# XY-Tokenizer: Mitigating the Semantic-Acoustic Conflict in Low-Bitrate Speech Codecs

**Yitian Gong**[1,2,3] **Luozhijie Jin**[1,2] **Kuangwei Chen**[1,2,3]
**Dong Zhang**[1] **Ruifan Deng**[1] **Xiaogui Yang**[3] **Xin Zhang**[1]
**Zhaoye Fei**[1,3] **Qinyuan Cheng**[1,3] **Shimin Li**[1,3] **Xipeng Qiu**[1,2,3]*
{ytgong24, chengqy21}@m.fudan.edu.cn xpqiu@fudan.edu.cn

[1]Fudan University [2]Shanghai Innovation Institute
[3]MOSI Intelligence

## Abstract

Speech codecs provide an important interface between continuous speech signals and large language models. An ideal codec for speech language models should not only preserve acoustic information but also capture rich semantic information. However, existing codecs struggle to balance these objectives at low bitrates. We propose **XY-Tokenizer**, a low-bitrate speech codec (around 1 kbps) trained with a structured multi-stage, multi-task strategy that aligns discrete speech representations with text while preserving fine-grained acoustic details for reconstruction. This design explicitly mitigates the semantic–acoustic conflict observed in prior low-bitrate codecs. Experiments show that XY-Tokenizer achieves stronger semantic alignment than representative semantic-distillation codecs such as SpeechTokenizer and Mimi, while maintaining high-quality speech reconstruction across both clean and out-of-distribution conditions. Furthermore, XY-Tokenizer consistently outperforms existing low-bitrate codecs in LLM-based speech understanding and generation tasks, demonstrating its effectiveness as a general-purpose speech representation for speech–language modeling.

## 1 Introduction

In recent years, large language models (LLMs) (Achiam et al., 2023; Dubey et al., 2024; Yang et al., 2024a; Guo et al., 2025; Zeng et al., 2025) have achieved significant advancements in natural language processing, enabling fluent and natural text-based interactions. Building on this progress, speech large language models have attracted increasing attention (Zhang et al., 2023a; Chu et al., 2024; Défossez et al., 2024; Zeng et al., 2024; Nguyen et al., 2025; Huang et al., 2025). A critical component of Speech LLMs is the speech codec (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2023; Zhang et al., 2023b; Défossez et al., 2024), which transforms continuous speech signals into discrete tokens to interface with token-based LLMs.

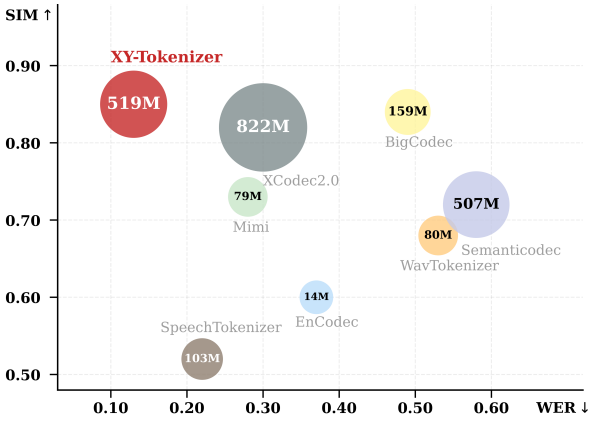


Figure 1: Semantic–acoustic comparison of speech codecs. WER from ASR probing task mentioned in Section 3.2 is shown on the x-axis (lower is better), and reconstruction quality on the y-axis (higher is better). Bubble size indicates number of parameters. XY-Tokenizer achieves a strong balance between semantic alignment and acoustic quality at around 1 kbps.

Speech tokens are commonly categorized into semantic tokens and acoustic tokens (Borsos et al., 2023). Semantic tokens, typically derived from discretized self-supervised or ASR-based speech representations, provide compact and structured units that are well-suited for sequence modeling in speech LLMs, but are less effective for high-fidelity speech reconstruction (Baevski et al., 2020; Hsu et al., 2021; Chung et al., 2021; Chen et al., 2022; Chiu et al., 2022; Radford et al., 2023). In contrast, acoustic tokens produced by neural speech codecs preserve fine-grained waveform details and enable high-quality synthesis, yet often exhibit weaker alignment with textual content (Défossez et al., 2022; Kumar et al., 2023; Wang et al., 2023; Yang et al., 2023; Xin et al., 2024). An ideal speech codec should support both semantic structure and acoustic fidelity. Several recent approaches jointly model semantic and acoustic information within a

* Corresponding author.



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9350–9369

unified speech codec and demonstrate strong performance on downstream speech language modeling tasks (Zhang et al., 2023b; Défossez et al., 2024; Ye et al., 2025a). However, a key challenge in modeling both semantic and acoustic information **lies in the inherent conflict between these tasks, particularly at low bitrates, where achieving high performance in both remains difficult** (Défossez et al., 2024).

In this work, we propose **XY-Tokenizer**, a low-bitrate speech codec that jointly models semantic and acoustic information. XY-Tokenizer adopts a dual-tower architecture to mitigate semantic–acoustic conflict by minimizing parameter sharing in a multi-task framework. We introduce a multi-stage training strategy: in the first stage, the codec is trained with an LLM-based ASR objective to align speech tokens with text, together with a standard codec reconstruction loss, forming an **X-shaped** architecture that supports joint semantic alignment and acoustic modeling. In the second stage, we further enhance fine-grained speech quality via adversarial training, where the encoder and quantizer are frozen to preserve semantic alignment and only the decoder is optimized, resulting in a **Y-shaped** architecture specialized for high-fidelity reconstruction.

Our contributions can be summarized as follows:

- We propose **XY-Tokenizer**, a 1 kbps speech codec that mitigates the semantic–acoustic conflict through a dual-tower, multi-stage, multi-task learning framework. It aligns speech tokens with text using an LLM-based ASR objective while ensuring high-quality speech reconstruction via a codec decoder.
- We analyze the limitations of existing speech codecs, particularly the semantic–acoustic conflict at low bitrates, and identify effective strategies to mitigate this issue, including leveraging pretrained ASR models, minimizing parameter sharing, and scaling model capacity.
- Extensive experiments and ablation studies demonstrate that XY-Tokenizer achieves competitive performance in both semantic alignment and acoustic reconstruction, matching state-of-the-art codecs that are typically specialized in a single aspect, and delivers strong results on LLM-based speech understanding and generation tasks.

## 2 Method

| Codec | Shared modules | SIM ↑ | WER ↓ |
|---|---|---|---|
| SPT | Encoder + Quantizer | 0.65 | 0.34 |
| Mimi-8 | Encoder | 0.73 | **0.28** |
| XCodec 2.0 | Quantizer | **0.82** | 0.30 |

Table 1: Effect of parameter sharing on semantic alignment and reconstruction quality. SPT denotes SpeechTokenizer-x (Appendix B); WER is reported in the range $[0, 1]$.

### 2.1 XY-Tokenizer

**Motivation.** An ideal speech codec should effectively balance two goals: high-fidelity audio reconstruction and strong semantic alignment with text (Zhang et al., 2023b; Yang et al., 2024b). However, these two objectives often conflict, as optimizing for one can degrade the other (Défossez et al., 2024). Our empirical analysis, as shown in Table 1, suggests that decreasing the number of shared parameters between semantic and acoustic modeling pathways effectively mitigates the trade-off between high-fidelity audio reconstruction and strong semantic alignment. Moreover, semantic modeling can be effectively approached through automatic speech recognition (ASR) tasks (Radford et al., 2023; Zeng et al., 2024; Zhao et al., 2025), while acoustic modeling aligns closely with reconstruction through a codec decoder (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2023). To this end, we propose a dual-channel codec architecture that jointly models semantic and acoustic information in a multi-task setup, combining ASR and audio reconstruction, with shared parameters limited to the residual vector quantization (RVQ) module and its adjacent components.

**Architecture.** The encoder comprises two parallel branches: a **semantic channel** and an **acoustic channel**, both initialized from a pretrained Whisper encoder. The semantic encoder is kept frozen to provide stable linguistic representations, while the acoustic encoder is trainable to capture fine-grained paralinguistic information for high-fidelity reconstruction. The outputs of the two encoders are concatenated and passed to a residual vector quantization (RVQ) module, which produces discrete speech tokens. The quantizer serves as the *only shared component* between the semantic and acoustic pathways. The decoder consists of two task-specific branches operating on the quantized

