenizer (REPA-H)	2.57	5.87	1.621	1.556	0.178	0.907	69.368	4.000
Ablation: REPA								
w/ Semantic REPA (CT)	6.54	11.87	1.656	1.394	0.185	0.903	97.133	3.970
w/ Semantic REPA (DiT)	8.09	15.17	1.671	1.671	0.174	0.908	65.809	4.151
w/o REPA	17.59	29.49	1.505	1.647	0.198	0.892	104.176	3.724
Ablation: VQ								
w/o DVQ (Only FSQ)	3.65	7.35	1.595	1.551	0.180	0.905	75.780	3.900
Ablation: Adversarial Tuning								
w/o Adversarial Fine-tuning	3.00	6.00	2.137	1.390	0.186	0.903	92.410	3.535
Ablation: One-step GAN								
Only Adversarial Training (1M steps)	10.93	19.30	1.576	1.688	0.177	0.906	55.864	4.132

over audible frequency. Details of the baseline, evaluation, and synthesis speed are described in Appendices C to E.

3.3 LibriSpeech benchmark: Multi-speaker Dataset with 16,000 Hz

Following (Ye et al., 2025), we also evaluate each model with the *test-clean* subset of LibriSpeech (Panayotov et al., 2015) consisting of 2,620 samples. By using only 25 TPS, our model achieve better semantic consistency in terms of WER compared to 24 kHz models under low-bitrate scenarios (<1,000 bps), and better perceptual quality compared to all 24 kHz models. Note that it is difficult to compare the model with 16 kHz due to different target sampling rate. While we focused on semantic consistency under low-bitrate scenarios in this work, however, we observed that our model has lower speaker similarity in terms of SPK-SIM using embeddings extracted by WavLM-based speaker verification models. We can discuss our models are trained with LibriTTS, which is a small-scale dataset so our model is limited to generate the unseen speaker details, and WavTokenizer (Ji et al., 2025) trained with the same LibriTTS dataset also show similar tendency under low-bitrate condition.

3.4 Ablation Study

As indicated in Table 3, we conducted ablation studies including REPA, semantic decoder structure, VQ, adversarial fine-tuning, and some training optimization methods.

REPA Vanilla DiT models could not guarantee the semantic consistency of reconstructed speech under low-bitrate scenarios. Adding semantic REPA significantly improve the semantic consistency of models in terms of ASR evaluations, and also improve the perceptual quality. Additionally, we compared DiT-based semantic distillation, and causal Transformers based semantic prediction structures. We found that there is a trade-off between semantic and acoustic reproduction qual-

ity. REPA-H could enhance both semantic and acoustic quality by aligning semantic and acoustic representations hierarchically. Also, disentangled alignment training improves the robustness and performance during training.

VQ We found that the basic FSQ and DVQ showed similar performance, but DVQ achieved better semantic consistency. Additionally, we can extract auxiliary semantic tokens when using DVQ.

Adversarial Fine-tuning While training CFM with REPA could align the representation efficiently compared to GAN-based models, however, it requires many sampling steps so we fine-tuned the model with adversarial feedback by fixing the sampling steps of four, and this improves entire performance with fewer sampling steps.

One-step GAN model We compare the performance of GAN with the same structure of ours using the same discriminator, Mel losses on the generated waveform, and semantic REPA loss with 1M steps (7 days). Due to long training, it achieve comparable acoustic performance but it do not preserve semantic information, resulting in high CER and WER. This result demonstrates the efficiency of CFM-based pre-training for GAN-based models.

Table 4: ASR Results using different tokens.

Input	CER (↓)	WER (↓)
Continuous SSL (MMS)	1.79	4.44
Acoustic VQ	18.27	30.81
Semantic VQ	2.98	6.77
DVQ	2.46	5.54

3.5 Semantic Consistency

We conduct automatic speech recognition (ASR) on different representations to evaluate the semantic consistency of each representation including GT MMS representation (Target continuous SSL features), acoustic representation ([4, 4, 4, 4]) from DVQ, semantic representation ([4, 4, 4, 4]) from DVQ, and factorized representation [4, 4, 4, 4, 4, 4, 4, 4] from DVQ. The results show that DVQ successfully factorizes semantic and acoustic representation with a single quantizer. This indicates that our semantic tokens can be used for auxiliary semantic token prediction. We will plan to train the LLM-based speech synthesis models using our tokens, and we will train the model with DVQ token classification loss and auxiliary semantic token classification loss for better semantic guidance.