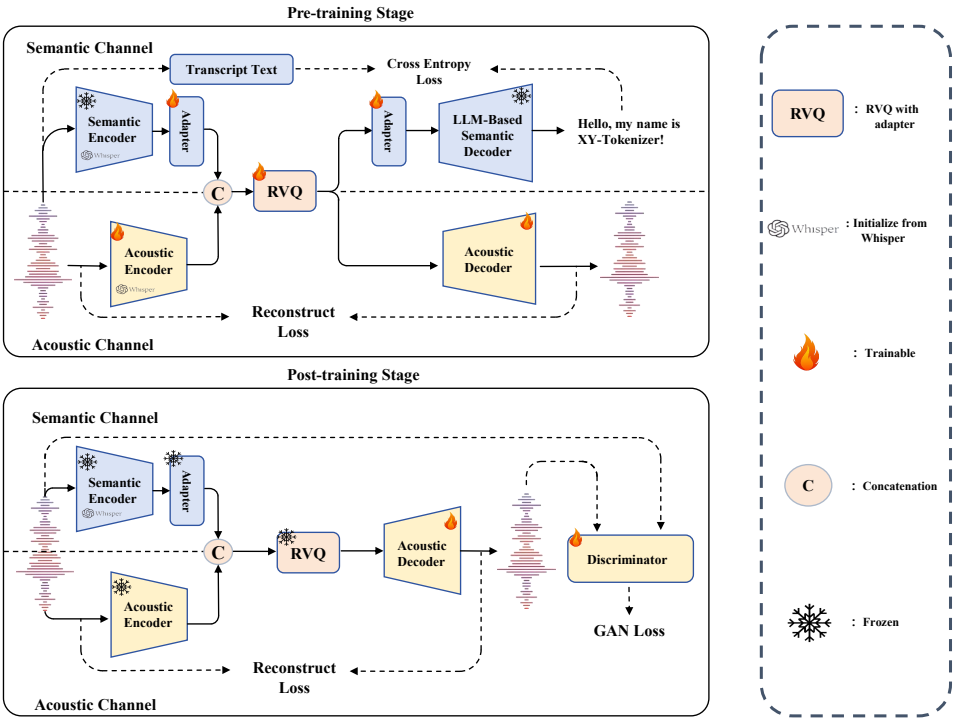


Figure 2: Illustration of **XY-Tokenizer**. The upper half depicts the pre-training stage, aligning **XY-Tokenizer** with text while preserving coarse acoustic features. The lower half illustrates the post-training stage, modeling finer-grained acoustic features. Model architecture and training procedure are detailed in Section 2.

representations: a decoder-only language model in the semantic branch generates text transcriptions, while the acoustic branch reconstructs the speech waveform. Architectural details are provided in Appendix A.

### 2.2 Two-Stage Training Strategy

We adopt a two-stage training strategy to balance semantic alignment and high-fidelity audio reconstruction. As illustrated in Figure 2, the first stage jointly models semantic and coarse acoustic information, while the second stage focuses on refining fine-grained acoustic details.

#### 2.2.1 Pre-Training Stage

In the pre-training stage, we focus on two tasks: audio reconstruction and automatic speech recognition (ASR). All model parameters are trainable, except for the weights of the semantic encoder, initialized from the Whisper encoder (Radford et al., 2023), and the large language model (LLM) which is initialized from Qwen2.5 (Yang et al., 2024a). To align with text generation, we use the cross-entropy loss for the LLM, defined as:

$$\mathcal{L}_{asr} = -\sum_{t=1}^{T} \log p(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{f}; \theta_{LLM}) \quad (1)$$

where $\mathbf{y}_t$ denotes the $t$-th text token in the output sequence, $\mathbf{y}_{<t}$ denotes the sequence of preceding tokens, $\mathbf{f}$ represents the audio features input to the LLM, $T$ is the total number of predicted text tokens, and $\theta_{LLM}$ denotes the parameters of the LLM.

For modeling acoustic features, we employ a multi-scale mel-spectrogram reconstruction loss:

$$\mathcal{L}_{rec} = \sum_{i \in e} \|S_i(\mathbf{x}) - S_i(\hat{\mathbf{x}})\|_1 \quad (2)$$

where $S_i$ is the mel-spectrogram at scale $i$, computed using a normalized short-time Fourier transform (STFT) with a window size of $2^i$ and a hop length of $2^{i-2}$. The set of scales is defined as $e = \{5, \ldots, 11\}$. Here, $\mathbf{x}$ is the ground-truth audio waveform, and $\hat{\mathbf{x}}$ is the predicted waveform from the acoustic decoder.

Additionally, we incorporate a commitment loss to ensure effective quantization:

$$\mathcal{L}_{cmt} = \sum_{i=1}^{N_q} \|\mathbf{z}_i - \mathrm{sg}(q_i(\mathbf{z}_i))\|_1 \quad (3)$$

where $\mathbf{z}_i$ is the input to the $i$-th layer of the quantizer, $q_i(\mathbf{z}_i)$ is its quantized output, $N_q$ is the number of quantizers, and sg denotes the stop-gradient

operation (Van Den Oord et al., 2017), which prevents gradients from propagating to the quantizer's codebook.

The total loss for the pre-training stage is a weighted combination of individual losses:

$$\mathcal{L}_{pre} = \lambda_{asr}\mathcal{L}_{asr} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{cmt}\mathcal{L}_{cmt} \quad (4)$$

where $\lambda_{asr}, \lambda_{rec}, \lambda_{cmt}$ are hyperparameters that balance the weights of each loss term.

#### 2.2.2 Post-Training Stage

While pre-training provides strong semantic representations, reconstructed audio may still exhibit perceptual artifacts. The post-training stage therefore focuses on refining fine-grained acoustic details. We adopt a generative adversarial network (GAN) framework for post-training. During this stage, the encoder and quantizer are fixed to preserve the learned speech token representations, and the semantic decoder is removed. All remaining components of the codec are kept identical to the pre-training stage and remain trainable. For the discriminator, we employ a combination of multi-period, multi-scale, and multi-scale STFT discriminators (MPD, MSD, and MS-STFTD) to improve perceptual audio quality (Kong et al., 2020; Kumar et al., 2019; Défossez et al., 2022).

The discriminator loss follows the least squares GAN (LSGAN) formulation (Mao et al., 2017), given by:

$$\mathcal{L}_D(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{K}\sum_{k=1}^{K}(1 - D_k(\mathbf{x}))^2 + D_k^2(\hat{\mathbf{x}}) \quad (5)$$

where $D_k$ represents the $k$-th discriminator, $K$ is the total number of discriminators, $\mathbf{x}$ is the ground-truth audio, and $\hat{\mathbf{x}}$ is the predicted audio.

For the generator loss, we use the same multi-scale mel-spectrogram reconstruction loss as in the pre-training stage, denoted $\mathcal{L}_{rec}$. Additionally, we include a feature matching loss:

$$\mathcal{L}_{feat}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L_k}\sum_{l=1}^{L_k}\frac{\left\|D_k^l(\mathbf{x}) - D_k^l(\hat{\mathbf{x}})\right\|_1}{\text{mean}(\left\|D_k^l(\mathbf{x})\right\|_1)} \quad (6)$$

where $D_k^l$ denotes the feature representation from the $l$-th layer of the $k$-th discriminator, $L_k$ is the number of layers of the $k$-th discriminator, and the mean is computed over all dimensions of $D_k^l(\mathbf{x})$. We also incorporate an adversarial loss:

$$\mathcal{L}_{adv}(\hat{\mathbf{x}}) = \frac{1}{K}\sum_{k=1}^{K}(1 - D_k(\hat{\mathbf{x}}))^2 \quad (7)$$

The total generator loss is a weighted combination of these terms:

$$\mathcal{L}_G(\mathbf{x}, \hat{\mathbf{x}}) = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{adv}\mathcal{L}_{adv} \quad (8)$$

where $\lambda_{rec}, \lambda_{feat}, \lambda_{adv}$ are hyperparameters that balance the contributions of each loss term.

## 3 Experiments

### 3.1 Settings

**Dataset and Training Details.** We trained XY-Tokenizer using the full Emilia dataset, comprising approximately 101k hours of audio data, equivalent to about 37 million (audio, transcription) pairs (He et al., 2024). All audio data was resampled to 16 kHz.

In the **pre-training stage**, audio clips longer than 30 seconds were truncated to the first 30 seconds, while clips shorter than 30 seconds were padded to 30 seconds, with loss computed only on the non-padded portions. We trained the model for 800k steps using DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) on 32 NVIDIA H100 GPUs, with a batch size of 4 per GPU and a maximum learning rate of $1 \times 10^{-4}$. We used the AdamW optimizer with a weight decay of 0.01 (Loshchilov and Hutter, 2017). A cosine learning rate scheduler was applied during pre-training.

In the **post-training stage**, we randomly sampled 5-second segments from each audio clip for training, using a single NVIDIA H100 GPU with a batch size of 16. The generator and discriminator were optimized with maximum learning rates of $1 \times 10^{-5}$ and $1 \times 10^{-4}$, respectively.

For both the **pre-training stage** and **post-training stage**, we set $\lambda_{rec} = 15$. In the pre-training stage, we set $\lambda_{asr} = 20$ and $\lambda_{cmt} = 1$. In the post-training stage, we set $\lambda_{feat} = 1$ and $\lambda_{adv} = 1$.