Table 5: TTS Results using different tokens.

Model	TPS	WER (↓)	SIM (↑)
GT (Seed-en)	-	2.14	0.734
Llasa-1B (X-Codec2)	50	3.22	0.572
Qwen2.5-0.5B (CosyVoice2)	25	2.57	0.652
Qwen2.5-0.5B (FlowTokenizer)	25	2.29	0.676
+ Auxiliary Semantic Token Loss	25	1.98	0.676
+ Tokenizer CFG	25	1.92	0.687

3.6 LLM-based Text-to-Speech

To demonstrate that the proposed FlowTokenizer can be effectively integrated into TTS system, we train an LLM-based TTS model using discrete speech tokens. Specifically, we adopt Qwen2.5-0.5B (Yang et al., 2024) as the backbone LM. We utilize Emilia-en and HiFiTTS-2 (Langman et al., 2025) dataset for TTS model. To improve training stability and semantic alignment between text and speech token, we introduce an auxiliary semantic token loss, where an additional prediction head predicts a semantic token based on DVQ. This auxiliary objective simply encourages the LLM to learn representations that are simultaneously discriminative at the fine-grained token level and consistent at the semantic level. As shown in Table 5, the auxiliary semantic loss reduces WER, and further improves both WER and speaker similarity when combined with tokenizer-level classifier-free guidance (CFG). Specifically, we use CFG of 0.3 by dropping only prompts.

4 Conclusion

We introduced FlowTokenizer, a novel CFM-based neural audio codec tailored for extremely low-bitrate scenarios. To further enhance learning efficiency, we introduce REPA-H, a hierarchical representation alignment learning method, which can align semantic and acoustic representation hierarchically within DiT layers. In addition, we propose DVQ, a dense vector quantization method which can factorize semantic and acoustic representation at a single layer representation. Through extensive experiments, we demonstrate that FlowTokenizer outperforms state-of-the-art neural audio codecs under low-bitrate scenarios in terms of audio quality and semantic intelligibility. These results validate the effectiveness of CFM for high-resolution waveform generation and position FlowTokenizer as a promising alternative to GAN-based models for future research in efficient and robust waveform generation.

Limitation

While we first introduce a CFM-based neural audio codec, there are still large room for improvement. Because we only train the model with small-scale speech dataset of 0.5k hours, our model has lower speaker consistency while preserving semantic consistency. We will train the model with large-scale universal dataset including speech, vocal, music, and sound effect. We see that it will increase the reproduction capability of quantized representation, and significantly improve the generative capability of DiT-based decoder similar to voicebox and F5-TTS style speech models. However, in this study, we investigate CFM-based neural audio codec training methods, and demonstrate the effectiveness of the proposed method including REPA-H, DVQ, and possibility of end-to-end neural audio codec training via CFM. Furthermore, we see that we could further optimize the alignment (distillation) method. To do this, we will explore different losses including adversarial distillation methods for detailed semantic and acoustic alignment during pretraining. Also, we will further investigate the DiT structure by incorporating Unet-style long skip connections to improve the efficiency and robustness for waveform modeling.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02283048, Developing the Next-Generation General AI with Reliability, Ethics, and Adaptability, IITP-2026-RS-2023-00255968, the Artificial Intelligence Convergence Innovation Human Resources Development, RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale), National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (RS-2025-16069227 and RS-2024-00356486).

References

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2021. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. 2025. F5-tts: A fairytaler 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.

Ha-Yeong Choi and Sang-Hoon Lee. 2025. Streaming audio generation from discrete tokens via streaming flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

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*.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, and 1 others. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 682–689. IEEE.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Robert Gray. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. 2023. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR.

Rongjie Huang, Max W. Y. Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. 2022. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. In *Proceedings of the Thirty-First*

International Joint Conference on Artificial Intelligence, IJCAI-22, pages 4157–4163. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, and Zhou Zhao. 2025. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *The Thirteenth International Conference on Learning Representations*.

Yidi Jiang, Qian Chen, Shengpeng Ji, Yu Xi, Wen Wang, Chong Zhang, Xianghu Yue, ShiLiang Zhang, and Haizhou Li. 2025a. Unicodec: Unified audio codec with single domain-adaptive codebook. *arXiv preprint arXiv:2502.20067*.

Ziyue Jiang, Yi Ren, Ruiqi Li, Shengpeng Ji, Boyang Zhang, Zhenhui Ye, Chen Zhang, Bai Jionghao, Xiaoda Yang, Jialong Zuo, and 1 others. 2025b. Megatts 3: Sparse alignment enhanced latent diffusion transformer for zero-shot speech synthesis. *arXiv preprint arXiv:2502.18924*.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiangyang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and sheng zhao. 2024. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Forty-first International Conference on Machine Learning*.

Sungwon Kim, Kevin Shih, Joao Felipe Santos, Evelina Bakhturina, Mikyas Desta, Rafael Valle, Sungroh Yoon, Bryan Catanzaro, and 1 others. 2024. P-flow: A fast and data-efficient zero-shot tts through speech prompting. *Advances in Neural Information Processing Systems*, 36.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2023. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993.

Ryan Langman, Xuesong Yang, Paarth Neekhara, Shehzeen Hussain, Edresson Casanova, Evelina Bakhturina, and Jason Li. 2025. Hifitts-2: A large-scale high bandwidth speech dataset. *arXiv preprint arXiv:2506.04152*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and 1 others. 2024. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36.

Keon Lee, Dong Won Kim, Jaehyeon Kim, Seungjun Chung, and Jaewoong Cho. 2025a. DiTTo-TTS: Diffusion transformers for scalable text-to-speech without domain-specific factors. In *The Thirteenth International Conference on Learning Representations*.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. 2022. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations*.

Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. 2023. BigVGAN: A universal neural vocoder with large-scale training. In *The Eleventh International Conference on Learning Representations*.

Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. 2024. Accelerating high-fidelity waveform generation via adversarial flow matching optimization. *arXiv preprint arXiv:2408.08019*.

Sang-Hoon Lee, Ha-Yeong Choi, and Seong-Whan Lee. 2025b. Periodwave: Multi-period flow matching for high-fidelity waveform generation. In *The Thirteenth International Conference on Learning Representations*.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2024. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*, 36.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Peng Liu, Dongyang Dai, and Zhiyong Wu. 2025. RFWave: Multi-band rectified flow for audio waveform reconstruction. In *The Thirteenth International Conference on Learning Representations*.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron Courville, and Yoshua Bengio. 2022. Chunked autoregressive GAN for conditional waveform synthesis. In *International Conference on Learning Representations*.

Takuma Okamoto, Haruki Yamashita, Yamato Ohtani, Tomoki Toda, and Hisashi Kawai. 2023. Wavenext: Convnext-based fast neural vocoder without istft layer. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

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

Julian D Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. 2025. Scaling transformers for low-bitrate high-quality speech coding. In *The Thirteenth International Conference on Learning Representations*.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Robin San Roman, Yossi Adi, Antoine Deleforge, Romain Serizel, Gabriel Synnaeve, and Alexandre Défossez. 2023. From discrete tokens to high-fidelity audio using multi-band diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Christian J Steinmetz and Joshua D Reiss. 2020. auraloss: Audio focused loss functions in pytorch. In *Digital music research network one-day workshop (DMRN+ 15)*.

Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302.

Simon Welker, Matthew Le, Ricky T. Q. Chen, Wei-Ning Hsu, Timo Gerkmann, Alexander Richard, and YI-CHIAO WU. 2025. Flowdec: A flow-based full-band general audio codec with high perceptual quality. In *The Thirteenth International Conference on Learning Representations*.

Haibin Wu, Naoyuki Kanda, Sefik Emre Eskimez, and Jinyu Li. 2024. Ts3-codec: Transformer-based simple streaming single codec. *arXiv preprint arXiv:2411.18803*.