**Model Details.** We have detailed the codec architecture in Section 2.1 and Appendix A.

**Baselines.** We use SpeechTokenizer (Zhang et al., 2023b), Semanticodec (Liu et al., 2024), Mimi (Défossez et al., 2024), and XCodec2.0 (Ye et al., 2025b), as our baseline codecs, which simultaneously model semantic and acoustic information. Details of these models are provided in Appendix B. Additionally, we include EnCodec (Défossez et al., 2022), BigCodec (Xin et al., 2024), Descript Audio Codec (Kumar et al., 2023), WavTokenizer (Ji et al., 2024b), which exclusively model acoustic information.

| | | EN – LibriSpeech | | | | | EN – VoxPopuli | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | kbps | WER↓ | SIM↑ | STOI↑ | PESQ↑ | MUS.↑ | WER↓ | SIM↑ | STOI↑ | PESQ↑ | MUS.↑ |
| *Reference* | – | – | – | – | – | 91.23 | – | – | – | – | 90.18 |
| EnCodec | 1.50 | 0.37 | 0.60 | 0.85 | 1.94 / 1.56 | 53.12 | 0.55 | 0.68 | 0.84 | 2.01 / 1.56 | 57.36 |
| DAC | 1.50 | 0.98 | 0.49 | 0.83 | 1.91 / 1.51 | 42.12 | 0.97 | 0.54 | 0.82 | 1.83 / 1.42 | 40.27 |
| BigCodec | 1.04 | 0.49 | 0.84 | **0.93** | **3.26 / 2.68** | 89.54 | 0.56 | 0.84 | **0.92** | **3.06 / 2.46** | 85.25 |
| WavTokenizer | 0.90 | 0.53 | 0.68 | 0.91 | 2.89 / 2.38 | 83.46 | 0.61 | 0.72 | 0.91 | 2.80 / 2.35 | 81.91 |
| SpeechTokenizer | 1.50 | 0.22 | 0.52 | 0.84 | 2.00 / 1.57 | 58.34 | 0.42 | 0.52 | 0.84 | 1.92 / 1.49 | 57.17 |
| Semanticodec | 1.50 | 0.58 | 0.72 | 0.88 | 2.63 / 2.02 | 77.75 | 0.67 | 0.75 | 0.88 | 2.58 / 2.07 | 74.74 |
| Mimi | 1.10 | 0.28 | 0.73 | 0.90 | 2.79 / 2.24 | 72.74 | 0.41 | 0.79 | 0.90 | 2.80 / 2.25 | 74.46 |
| XCodec 2.0 | 0.80 | 0.30 | 0.82 | 0.91 | 3.03 / 2.43 | 89.01 | 0.43 | 0.85 | 0.91 | 2.94 / 2.39 | 90.02 |
| **XY-Tokenizer** | 1.00 | **0.13** | **0.85** | 0.92 | 3.10 / 2.50 | **90.17** | **0.25** | **0.87** | 0.91 | 2.99 / 2.49 | **90.14** |
| | | **ZH – AISHELL-2** | | | | | **ZH – CommonVoice** | | | | |
| | kbps | CER↓ | SIM↑ | STOI↑ | PESQ↑ | MUS.↑ | CER↓ | SIM↑ | STOI↑ | PESQ↑ | MUS.↑ |
| *Reference* | – | – | – | – | – | 91.00 | – | – | – | – | 90.10 |
| EnCodec | 1.50 | 0.44 | 0.45 | 0.82 | 1.80 / 1.48 | 40.54 | 0.53 | 0.60 | 0.81 | 1.94 / 1.59 | 41.44 |
| DAC | 1.50 | 0.50 | 0.41 | 0.79 | 1.67 / 1.37 | 39.47 | 0.60 | 0.57 | 0.80 | 1.86 / 1.53 | 49.45 |
| BigCodec | 1.04 | 0.50 | 0.69 | **0.87** | 2.54 / 2.06 | 87.34 | 0.55 | 0.75 | **0.86** | 2.48 / 2.01 | 88.13 |
| WavTokenizer | 0.90 | 0.43 | 0.61 | 0.85 | 2.24 / 1.88 | 80.34 | 0.54 | 0.71 | 0.84 | 2.36 / 1.95 | 85.97 |
| SpeechTokenizer | 1.50 | 0.38 | 0.38 | 0.76 | 1.60 / 1.33 | 35.42 | 0.48 | 0.48 | 0.77 | 1.73 / 1.41 | 37.14 |
| Semanticodec | 1.50 | 0.34 | 0.67 | 0.84 | 2.39 / 1.92 | 77.42 | 0.50 | 0.73 | 0.82 | 2.38 / 1.87 | 75.91 |
| Mimi | 1.10 | 0.53 | 0.59 | 0.85 | 2.24 / 1.78 | 73.43 | 0.60 | 0.72 | 0.84 | 2.43 / 1.97 | 79.21 |
| XCodec 2.0 | 0.80 | 0.37 | 0.73 | 0.86 | 2.45 / 1.96 | 89.49 | 0.43 | 0.80 | 0.84 | 2.45 / 1.96 | 86.55 |
| **XY-Tokenizer** | 1.00 | **0.11** | **0.79** | **0.87** | **2.63 / 2.12** | **89.66** | **0.21** | **0.84** | 0.85 | **2.54 / 2.05** | **89.64** |

Table 2: Semantic and acoustic evaluation on English and Chinese datasets. WER/CER are measured by an ASR probing task and reported in the range $[0, 1]$ (lower is better), while higher SIM, STOI, PESQ (NB/WB), and MUS. indicate better speech reconstruction quality; MUS. denotes the MUSHRA score.

### 3.2 Metrics

**Reconstruction Evaluation.** Speech reconstruction quality is evaluated using both objective and subjective metrics. For objective evaluation, we report speaker similarity (SIM), which is computed as the cosine similarity between speaker embeddings extracted from the original and reconstructed audio using a pretrained speaker verification model[1]. We additionally report short-time objective intelligibility (STOI) (Taal et al., 2010) and perceptual evaluation of speech quality (PESQ) (Rix et al., 2001), following common practice. For subjective evaluation, we conduct a listening test inspired by the MUSHRA protocol (Series, 2014), where listeners rate the perceptual quality of each audio sample on a scale from 1 to 100.

Reconstruction evaluation is conducted on multiple datasets covering both English and Chinese, including LibriSpeech test-clean (Panayotov et al., 2015), VoxPopuli-EN dev-clean (Wang et al., 2021), AISHELL-2 dev_ios (Du et al., 2018), and the Common Voice ZH test set (Ardila et al., 2020). These datasets span diverse recording conditions, ranging from clean speech to in-the-wild noisy scenarios, enabling a comprehensive assessment of reconstruction quality and robustness across languages and acoustic environments.

**Semantic Evaluation.** To evaluate the semantic alignment between the codec and textual representations, we conduct an **automatic speech recognition probing task**, following the protocol of the SUPERB benchmark (Yang et al., 2021). In this task, the pretrained codec is kept frozen, and a lightweight downstream ASR model is trained on top of the **quantized speech embeddings** to assess how well semantic information is preserved. We use word error rate (WER) and character error rate (CER) to measure the alignment between the speech tokens and the corresponding text. Details of this task can be found in Appendix C.

### 3.3 Evaluation Results.

As shown in Table 2, XY-Tokenizer achieves a strong balance between semantic alignment and acoustic reconstruction at comparable bitrates. At approximately 1 kbps, XY-Tokenizer delivers

[1] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification

| | EN | ZH |
|---|---|---|
| EnCodec | 0.15 / 0.14 / 0.34 | 0.38 / 0.21 / 0.77 |
| DAC | 0.18 / 0.18 / 0.40 | 0.55 / 0.24 / 1.63 |
| Semantic Codec | 0.18 / 0.19 / 0.39 | 0.41 / 0.19 / 1.47 |
| BigCodec | 0.12 / 0.13 / 0.30 | 0.58 / 0.18 / 1.37 |
| SpeechTokenizer | 0.10 / 0.10 / 0.24 | 0.39 / 0.21 / 1.08 |
| Mimi | 0.13 / 0.13 / 0.30 | 0.64 / 0.33 / 0.99 |
| WavTokenizer | 0.17 / 0.17 / 0.38 | 0.51 / 0.19 / 1.57 |
| XCodec 2.0 | 0.12 / 0.13 / 0.27 | 0.29 / 0.18 / 0.38 |
| **XY-Tokenizer** | **0.06 / 0.07 / 0.12** | **0.17 / 0.12 / 0.19** |

Table 3: LLM-based speech understanding performance on English and Chinese datasets. For EN, values correspond to WER on LibriSpeech *dev-clean / test-clean / test-other*; for ZH, values correspond to CER on AISHELL-2 test *iOS / Android / Mic*.

| | ZH | | EN | |
|---|---|---|---|---|
| | **SIM↑** | **WER↓** | **SIM↑** | **WER↓** |
| Llasa-1B-250k | 0.669 | 0.0189 | 0.572 | 0.0322 |
| Llasa-3B-250k | 0.675 | 0.0160 | 0.579 | 0.0314 |
| Llasa-8B-250k | 0.684 | 0.0159 | 0.574 | 0.0297 |
| Spark-TTS | 0.672 | **0.0120** | 0.584 | 0.0198 |
| Mimi-TTS | 0.590 | 0.0247 | 0.550 | 0.0430 |
| **XY-Tokenizer-TTS** | **0.695** | 0.0174 | **0.629** | **0.0181** |

Table 4: LLM-based speech generation performance on the Seed-TTS-Eval benchmark.

high reconstruction quality while maintaining low WER/CER, outperforming or matching state-of-the-art codecs that typically excel in only one aspect. This advantage is consistent across both English and Chinese datasets and holds under clean, noisy, and in-the-wild conditions. These results demonstrate that XY-Tokenizer provides a robust low-bitrate representation that preserves both linguistic structure and acoustic fidelity across languages and recording scenarios.

### 3.4 LLM-Based Speech Understanding and Generation

We evaluate **XY-Tokenizer** and prior speech tokenizers on LLM-based speech understanding and generation tasks using a *purely autoregressive* decoder-only formulation for both tasks. Implementation details are provided in Appendix D.

For speech understanding, as shown in Table 3, XY-Tokenizer consistently achieves the lowest error rates across all English and Chinese test sets, outperforming both reconstruction-oriented codecs and semantic tokenizers. The improvements are especially pronounced under challenging conditions such as LibriSpeech *test-other* and AISHELL-2 *Mic*.

For speech generation, Table 4 shows that XY-Tokenizer outperforms Mimi-8 in bilingual zero-shot TTS, achieving higher speaker similarity and lower word error rates across both languages. We compare against Mimi-8 as it shares a similar codec configuration (12.5 Hz frame rate and 8-layer RVQ), enabling a controlled and fair comparison under the same autoregressive setting. Moreover, XY-Tokenizer matches or exceeds strong purely autoregressive TTS systems in speaker similarity.

Overall, these results demonstrate that XY-Tokenizer produces speech tokens that are effective for both speech understanding and autoregressive speech generation, validating its ability to mitigate the semantic–acoustic conflict in low-bitrate settings.

## 4 Analysis

### 4.1 Ablation Study

We conduct ablation studies to analyze how different design choices affect the balance between semantic alignment and acoustic reconstruction in XY-Tokenizer. Unless otherwise specified, all ablations are performed during pre-training under identical settings.

**Shared Parameters Cause Conflicts.** We compare XY-Tokenizer with single-channel variants that share parameters between semantic and acoustic modeling by removing the semantic encoder. As shown in Table 5, increasing the encoder capacity in the single-channel setting does not improve reconstruction quality, while achieving similar ASR probing performance. In contrast, the dual-encoder design of XY-Tokenizer consistently yields higher speaker similarity. These results indicate that **disentangling semantic and acoustic modeling is more effective than scaling model capacity under shared representations**, and that parameter sharing is a primary source of semantic–acoustic conflict.

**Fixing the LLM Encourages Unified Discrete Representations.** We study the effect of making the pretrained semantic decoder (LLM) trainable during pre-training. As shown in Table 6, a trainable LLM reduces the decoder-side LLM WER, but consistently degrades ASR probing performance. This indicates that when the LLM is trainable, se-

| Model | Encoder | Params | WER ↓ | SIM ↑ | STOI ↑ | PESQ ↑ |
|---|---|---|---|---|---|---|
| **XY-Tokenizer** | small | 259M | 0.13 | **0.80** | **0.92** | **3.12 / 2.61** |
| Single-channel | small | 115M | 0.13 | 0.77 | 0.92 | 2.92 / 2.45 |
| Single-channel | medium | 356M | 0.12 | 0.77 | 0.92 | 2.88 / 2.38 |

Table 5: Impact of reducing parameter sharing between semantic and acoustic modeling. Params denote the number of encoder parameters.

| LLM Setting | Steps | Probing WER ↓ | LLM WER ↓ | SIM ↑ | STOI ↑ | PESQ ↑ |
|---|---|---|---|---|---|---|
| Fixed | 200K | 0.13 | 0.06 | 0.80 | 0.92 | 3.12 / 2.61 |
| | 400K | 0.13 | 0.05 | 0.81 | 0.93 | 3.22 / 2.67 |
| | 800K | **0.13** | 0.05 | **0.84** | 0.93 | 3.35 / 2.83 |
| Trainable | 200K | 0.18 | **0.03** | 0.80 | 0.93 | 3.21 / 2.69 |
| | 400K | 0.20 | 0.04 | 0.82 | 0.93 | 3.32 / 2.77 |
| | 800K | 0.24 | **0.03** | **0.84** | **0.94** | **3.46 / 2.96** |

Table 6: Effect of LLM trainability during pre-training. Probing WER is measured on the ASR probing task (Section 3.2), while LLM WER is computed from text decoded by the semantic decoder.

| | Params | WER↓ | SIM↑ | PESQ↑ | RTF↓ |
|---|---|---|---|---|---|
| Base | 246M | 0.57 | 0.75 | 2.96 / 2.41 | **0.0034** |
| Small | 520M | 0.13 | 0.80 | 3.12 / 2.61 | 0.0053 |
| Large | 2185M | **0.09** | **0.82** | **3.18 / 2.68** | 0.0136 |

Table 7: Effect of model scaling on semantic–acoustic performance.

| Encoder Init | ASR Superv. | WER↓ |
|---|---|---|
| × | ✓ | 0.27 |
| ✓ | × | 0.58 |
| ✓ | ✓ | **0.13** |

Table 8: Effect of audio-to-text supervision on semantic alignment. ✓ indicates that the encoder is initialized from pretrained Whisper weights or that an LLM-based ASR objective is applied during pretraining.

mantic and acoustic objectives tend to be optimized separately, with semantic modeling increasingly handled by the decoder rather than the tokenizer. Fixing the LLM instead places semantic supervision on the encoder and quantizer, encouraging the tokenizer to learn more unified, text-aligned discrete representations.

**Scaling Model Capacity Improves Semantic–Acoustic Trade-off.** We study the effect of scaling up model size while keeping the architecture unchanged. As shown in Table 7, increasing model size consistently improves both semantic alignment and acoustic reconstruction.

The performance gap between the base model and larger variants suggests that limited capacity constrains the model's ability to learn a unified discrete representation that jointly captures semantic and acoustic information. While further scaling continues to bring gains, it also incurs substantially higher training and inference costs. Overall, the *small* configuration provides a favorable balance between performance and computational efficiency.

**Audio-to-Text Supervision Improves Semantic Alignment.** We analyze two mechanisms that contribute to speech–text alignment during speech codec training: encoder initialization and explicit ASR supervision. As shown in Table 8, initializing the encoder with pretrained Whisper weights significantly improves ASR probing performance compared to training from scratch under the same architecture, indicating a strong inductive bias for text alignment that reduces training difficulty.

Beyond initialization, we further examine the effect of attaching a semantic decoder and optimizing an explicit ASR loss during codec training. Removing this ASR supervision leads to a substantial degradation in ASR probing performance, confirming that explicit ASR supervision remains essential for aligning discrete speech representations with text, even when starting from a strongly pretrained encoder.

| | LibriSpeech | | VoxPopuli | |
|---|---|---|---|---|
| | WER | ΔWER | WER | ΔWER |
| HuBERT-base | 0.06 | – | 0.20 | – |
| WavLM-large | 0.04 | – | 0.12 | – |
| W2v-BERT | 0.12 | – | 0.19 | – |
| Whisper-small | 0.07 | – | 0.15 | – |
| SpeechTokenizer | 0.22 | +0.16 | 0.42 | +0.22 |
| Mimi | 0.28 | +0.24 | 0.41 | +0.29 |
| XCodec 2.0 | 0.30 | +0.18 | 0.43 | +0.24 |
| **XY-Tokenizer** | **0.13** | **+0.06** | **0.25** | **+0.10** |

Table 9: Semantic alignment comparison between pretrained SSL/ASR encoders and speech codecs. WER is measured on the ASR probing task (Section 3.2). Pretrained encoders are listed above the horizontal divider, and speech codecs are listed below. ΔWER denotes the absolute gap between each speech codec and its corresponding teacher model (for distillation-based codecs) or pretrained encoder.

### 4.2 ASR Supervision Narrows the Semantic Gap to Pretrained Encoders

We further analyze ASR-driven semantic modeling in XY-Tokenizer by comparing its semantic alignment with strong pretrained speech encoders. Unlike prior codecs that rely on distillation from pretrained encoders, XY-Tokenizer directly leverages speech–text pairs through an explicit LLM-based ASR loss to align discrete representations with text.

Table 9 reports WER on the ASR probing task for pretrained encoders and speech codecs, together with the absolute gap (ΔWER) to the corresponding pretrained encoder. XY-Tokenizer consistently shows a substantially smaller gap to its semantic upper bound than prior codec baselines across datasets.

These results indicate that **explicit ASR supervision provides a more effective mechanism for injecting linguistic structure into low-bitrate discrete representations than distillation-based approaches**, enabling XY-Tokenizer to more closely match the semantic abstraction level of pretrained ASR/SSL models.

### 4.3 Encoder-level Analysis: Semantic vs. Acoustic Representations

To understand why separating the encoder into semantic and acoustic components mitigates the semantic–acoustic conflict, we analyze the representational properties of each encoder while controlling for other factors. We evaluate their ability to align speech representations with text and to model fine-grained acoustic details. Specifically, we freeze the encoder and independently feed its outputs into a decoder trained without an adversarial discriminator, and perform the ASR probing task described in Section 3.2.

| | WER↓ | SIM↑ | STOI↑ | PESQ↑ |
|---|---|---|---|---|
| SemEnc | **0.06** | 0.51 | 0.83 | 1.53 / 1.27 |
| AcuEnc | 0.23 | **0.93** | **0.97** | **3.99 / 3.63** |

Table 10: Comparison between semantic and acoustic encoders. SemEnc and AcuEnc denote the semantic encoder and acoustic encoder, respectively.

Table 10 summarizes the results. The semantic encoder achieves substantially lower WER in ASR probing, indicating stronger speech–text alignment. In contrast, the acoustic encoder consistently outperforms the semantic encoder on reconstruction-related metrics, including SIM, STOI, and PESQ, demonstrating superior modeling of acoustic fidelity.

These results reveal a clear functional specialization between the two encoders: the semantic encoder primarily captures text-aligned representations, while the acoustic encoder focuses on signal fidelity. This separation of roles provides direct evidence that reducing parameter sharing between semantic and acoustic modeling effectively alleviates the semantic–acoustic conflict in XY-Tokenizer.