Detai Xin, Xu Tan, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2024. Bigcodec: Pushing the limits of low-bitrate neural speech codec. *arXiv preprint arXiv:2409.05377*.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report, 2024. *URL https://arxiv.org/abs/2407.10671*, 7:8.

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

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023b. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi DAI, and 1 others. 2025. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. 2025. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*.

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. In *Proc. Interspeech 2019*, pages 1526–1530.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*.

Table 6: Hyperparameters of FlowTokenizer

Module	Hyperparameter	FlowTokenizer
Input Linear Reshape	Input waveform	480
	Linear1 (No Bias)	[480, 1024]
	Linear2	[1024, 1024]
Acoustic Encoder (Causal Transformer)	Input Dim.	1024
	Hidden Dim.	4096
	Layer	6
	Head	16
Semantic Encoder (MMS)	Hidden Dim.	1024
	Layer	7
VQ	Token Hz	25 Hz
	Frame per token	960
	Token initial Dim.	2048
	Causal Convolution1	[2048, 2048], stride=2, group=128
	Causal Convolution2	[2048, 8], stride=1, group=2
	Method	DVQ
	Causal Convolution3	[8, 2048], stride=1, group=2
	Causal Transposed Convolution	[2048,2048], stride=2, group=128
x_t&Prompt Linear Reshape	Input waveform	480
	Linear1 (No Bias)	[480, 1024]
	Linear2	[1024, 1024]
Output Linear Reshape	Linear1	[1024,1024]
	Linear2 (No Bias)	[1024,480]
	Vector Fields	480
Semantic Decoder (Causal Transformer)	Input Linear.	[2048,1024]
	Input Dim.	1024
	Hidden Dim.	4096
	Layer	6
	Head	16
CFM Time	Time Embedding	256
	Linear1	[256, 1024]
	Activation	SiLU
	Linear2	[1024, 1024]
Acoustic DiT	Input Linear.	[3072,1024]
	Input Dim.	1024
	Hidden Dim.	4096
	Layer	6
	Head	16
DiT	Input Linear	[4096,1024]
	Input Dim.	1024
	Hidden Dim.	4096
	Layer	12
	Head	16
Pre-train	Training Step	1M
	Learning Rate	2×10^{-4}
	Learning Scheduling	Warm-up (3,000 steps)
	Batch Size	256
	Noise Scale	0.25
	Segment Size	96000
	Limited Context Attention Window	96000
	Audio/Token Drop	0.3/0.2
Fine-tuning	Training Step	0.1M
	Learning Rate	2×10^{-5}
	Learning Scheduling	-
	Batch Size	128
	Segment Size	96000
	Limited Context Attention Window	96000
	Audio/Token Drop	0.3/0.2

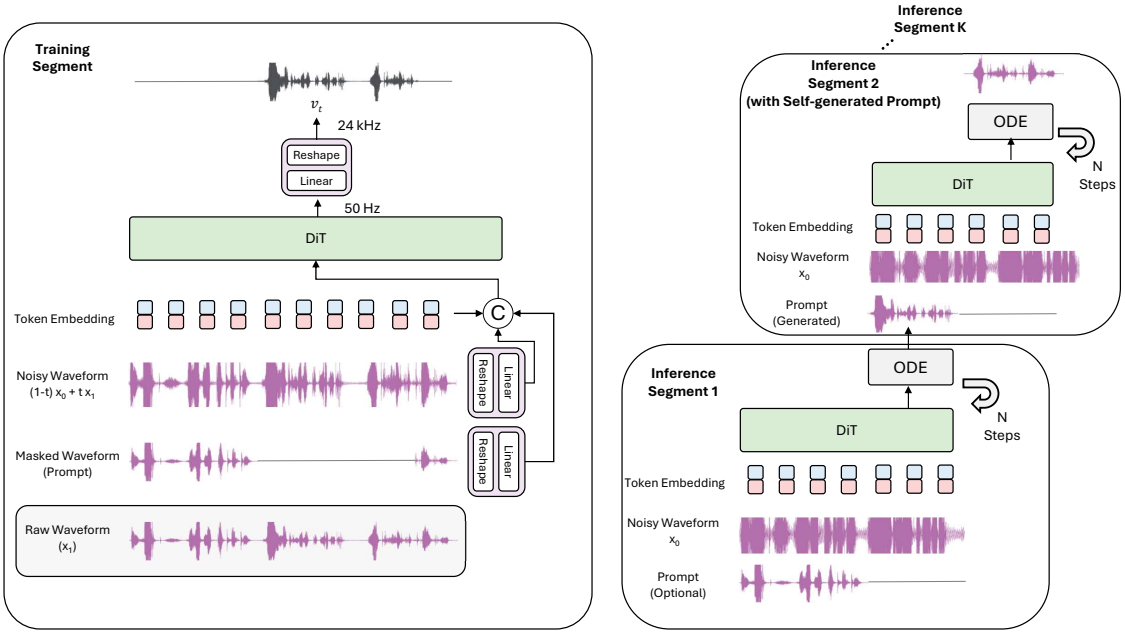

Figure 4: Training and inference details. We train the model with a slicing window of 96,000 frames (4 seconds) with randomly masked prompts. During inference, the model can generate waveforms in a streaming manner using self-generated prompt.

Table 7: Streaming generation with self-generated prompt

Method	WER (↓)	STOI (↑)	PESQ -WB (↑)	PESQ -NB (↑)	SPK-SIM (↑)	UTMOS (↑)
FlowTokenizer (Parallel Gen.)	3.38	0.86	1.56	1.94	0.41	4.09
FlowTokenizer (Streaming Gen.)	3.50	0.86	1.54	1.93	0.43	4.10
FlowTokenizer w/o Self-generated Prompt (Streaming Gen.)	3.54	0.85	1.51	1.89	0.43	4.08
FlowTokenizer w/ Target Prompt (Streaming Gen.)	2.91	0.86	1.62	2.05	0.60	4.02

A Implementation Details

We describe the hyperparameter details in Table 6. Figure 6 illustrates the training and inference pipeline of FlowTokenizer for CFM training with REPA-H.

A.1 Streaming Generation with Self-generated Prompt

Our model is trained with waveform-level prompts by masking the target x_1. Therefore, the model can generate waveforms conditioned on a prompt waveform. In this work, we leverage waveform-level prompting and utilize self-generated prompts for streaming generation, as illustrated in Figure 4.

Table 7 demonstrates the robustness of our model in streaming generation, achieving comparable performance to parallel generation while using only a small context window thanks to causal encoding and carefully designed streaming decoding structure. Furthermore, with self-generated prompts, our streaming generation achieves nearly the same performance as parallel generation, and even outperforms it in SPK-SIM and UTMOS metrics. We also found that our model could generate any size of context windows due to the limited context attention and relative positional embedding. We follow the implementation of Mimi for limited context attention window in Transformer Networks for real-time streaming.

Table 8: RVQ-based model comparison. All models use eight RVQ layers.

Model	WER	STOI	PESQ	SIM	UTMOS
Mimi	6.91	0.90	2.26	0.70	3.50
FlowTokenizer (w/o Prompt)	3.38	0.92	2.46	0.64	3.99
FlowTokenizer (w/ 1s Prompt)	3.25	0.93	2.69	0.80	4.00
FlowTokenizer (w/ 2s Prompt)	3.08	0.94	2.74	0.82	3.99
FlowTokenizer (w/ 3s Prompt)	3.40	0.94	2.74	0.83	3.98

B RVQ-based FlowTokenizer

We trained a higher-bitrate version using 8 RVQ with reduced codebook size (6,561, [3,3,3,3,3,3,3,3]), same encoder/decoder architectures, and LibriTTS dataset. We only train for 0.5M steps (1.5 d), and fine-tune for 50,000 steps (9 hours). However, table 8 highlights the effectiveness of FlowTokenizer.

Table 9: Synthesis Speed for each model.