## 5 Related Work

Related work on speech large language models and neural speech codecs is provided in Appendix I.

## 6 Conclusion

We presented **XY-Tokenizer**, a low-bitrate (around 1 kbps) speech codec designed to mitigate the semantic–acoustic conflict. XY-Tokenizer adopts a dual-tower architecture with minimal parameter sharing and a two-stage training pipeline that combines text-aligned supervision with codec reconstruction and perceptual refinement. Experimental results and analyses support the effectiveness of these design choices in balancing semantic alignment and reconstruction fidelity. Overall, XY-Tokenizer offers a practical low-bitrate representation that effectively facilitates both speech understanding and speech generation for LLM-based models.

## Limitations

Despite the effectiveness of the proposed approach, this work has several limitations. The scaling behavior of speech codecs remains insufficiently understood. In particular, how parameter count and dataset size affect training efficiency and generalization warrants further investigation. In addition, the proposed multi-stage training strategy introduces extra complexity, which may limit scalability under constrained computational budgets.

## Ethical Considerations

This work studies a speech codec at the representation level and is intended for research purposes. Ethical considerations such as privacy, bias, and misuse are primarily determined by downstream applications rather than the codec itself. We encourage responsible use in accordance with established ethical guidelines.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Christopher F Barnes, Syed A Rizvi, and Nasser M Nasrabadi. 1996. Advances in residual vector quantization: A review. *IEEE transactions on image processing*, 5(2):226–262.

James Betker. 2023. Better speech synthesis through scaling. *arXiv preprint arXiv:2305.07243*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. 2022. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720.

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

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. 2018. Aishell-2: Transforming mandarin asr research into industrial scale. *arXiv preprint arXiv:1808.10583*.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, and 1 others. 2024a. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Zhihao Du, Shiliang Zhang, Kai Hu, and Siqi Zheng. 2024b. Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 591–595. IEEE.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, and 1 others. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890. IEEE.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, and 1 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024a. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, and 1 others. 2024b. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. *arXiv preprint arXiv:2408.16532*.

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.

Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

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.

Siddique Latif, Moazzam Shoukat, Fahad Shamshad, Muhammad Usama, Yi Ren, Heriberto Cuayáhuitl, Wenwu Wang, Xulong Zhang, Roberto Togneri, Erik Cambria, and 1 others. 2023. Sparks of large audio models: A survey and outlook. *arXiv preprint arXiv:2308.12792*.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. 2025. Dualcodec: A low-framerate, semantically-enhanced neural audio codec for speech generation. *arXiv preprint arXiv:2505.13000*.

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

Haohe Liu, Xuenan Xu, Yi Yuan, Mengyue Wu, Wenwu Wang, and Mark D Plumbley. 2024. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *IEEE Journal of Selected Topics in Signal Processing*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, and 1 others. 2025. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52.

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. 2024. Scaling transformers for low-bitrate high-quality speech coding. *arXiv preprint arXiv:2411.19842*.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. 2001. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE.

B Series. 2014. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*, 2.

Hubert Siuzdak. 2023. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE.

Marco Tagliasacchi, Yunpeng Li, Karolis Misiunas, and Dominik Roblek. 2020. Seanet: A multi-modal speech enhancement network. *arXiv preprint arXiv:2009.02095*.

Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, and 1 others. 2025. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2024. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*.

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

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

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*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. 2024b. Uniaudio 1.5: Large language model-driven audio codec is a few-shot audio task learner. *Advances in Neural Information Processing Systems*, 37:56802–56827.

Dongchao Yang, Songxiang Liu, Haohan Guo, Jiankun Zhao, Yuanyuan Wang, Helin Wang, Zeqian Ju, Xubo Liu, Xueyuan Chen, Xu Tan, and 1 others. 2025b. Almtokenizer: A low-bitrate and semantic-rich audio

codec tokenizer for audio language modeling. *arXiv preprint arXiv:2504.10344*.

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

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, and 1 others. 2021. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.

Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, and 1 others. 2025a. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25697–25705.

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

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.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Bowen Zhang, Congchao Guo, Geng Yang, Hang Yu, Haozhe Zhang, Heidi Lei, Jialong Mai, Junjie Yan, Kaiyue Yang, Mingqi Yang, and 1 others. 2025. Minimax-speech: Intrinsic zero-shot text-to-speech with a learnable speaker encoder. *arXiv preprint arXiv:2505.07916*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023a. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.

Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao, Yuxuan Wang, Bin Zhang, and 1 others. 2024. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *arXiv preprint arXiv:2410.08035*.

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

Xingjian Zhao, Zhe Xu, Qinyuan Cheng, Zhaoye Fei, Luozhijie Jin, Yang Wang, Hanfu Chen, Yaozhou Jiang, Qinghui Gao, Ke Chen, and 1 others. 2025. Moss-speech: Towards true speech-to-speech models without text guidance. *arXiv preprint arXiv:2510.00499*.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. 2025. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22968–22977.

# A Details of Architecture

## A.1 Adapter

To enhance the flexibility of embeddings, we incorporate lightweight Transformer-based (Vaswani et al., 2017) adapter modules at multiple components of the XY-Tokenizer. Each adapter consists of a 4-layer Transformer with a hidden dimension of 768, a feed-forward network (FFN) dimension of 3072, and 12 attention heads. Adapters are placed after the semantic encoder, before and after the quantizer, and before the LLM-based semantic decoder.

## A.2 Encoder

The input waveform is resampled to 16 kHz, and an 80-channel mel-spectrogram is computed using a 25 ms window length and a 10 ms hop length to serve as the input to the encoder.

The semantic encoder adopts the Whisper-small encoder configuration and processes the mel-spectrogram through the following modules: (1) a 1D convolutional layer with a kernel size of 3 and a stride of 1, projecting the 80-dimensional input to a hidden dimension of 768; (2) a GELU activation function (Hendrycks and Gimpel, 2016); (3) a second 1D convolutional layer with a kernel size of 3 and a stride of 2, reducing the sequence length by a factor of 2; (4) another GELU activation function; (5) sinusoidal positional embeddings; (6) a transformer with 12 layers, 12 attention heads, a dimension of 768, and a feed-forward network dimension of 3072. The semantic encoder's output is then passed to (7) an adapter module (detailed in Appendix A.1). The semantic encoder's parameters are fixed during training.

The acoustic encoder follows a similar architecture to the semantic encoder but is trainable and excludes the adapter module. The outputs of the semantic and acoustic encoders are concatenated along the feature dimension.

## A.3 Quantizer

We employ a residual vector quantizer (RVQ) with 8 layers and a codebook size of 1024 per layer. The codebook is updated using an exponential moving average (EMA) with a weight decay of 0.99. To prevent codebook collapse, unused codebook entries are randomly replaced with input vectors from the current batch after several training steps. The codebook is initialized using $k$-means clustering with 10 iterations. A $4\times$ downsampling convolutional layer is applied before the quantizer, reducing the encoder's 50 Hz embeddings to 12.5 Hz, resulting in a bitrate of 1 kbps for our proposed XY-Tokenizer. Adapter modules (detailed in Appendix A.1) are placed before the downsampling convolution and after the quantizer.

## A.4 Decoder

The decoder processes quantized features through two distinct pathways: the semantic decoder for text prediction and the acoustic decoder for audio reconstruction.

The semantic decoder takes the output of the quantizer as input, passes it through an adapter (detailed in Appendix A.1), and uses the resulting features as conditioning input for a decoder-only large language model (LLM). The LLM, based on Qwen2.5-0.5B (Yang et al., 2024a), has a hidden dimension of 896, an intermediate layer size of 4864, and 24 layers, generating the final predicted text corresponding to the input speech.

The acoustic decoder takes the output of the quantizer as input, applies a $4\times$ upsampling convolution to reach 50 Hz, and follows a structure symmetric to the acoustic encoder to achieve 100 Hz. Finally, a 30-layer Vocos model (Siuzdak, 2023) with a hop size of 160 reconstructs the 16 kHz audio waveform.

## A.5 Discriminators.

To ensure high perceptual quality, we employ three discriminator models: multi-period discriminator (MPD) (Kong et al., 2020), multi-scale discriminator (MSD) (Kumar et al., 2019), and multi-scale short-time Fourier transform discriminator (MS-STFTD) (Défossez et al., 2022). The parameters of our discriminator models are consistent with those used in SpeechTokenizer (Zhang et al., 2023b).

# B Details of Baseline Model

In this section, we provide detailed descriptions of the baseline speech codecs used in our experiments. For a fair comparison, we focus on representative open-source speech codecs operating at approximately 1 kbps.

**Encodec.** We use the official 24 kHz model with a 2-layer residual vector quantization (RVQ) bottleneck[2].

[2] https://huggingface.co/facebook/encodec_24khz

**Descript Audio Codec (DAC).** We adopt the official implementation with a sampling rate of 24 kHz and a 2-layer RVQ bottleneck.

**BigCodec.** We use the official checkpoints released by the authors.

**WavTokenizer.** We use the official `WavTokenizer-large-320-24k-4096` model configuration.

**SpeechTokenizer.** In Sections 3 and 4, we adopt the official `speechtokenizer_hubert_avg` model[3], and use a 3-layer residual vector quantization (RVQ-3) configuration. Additionally, we train a modified variant of SpeechTokenizer using the official codebase, modifying only the RVQ configuration and the distillation weight (`distill_loss_lambda`). Specifically, we reduce the number of RVQ layers from 8 to 3 and set `distill_loss_lambda` to 24 ($5\times$ smaller than the official setting). We refer to this variant as SpeechTokenizer-x.