Method	Params. (M)	Hz	TPS	N_q	Stream.	xRT(↑)
SpeechTokenizer (Zhang et al., 2024)	103M	16k	400	8	✓	×76.899
BigCodec (Xin et al., 2024)	159M	16k	80	1	✗	×21.472
X-codec2 (Ye et al., 2025)	822M	16k	50	1	✗	×14.026
StableCodec (Parker et al., 2025)	953M	16k	50	2	✗	×63.456
EnCodec (Défossez et al., 2023)	15M	24k	600	8	✓	×64.267
DAC (Kumar et al., 2023)	74M	24k	600	8	✗	×60.237
Mimi (Défossez et al., 2024)	79M	24k	100	8	✓	×45.538
WavTokenizer (Ji et al., 2025)	80M	24k	75	1	✗	×29.982
WavTokenizer (Ji et al., 2025)	80M	24k	40	1	✗	×32.494
FlowTokenizer	578M	24k	25	1	✓	×44.096

C Synthesis Speed

We calculated the synthesis speed (xRT) with an NVIDIA A100 GPU and reported parameter size of each model. Table 9 shows that our model has competitive synthesis speed even with four iterative generation steps, thanks to low-resolution feature modeling with linear-reshape transformation. Our model operates only on 50 Hz features during the forward pass, unlike other models using waveform-level feature modeling. Meanwhile, StableCodec utilized FlashAttention to accelerate the attention modules. Following this approach, we plan to incorporate FlashAttention to further improve the sampling speed.

D Baseline Details

D.1 Neural audio codec operating at 16 kHz

SpeechTokenizer We utilized the official implementation and checkpoint of SpeechTokenizer.[2] SpeechTokenizer distills self-supervised representation into the first RVQ token.

BigCodec We used the official implementation and checkpoint of BigCodec.[3] BigCodec scales up the model to achieve improved performance with a single quantizer under low-bitrate scenarios.

X-codec2 We follow the official implementation.[4] X-codec2 concatenated semantic representation (Wav2Vec2-BERT) with acoustic representation before quantization to preserve the semantic information for low-bitrate neural audio codec.

StableCodec StableCodec adopts Transformers, and scaled up the model for low-bitrate neural audio codec modeling. We utilized the official implementation and checkpoint of StableCodec using additional CTC loss to distill the phonetic information.[5]

D.2 Neural audio codec operating at 24 kHz

Encodec We use the official implementation and checkpoint of EnCodec.[6] EnCodec introduces a fully causal structure for streaming modeling.

DAC We utilized the official implementation and checkpoint of DAC.[7] DAC consists of non-causal layers for effective neural audio codec modeling.

Mimi We follow the official implementation and checkpoint of Mimi.[8] Mimi utilized split RVQ modeling for effective semantic distillation, and used fully-causal layers for streaming modeling.

WavTokenizer We use the official implementation and checkpoints of WavTokenizer with at 40 TPS and 75 TPS.[9] WavTokenizer introduced efficient low-bitrate neural audio codecs with strong reconstruction.

E Evaluation Details

E.1 Objective Evaluation

Following (Lee et al., 2025b), we adopt six objective metrics: multi-resolution STFT (M-STFT), perceptual evaluation of speech quality (PESQ), Periodicity error (Period.), F1 score of voice/unvoice

[2] https://github.com/ZhangXInFD/SpeechTokenizer
[3] https://github.com/Aria-K-Alethia/BigCodec
[4] https://github.com/zhenye234/X-Codec-2.0
[5] https://github.com/Stability-AI/stable-codec
[6] https://github.com/facebookresearch/encodec
[7] https://github.com/descriptinc/descript-audio-codec
[8] https://github.com/kyutai-labs/moshi
[9] https://github.com/jishengpeng/WavTokenizer

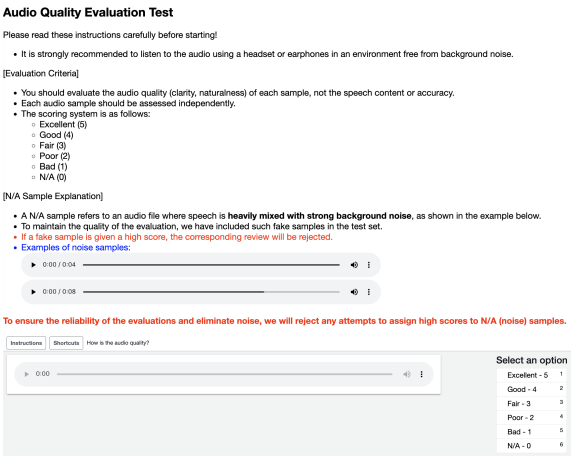

Figure 5: Detailed information on listeners restrictions and task completion interfaces.

classification (V/UV F1), Pitch distance, and UTMOS.

M-STFT we use the multi-resolution STFT loss (M-STFT) from the Auraloss library (Steinmetz and Reiss, 2020).[10] Originally proposed in Parallel WaveGAN (Yamamoto et al., 2020), this metrics measure the spectral difference between ground-truth and generated waveforms across multiple STFT resolutions.

PESQ We evaluate the wideband version of perceptual evaluation of speech quality[11], which is a standardized measure for speech reproduction assessment. All audio is downsampled to 16 kHz prior to calculation.

Periodicity, V/UV F1, and Pitch Following CarGAN evaluation methods (Morrison et al., 2022), we utilized a Periodicity RMSE to measure the periodicity error which perceptually degrades the audio.[12]. Additionally, we evaluate Voice/Unvoice F1 score, and measure pitch distance to access pitch accuracy.

UTMOS We assess naturalness using UTMOS, a pre-trained MOS prediction model.[13]. UTMOS provides a reliable approximation for subjective MOS.

E.2 Subjective Evaluation

MOS We conducted an audio quality evaluation using Amazon Mechanical Turk for crowdsourcing MOS test. The MOS was rated on a 5-point scale. To reduce noise in the crowdsourcing results, we introduced randomly inserted Gaussian noise samples, which were mapped to "N/A". Any response that incorrectly selected a score for these noise samples were excluded from the final analysis. To ensure fairness and consistency, we used all 208 samples that were also employed for objective metric evaluations. Each audio sample was rated by 20 unique workers. Each evaluation task was compensated at $1.00 per task. The user interface used for the evaluation is illustrated in Figure 5.

F Potential Broader Impact

Practical Application In this work, we present low-bitrate neural audio codec which can be utilized for speech large language models as an efficient speech tokenizer. This simply reduces the sequence lengths for language models, resulting in efficient training and inference of Transformer networks. Also, we have thoroughly explored representation learning of DiT structure for neural audio codec, and our structure could enhance the training efficiency. Then, the proposed DVQ could factorize multiple attributes at the single layer representation, giving auxiliary semantic tokens. Furthermore, our model can be used for VAE-based waveform autoencoder and its applications such as LDM.

Social Negative Impact Although neural audio codecs are not directly employed in applications like text-to-speech or voice conversion, high-quality speech synthesis models built upon them could be exploited to impersonate individuals and disseminate misleading content. Audio watermarking and fake audio detection models could mitigate these issues.

[10] https://github.com/csteinmetz1/auraloss
[11] https://github.com/ludlows/PESQ
[12] https://github.com/descriptinc/cargan
[13] https://github.com/tarepan/SpeechMOS

Table 10: Frequency-wise Mel-spectrogram L1 distance

Model	0-12kHz	12-18kHz	18-24kHz
FlowTokenizer (No REPA)	1.32	1.61	1.75
FlowTokenizer (REPA)	1.24	1.39	1.68
FlowTokenizer (REPA-H)	0.90	1.10	1.17

G Frequency (Band)-wise Comparison

We measured the frequency-wise (band-wise) Mel-spectrogram L1 distance of our pre-trained models including FlowTokenizer (No REPA), FlowTokenizer (REPA), FlowTokenizer (REPA-H) in Table 10. While REPA improves the convergence speed and all band-wise metrics, the model still fails to effectively reconstruct high-frequency components of the waveform. Introducing REPA-H could enhance the acoustic capacity in representation, resulting better acoustic reproduction.