**Semanticodec.** We use the official model with a token rate of 100 and a semantic vocabulary size of 32,768.

**Mimi.** We employ the official RVQ-8 version provided by the authors.

**XCodec 2.0.** We employ the official checkpoints provided by the authors.

## C Details of ASR Probing Task

This section provides detailed descriptions of the automatic speech recognition (ASR) probing task used to evaluate the semantic alignment of discrete speech representations.

**Downstream ASR Model.** The downstream ASR model used for probing consists of a two-layer bidirectional LSTM trained with a CTC loss for character-level prediction (Hochreiter and Schmidhuber, 1997; Graves et al., 2006). During probing, the pretrained speech codec is kept frozen, and only the ASR model parameters are updated.

**Embedding Upsampling for Low-Bitrate Codecs.** To enable effective alignment in the ASR probing task, particularly for low-bitrate speech codecs, we upsample the quantized speech embeddings for models with a frame rate below 50 Hz to a minimum of 50 Hz using simple replication. This design choice is motivated by the requirements of the connectionist temporal classification (CTC) loss, which relies on a sufficiently long input sequence length ($T$) relative to the target sequence length ($U$) to perform alignment. Specifically, CTC requires $T \geq U$ to assign at least one time step to each target label, and in the worst case $T \geq 2U + 1$ to account for blank symbols between labels and at sequence boundaries. Without upsampling, low-frame-rate discrete representations may violate these constraints, preventing effective optimization. Upsampling ensures that the input sequence length is sufficiently large to satisfy the CTC alignment requirements while preserving the original token information.

**Datasets and Evaluation Protocol.** We evaluate semantic performance on both English and Chinese datasets. For English, we use the LibriSpeech (Panayotov et al., 2015) and VoxPopuli-EN (Wang et al., 2021) corpora. All models are trained on the LibriSpeech `train-clean-100` subset and evaluated on the `dev-clean` set. To assess robustness to domain shift, we further test on the VoxPopuli-EN `dev` split, where 100 hours of speech are randomly sampled from the training subset. Semantic alignment on English datasets is measured using word error rate (WER).

For Chinese evaluation, we adopt AISHELL-2 (Du et al., 2018) and Common Voice ZH (Ardila et al., 2020) as testbeds. We randomly select 100 hours of audio from the AISHELL-2 training subset for training and evaluate on the `AISHELL-2_iOS_dev` set. For Common Voice ZH, models are trained on the official training split and evaluated on a randomly sampled set of 2,000 utterances from the test split. Semantic performance on Chinese datasets is measured using character error rate (CER).

**Training Configuration.** All ASR probing models are trained for 400,000 steps with a maximum learning rate of $1 \times 10^{-4}$. We use a batch size of 4 for English datasets and 16 for Chinese datasets.

## D Details of LLM-based Speech Understanding and Generation Task

### D.1 Speech Understanding

**Model and Training Details.** We adopt Qwen3-0.6B (Yang et al., 2025a) as the backbone of the

[3] https://huggingface.co/fnlp/SpeechTokenizer/tree/main/speechtokenizer_hubert_avg

LLM-based speech understanding model. To accommodate discrete speech representations, we extend the text token vocabulary to include codec-specific speech tokens. Specifically, if a codec produces $N_q$ discrete tokens at each time step, we introduce $N_q$ corresponding speech token embeddings into the vocabulary. At each time step, the embeddings of the $N_q$ speech tokens are summed and used as the input embedding to the LLM, which is trained to autoregressively predict the corresponding text token sequence.

All models are trained under identical settings to ensure fair comparison. We use a global batch size of 128 samples, train the model for 12 epochs, and adopt a maximum learning rate of $1 \times 10^{-4}$ for all experiments.

**Datasets.** We perform bilingual training using both English and Chinese speech data. For English, we use the LibriSpeech training subsets `train-clean-100`, `train-clean-360`, and `train-other-500`, totaling approximately 960 hours of speech. For Chinese, we use the AISHELL-2 training set, which contains approximately 800 hours of speech. All speech data are first encoded into discrete codec tokens before being fed into the LLM.

**Evaluation Protocol.** During evaluation, we report word error rate (WER) on the LibriSpeech `dev-clean`, `test-clean`, and `test-other` sets, and character error rate (CER) on the AISHELL-2 `iOS`, `Android`, and `Mic` test sets. Lower error rates indicate stronger alignment between the discrete speech tokens and the underlying linguistic content.

### D.2 Speech Generation

**Model Architecture and Training Details** We adopt Qwen3-1.7B (Yang et al., 2025a) as the backbone language model and follow the RQ-Transformer architecture used in Moshi (Défossez et al., 2024), which consists of a Temporal Transformer and a lightweight Depth Transformer. The Temporal Transformer is initialized from the pretrained Qwen3-1.7B model, while the Depth Transformer contains four Transformer blocks with a hidden size of 1536 and a feed-forward dimension of 8960. All other architectural and optimization settings are kept consistent with the Temporal Transformer. During training, text tokens and speech tokens are concatenated along the temporal dimension, and the training loss is computed only on the speech tokens. We use a global batch size of approximately 0.52M tokens and train the model for 200k steps, and set the maximum learning rate to $1 \times 10^{-4}$. The model is trained on the VoxBox dataset proposed in Spark-TTS (Wang et al., 2025)

**Evaluation Protocol.** We evaluate zero-shot TTS performance using the Seed-TTS-Eval benchmark (Anastassiou et al., 2024). Both English (EN) and Chinese (ZH) subsets are evaluated using word error rate (WER) and speaker similarity (SIM). During inference, the prompt text, reference speech, and target text are concatenated along the temporal dimension, and the model autoregressively generates the target speech tokens.

## E Details of Ablation Studies

We provide details of ablation studies as mentioned in Section 4.1

**Common Setup.** Unless otherwise stated, all ablation variants are trained during the pre-training stage without post-training refinement. All models use the same training data and preprocessing pipeline described in Section 3.1, as well as an identical quantizer configuration operating at 1 kbps and the same codec decoder architecture.

For all ablation experiments, we employ a global batch size of 128 and train the models for 200,000 optimization steps. We use the AdamW optimizer with a maximum learning rate of $1 \times 10^{-4}$. All loss terms and their corresponding weights are kept identical for all ablations.

**Shared-Parameter Ablations.** For the shared-parameter ablations, we compare the dual-encoder design of XY-Tokenizer with single-channel variants. In the single-channel setting, the semantic and acoustic encoders are replaced by a single shared encoder, while the quantizer and decoder remain unchanged. To control for model capacity, we additionally evaluate a larger single-channel encoder initialized from Whisper-medium.

**LLM Trainability.** To study the effect of LLM trainability, the semantic decoder is either fixed or jointly optimized during pre-training. In the trainable setting, all LLM parameters are updated together with the codec, while the tokenizer architecture and capacity are kept identical. No other training configurations are changed.

**Model Scaling.** For model scaling experiments, we increase encoder capacity by replacing the

Whisper-small initialization with Whisper-base or Whisper-large, while keeping the dual-encoder architecture unchanged. The quantizer, decoder, and training procedure are identical across all scaled variants.

**ASR Supervision.** For the ASR supervision ablations, all experiments are conducted under the single-channel encoder setting. We consider three configurations that vary encoder initialization and explicit ASR supervision: (1) a model initialized from pretrained Whisper weights and trained with an explicit LLM-based ASR loss; (2) a model trained with the same ASR loss but without Whisper initialization, i.e., trained from scratch; and (3) a model initialized from pretrained Whisper weights but trained without the ASR loss. In all cases, the codec architecture, quantizer configuration, and training schedule are kept identical, and only the presence of Whisper initialization and ASR supervision differs across variants.

# F Additional Ablations

## F.1 Choice of Acoustic Decoder

| | SIM↑ | STOI↑ | PESQ↑ |
|---|---|---|---|
| Vocos | **0.82** | **0.93** | **3.23 / 2.69** |
| HiFi-GAN | 0.81 | 0.92 | 3.07 / 2.55 |
| SEANet | **0.82** | **0.93** | 3.11 / 2.58 |

Table 11: Comparison of various acoustic decoders.

We conducted an ablation study on the acoustic decoder. We trained a 12.5Hz RVQ-8 autoencoder (consisting of only an acoustic encoder, quantizer, and acoustic decoder, with no semantic encoder or decoder, and only the pre-training stage) for 180k steps. We used three models as the acoustic decoder: Vocos (used in XY-Tokenizer), SEANet decoder (Tagliasacchi et al., 2020) (also used in Mimi Codec), and HiFi-GAN vocoder (Kong et al., 2020).

Based on the results detailed in Table 11, Vocos demonstrated superior reconstruction performance compared to both HiFi-GAN and SEANet. This finding validates the effectiveness of our method in generating high-quality speech.