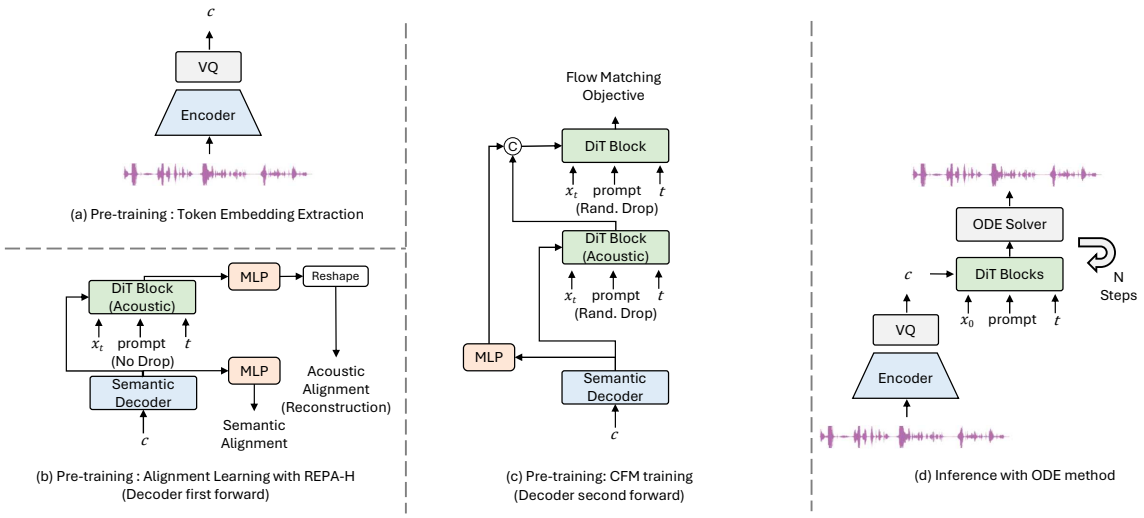

Figure 6: Training and inference pipeline for CFM pre-training with REPA-H

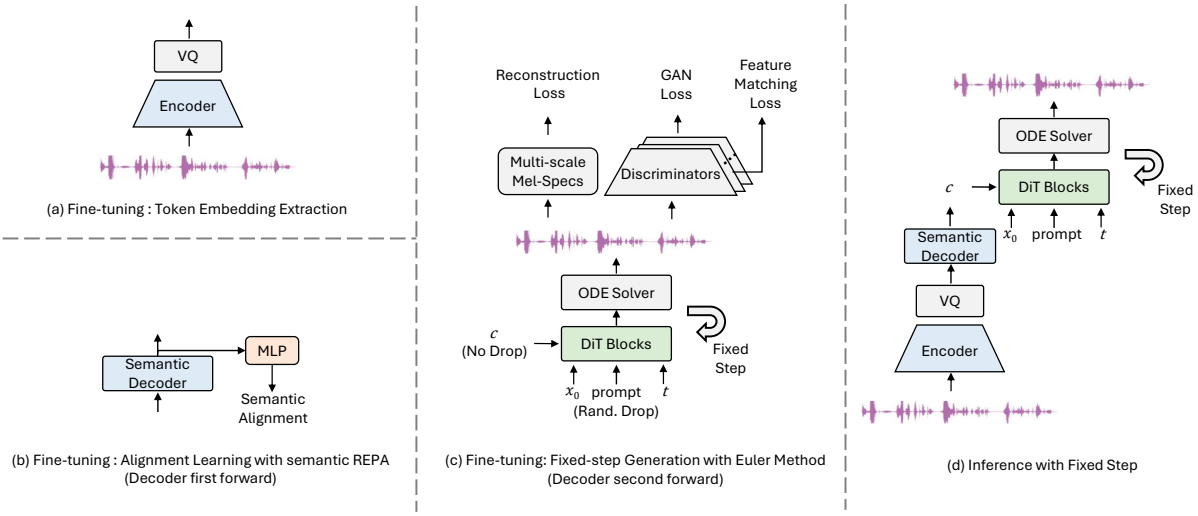

Figure 7: Training and inference pipeline for adversarial fine-tuning with semantic REPA