## F.2 Necessity of Two-Stage Training Strategy

We adopt a two-stage training strategy. In the **pre-training stage**, the encoder and quantizer of the XY-Tokenizer are optimized through an ASR task to align their representations with text, while a reconstruction task is employed to capture coarse-grained acoustic features. In the **post-training stage**, we freeze the encoder and quantizer to **preserve the text-token alignment ability** of the XY-Tokenizer, and introduce a discriminator to model fine-grained acoustic information. This design choice is motivated by the following considerations:

| | Enc | Dec | LLM | Disc | TP |
|---|---|---|---|---|---|
| Single-stage | ✓ | ✓ | ✓ | ✓ | 40.8 |
| Pre-training | ✓ | ✓ | ✓ | × | 136.7 |
| Post-training | × | ✓ | × | ✓ | 46.6 |

Table 12: Training efficiency comparison between single-stage and two-stage training strategies. ✓indicate trainable modules. TP denotes training throughput, measured as processed audio seconds per GPU per second.

**Training stability** From our empirical experiments, we found that the RVQGAN structure becomes unstable at low bitrates (approximately $\leq 1.5$ kbps). Removing the discriminator during pre-training significantly improved stability.

**Training efficiency** As shown in Section 3 and Table 12, during the pre-training stage we use a batch size of 4, with each audio padded to 30 seconds (same as Whisper), resulting in 120 seconds of audio per GPU. In the post-training stage, we use a batch size of 16, with each audio clipped to 5 seconds, resulting in 80 seconds of audio per GPU. In both stages, GPU memory utilization is already close to its maximum capacity. We further observe that **incorporating the discriminator in the post-training stage substantially increases computational cost and reduces throughput during training**. If the two stages were merged, the batch size would need to be reduced even further, resulting in much lower training efficiency.

Therefore, we employ the two-stage training strategy to balance stability, efficiency, and modeling capability.

# G Semantic vs. Acoustic Encoder Analysis

## G.1 t-SNE Visualization of Latent Representations

To gain qualitative insights into how different model components encode semantic and acous-

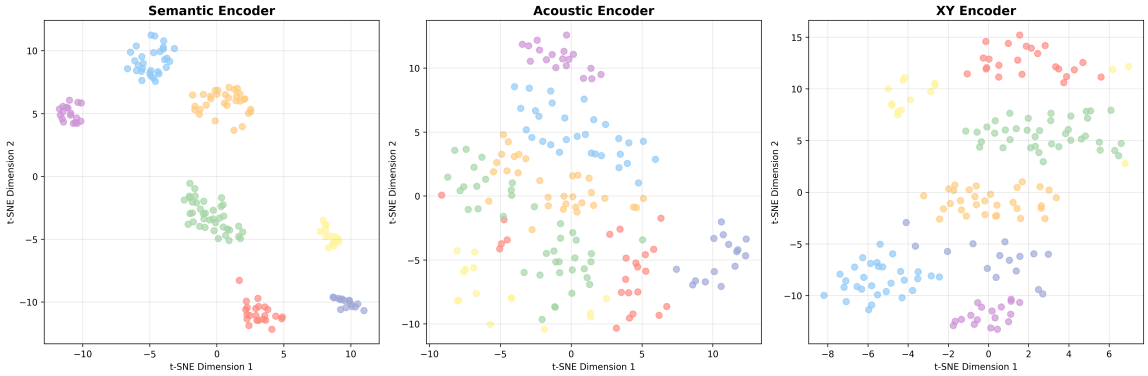


Figure 3: t-SNE visualization of latent representations from the semantic encoder, acoustic encoder, and the quantized outputs of XY-Tokenizer. Each color represents a unique text transcript spoken by multiple different speakers.

tic information, we visualize their latent representations using t-SNE (Maaten and Hinton, 2008). Specifically, we compare the latent spaces produced by the semantic encoder, the acoustic encoder, and the final quantized outputs of XY-Tokenizer. Both semantic encoder and acoustic encoder follow the same configuration as described in Section 4.3.

We randomly select several text transcripts from the AISHELL-2 dataset and collect multiple utterances for each transcript spoken by different speakers. For all models, frame-level representations are mean-pooled along the temporal dimension before applying t-SNE to obtain a two-dimensional visualization. Figure 3 visualizes the resulting latent spaces, with colors indicating different transcripts.

The **semantic encoder** exhibits highly compact and well-separated clusters organized primarily by textual content. Utterances sharing the same transcript are tightly grouped together despite speaker variation, indicating that the semantic encoder is more biased toward preserving content-related information, with reduced sensitivity to acoustic variations such as speaker identity.

In contrast, representations produced by the **acoustic encoder** are considerably more dispersed. Samples corresponding to the same text are scattered across the latent space, while being less structured with respect to textual content compared to the semantic encoder.

Notably, the **XY-Tokenizer** produces latent representations that are more structured than those of the acoustic encoder, while remaining less compact than those of the semantic encoder. Its quantized outputs form discernible text-dependent clusters, yet exhibit increased intra-cluster variance compared to the semantic encoder. This observation suggests that XY-Tokenizer strikes a balance between semantic abstraction and acoustic diversity, yielding representations that retain semantic coherence while preserving relevant acoustic details.

### G.2 Mel-spectrogram Analysis

Figure 4 shows mel-spectrograms of raw speech and the corresponding resynthesized outputs from two speakers uttering the same text at comparable speaking rates. The resynthesized speech from the **semantic encoder** exhibits noticeably smoother spectral patterns, with attenuated high-frequency components and reduced fine-grained spectral detail. Across the two speakers, the resulting spectrograms appear visually similar, suggesting that speaker-dependent acoustic characteristics are deemphasized while coarse structures related to linguistic content are retained.

In contrast, the **acoustic encoder** preserves substantially richer spectral detail across both low- and high-frequency regions. The reconstructed spectrograms maintain clearer harmonic structures and speaker-specific variations, indicating stronger sensitivity to timbre and fine-grained acoustic properties.

## H Use of LLMs

In this work, we used a large language model (LLM) to assist with language polishing and improving the clarity of writing.

## I Related Work

### I.1 Speech Language Models

Recent research on speech large language models has attracted considerable interest (Latif et al.,

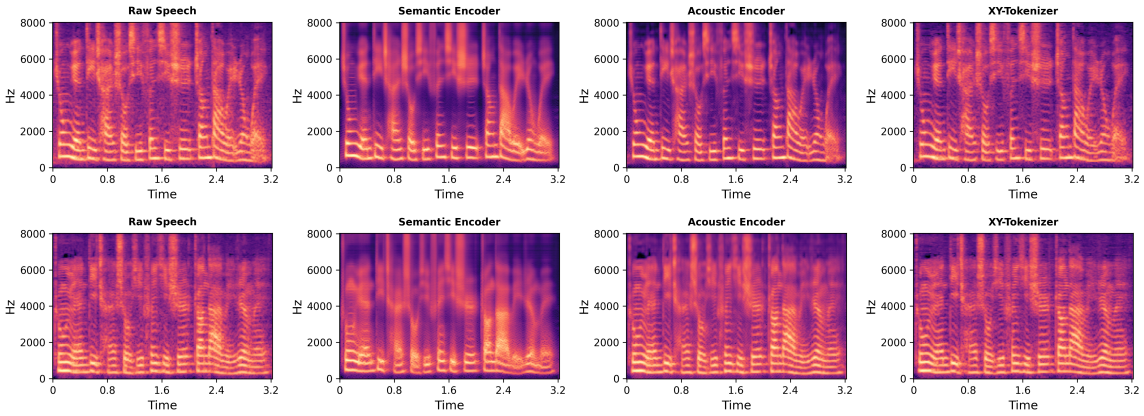


Figure 4: Mel-spectrogram comparison of raw speech and synthesized speech produced by the semantic encoder, acoustic encoder, and XY-Tokenizer. Top: Speaker A. Bottom: Speaker B, who utters the same text as Speaker A.

2023; Wu et al., 2024; Ji et al., 2024a). AudioLM (Borsos et al., 2023) achieves high-quality audio generation with coherent long-term structure through coarse-to-fine token modeling. SpeechGPT (Zhang et al., 2023a), the first end-to-end speech large language model, features strong instruction-following capabilities and effective spoken dialogue interaction, employing a three-stage training methodology to facilitate cross-modal transfer and efficient training. IntrinsicVoice (Zhang et al., 2024) implements GroupFormer to diminish the modality gap between text and speech, thereby enabling the transfer of capabilities from pre-trained large language models to the speech domain, facilitating low-latency and high-quality speech interaction in multi-turn dialogue contexts. Moshi (Défossez et al., 2024) employs a multi-stream architecture that concurrently processes audio streams from both the user and the system (Moshi itself), supporting dynamic conversations with overlaps and interruptions, thereby achieving full-duplex dialogue. Despite their differences, these models critically depend on the quality of speech tokenization, highlighting the importance of speech codecs that balance linguistic structure and acoustic detail (Betker, 2023; Wang et al., 2023; Copet et al., 2023; Wang et al., 2024; Du et al., 2024a).

### I.2 Speech Codecs

Speech codecs play a vital role in speech large language models by converting continuous speech signals into discrete tokens (Barnes et al., 1996; Mentzer et al., 2023; Zhu et al., 2025), enabling LLMs to process speech as a form of "foreign language." Neural network-based speech codecs predominantly utilize the RVQGAN paradigm, which can compress audio signals into low-bitrate representations through end-to-end training (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2023), making them ideal for real-time communication applications (Wu et al., 2023; Parker et al., 2024; Du et al., 2024b; Li et al., 2025; Yang et al., 2025b). BigCodec (Xin et al., 2024) achieves excellent reconstruction quality even at low bitrates by scaling the encoder and decoder parameters. To align speech codec tokens with large text models, recent efforts have explored modeling both semantic and acoustic features simultaneously (Zhang et al., 2023b; Défossez et al., 2024; Ye et al., 2025a). SpeechTokenizer (Zhang et al., 2023b) enhances the RVQGAN paradigm with semantic distillation to guide the first layer of RVQ to align with a teacher SSL model (Hsu et al., 2021). X-Codec (Ye et al., 2025a) proposes an X-shaped structure where each layer of RVQ contains both semantic and acoustic information. Mimi (Défossez et al., 2024) introduces a split RVQ architecture, with one channel distilled from a pretrained SSL model to capture semantic information. However, these approaches often face challenges in balancing semantic alignment and acoustic fidelity, particularly at low bitrates, motivating codec designs that explicitly mitigate the semantic–acoustic conflict.

## J More Comparisons

### J.1 Comparison with DualCodec

We further compare XY-Tokenizer with DualCodec (Li et al., 2025), a representative neural audio codec with a similar architectural design.

| | EN – LibriSpeech | | | | | EN – VoxPopuli | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | WER↓ | SIM↑ | STOI↑ | PESQ-NB↑ | PESQ-WB↑ | WER↓ | SIM↑ | STOI↑ | PESQ-NB↑ | PESQ-WB↑ |
| DualCodec | 0.30 | 0.81 | 0.92 | 3.14 | 2.60 | 0.38 | 0.85 | 0.92 | 3.09 | 2.61 |
| **XY-Tokenizer** | **0.13** | **0.85** | 0.92 | 3.10 | 2.50 | **0.25** | **0.87** | 0.91 | 2.99 | 2.49 |
| | **ZH – AISHELL-2** | | | | | **ZH – CommonVoice** | | | | |
| | CER↓ | SIM↑ | STOI↑ | PESQ-NB↑ | PESQ-WB↑ | CER↓ | SIM↑ | STOI↑ | PESQ-NB↑ | PESQ-WB↑ |
| DualCodec | 0.24 | 0.75 | 0.87 | 2.60 | 2.14 | 0.37 | 0.81 | 0.86 | 2.60 | 2.14 |
| **XY-Tokenizer** | **0.11** | **0.79** | **0.87** | **2.63** | 2.12 | **0.21** | **0.84** | 0.85 | 2.54 | 2.05 |

Table 13: Comparison between XY-Tokenizer and DualCodec on semantic and reconstruction metrics.

| | ZH | | EN | |
|---|---|---|---|---|
| | **SIM↑** | **CER↓** | **SIM↑** | **WER↓** |
| DualCodec-TTS | 0.659 | **0.0167** | 0.568 | 0.0251 |
| **XY-Tokenizer-TTS** | **0.695** | 0.0174 | **0.629** | **0.0181** |

Table 14: Comparison between XY-Tokenizer and DualCodec on LLM-based speech generation on the Seed-TTS-Eval benchmark.

DualCodec consists of a semantic encoder (initialized from w2v-BERT 2.0 and kept frozen) and an acoustic encoder, and also adopts a residual vector quantization (RVQ) scheme. Specifically, RVQ-1 is derived from the frozen SSL encoder, while subsequent layers are obtained by modeling the residual between the codec encoder output and a ResNet-based mapping.

We evaluate both codecs on reconstruction quality and downstream LLM-based TTS tasks. For a fair comparison, we select the DualCodec 12.5Hz (16384+4096) RVQ-6 configuration, which operates at 0.925 kbps. This setting is the closest open-source counterpart to XY-Tokenizer, sharing the same frame rate and a comparable bitrate. All evaluation protocols follow Section 3.

As shown in Table 13, the two methods achieve comparable objective reconstruction metrics at similar bitrates. Notably, XY-Tokenizer yields slightly higher speaker similarity, indicating better preservation of speaker characteristics. Moreover, XY-Tokenizer achieves lower WER/CER in the ASR probing task, suggesting that its speech tokens are better aligned with textual content.

Furthermore, on the LLM-based speech generation task, as shown in Table 14, both methods achieve low and comparable error rates, suggesting that tokens from both codecs are suitable for autoregressive generation. However, XY-Tokenizer-TTS consistently achieves higher speaker similarity (SIM). We hypothesize that this advantage stems from our unified token fusion design (where different RVQ layers follow a direct residual hierarchy without introducing additional transformation modules), which produces stronger inter-layer correlations and simplifies the modeling process for autoregressive decoders. In contrast, DualCodec relies on a ResNet-based residual mapping, which may introduce more complex dependencies between RVQ-1 and subsequent layers, increasing the modeling difficulty.

### J.2 Comparison with S3Tokenizer

| | SIM↑ | STOI↑ | PESQ-NB↑ | PESQ-WB↑ |
|---|---|---|---|---|
| S3Tok (w/o p) | 0.15 | 0.60 | 1.10 | 1.06 |
| S3Tok (w/ p) | **0.88** | 0.64 | 1.95 | 1.62 |
| **XY-Tokenizer** | 0.85 | **0.92** | **3.10** | **2.50** |

Table 15: Comparison between XY-Tokenizer and S3Tokenizer on reconstruction metrics. "w/ p" and "w/o p" denote with and without speaker prompt, respectively.

We further compare XY-Tokenizer with S3Tokenizer used in CosyVoice (Du et al., 2024a). Similar to XY-Tokenizer, S3Tokenizer leverages ASR-based supervision to capture semantic information. However, unlike XY-Tokenizer, it does not incorporate reconstruction supervision during training. As a result, speech synthesis based on S3Tokenizer typically requires additional generation modules (e.g., flow matching) (Lipman et al., 2022) and a vocoder (e.g., HiFi-GAN) (Kong et al., 2020) to reconstruct waveform-level audio.

We conduct a comparison on the LibriSpeech test-clean set, and the results are shown in Table 15. Compared to S3Tokenizer with speaker prompts, XY-Tokenizer achieves comparable speaker similarity while consistently outperforming it on STOI,

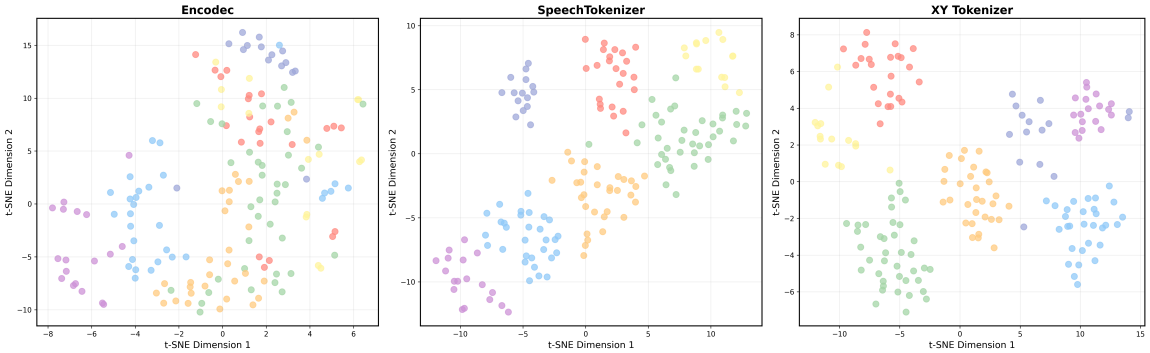


Figure 5: t-SNE visualization of latent representations from XY-Tokenizer, SpeechTokenizer, and Encodec. Each color represents a unique text transcript spoken by multiple different speakers.

PESQ-NB, and PESQ-WB, indicating better intelligibility and perceptual quality. Without speaker prompts, S3Tokenizer exhibits significantly lower speaker similarity to the input audio. We hypothesize that this is because S3Tokenizer tokens discard a substantial amount of paralinguistic information, such as timbre and prosody, making it difficult to faithfully reconstruct speech without external speaker conditioning.

In contrast, XY-Tokenizer jointly models both semantic and acoustic information. While maintaining good alignment with textual content, it is also capable of near-lossless reconstruction of the input audio. These observations indicate that XY-Tokenizer is more suitable for end-to-end speech generation, where high-fidelity speech tokens can be directly predicted from text and then decoded into waveform using a codec decoder. On the other hand, S3Tokenizer serves more as an intermediate representation and is better suited for a two-stage generation paradigm, where text is first used to predict S3 tokens, followed by an additional generative model (e.g., flow matching) that predicts high-quality acoustic representations (e.g., mel-spectrograms or VAE latents) (Zhang et al., 2025), which are finally synthesized into waveform using a vocoder or decoder.

### J.3 Comparison with Encodec and SpeechTokenizer

Following the analysis in Appendix G.1, we visualize the latent representations of XY-Tokenizer, Encodec, and SpeechTokenizer using t-SNE. The results are shown in Figure 5. Both XY-Tokenizer and SpeechTokenizer exhibit highly compact and well-separated clusters that are primarily organized by textual content. Utterances sharing the same transcript are tightly grouped together despite variations in speaker identity, indicating that both distillation-based and ASR-based supervision can effectively align the tokenizer latent space with linguistic content.

In contrast, the latent space of Encodec is considerably more dispersed. Samples corresponding to the same text are scattered across the space, and the overall structure is less organized with respect to textual content compared to XY-Tokenizer and SpeechTokenizer. This indicates that linguistic structure is less explicitly reflected in the latent space of Encodec.