# STABLETOKEN: A NOISE-ROBUST SEMANTIC SPEECH TOKENIZER FOR RESILIENT SPEECHLLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Prevalent semantic speech tokenizers, designed to capture linguistic content, are surprisingly fragile. We find they are not robust to meaning-irrelevant acoustic perturbations; even at high Signal-to-Noise Ratios (SNRs) where speech is perfectly intelligible, their output token sequences can change drastically, increasing the learning burden for downstream LLMs. This instability stems from two flaws: a brittle single-path quantization architecture and a distant training signal indifferent to intermediate token stability. To address this, we introduce StableToken, a tokenizer that achieves stability through a consensus-driven mechanism. Its multi-branch architecture processes audio in parallel, and these representations are merged via a powerful bit-wise voting mechanism to form a single, stable token sequence. StableToken sets a new state-of-the-art in token stability, drastically reducing Unit Edit Distance (UED) under diverse noise conditions. This foundational stability translates directly to downstream benefits, significantly improving the robustness of SpeechLLMs on a variety of tasks.

## 1 INTRODUCTION

The application of Large Language Models (LLMs) to the speech domain has given rise to a new class of powerful models: Speech Large Language Models (SpeechLLMs) (Hurst et al., 2024; Défossez et al., 2024; Zeng et al., 2024). These models rely on a discrete speech tokenizer to convert continuous audio into tokens sequences that the LLM can process. Among available methods, semantic tokenizers have been widely adopted, as their low-bitrate, semantically-aligned outputs are highly compatible with LLM architectures (Défossez et al., 2024; Zeng et al., 2024; Ding et al., 2025; Wu et al., 2025).

The design of semantic speech tokenizers has evolved from early self-supervised learning (SSL) methods (Hsu et al., 2021; Baevski et al., 2020) towards a more direct, supervised paradigm (Du et al., 2024a;b; Zeng et al., 2024). This modern paradigm centers on optimizing a VQ-based quantizer (Van Den Oord et al., 2017) with a direct, end-to-end objective such as automatic speech recognition (ASR) . This powerful combination has proven highly effective at producing semantically-rich and compact discrete representations, leading to the widespread adoption of supervised semantic tokenizers as the backbone of many modern SpeechLLMs (Zeng et al., 2024; Ding et al., 2025; Wu et al., 2025; Huang et al., 2025; Fang et al., 2025).

Despite their widespread adoption and apparent success, we find that these semantic tokenizers harbor a critical vulnerability: a profound lack of robustness. Contrary to their core design principle of encoding semantics, even imperceptible acoustic noise can induce drastic shifts in their discrete outputs (Figure 1). The general issue of tokenizer instability is also supported by recent findings on earlier non-VQ-based SSL tokenizers (Messica & Adi, 2024). This instability creates a damaging downstream effect: small acoustic changes trigger large token jumps, which break the crucial speech–text alignment and pose a significant modeling challenge for the LLM, forcing it to learn from an inconsistent or even chaotic input stream. This inherent fragility is severely amplified by environmental noise, serving as a key cause for the performance degradation of SpeechLLMs in real-world conditions (Ma et al., 2025b; Zhang et al., 2025; Yang et al., 2024c; Jiang et al., 2025). We argue that enhancing tokenizer robustness is therefore a direct and promising path toward building more resilient models.

We pinpoint the fragility of semantic tokenizers to two fundamental weaknesses. First, an **architectural flaw**: these tokenizers rely on single-path quantization, a design that lacks fault tolerance. A

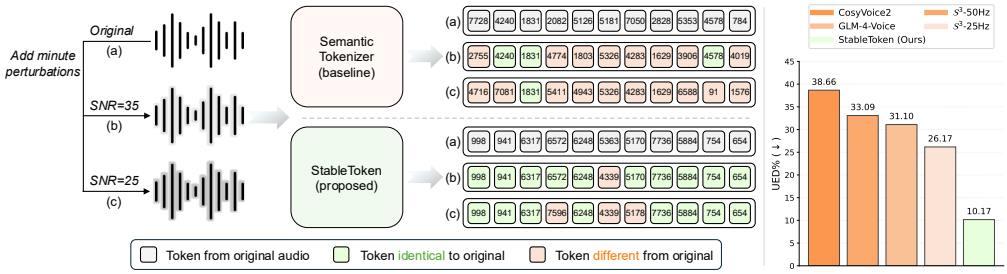


Figure 1: Illustration of StableToken: unlike traditional methods, StableToken yields consistent token sequences under minute perturbations with different Signal-to-Noise Ratios (SNRs). Robustness is measured by Unit Edit Distance (UED, ↓) between token sequences of the original and noise-perturbed audio. StableToken achieves significantly lower UED, indicating enhanced token stability.

minor perturbation near a quantization boundary is inevitably magnified into a completely different output token. This architectural vulnerability is then compounded by a **distant supervisory signal**: the standard ASR loss is indifferent to this intermediate token instability, as it only supervises the final transcribed text. This allows models to converge on solutions that are functionally correct but representationally fragile. The dual challenge of a brittle architecture and a distant supervisory signal necessitates a new tokenization paradigm.

Addressing this dual challenge is non-trivial. For the brittle architecture, an offline ensemble of models seems intuitive. However, this approach is untenable: (1) it prohibitively increases inference cost; (2) aggregating independently trained models is non-trivial, as their quantization boundaries are arbitrarily aligned; and (3) a token-level majority vote is too coarse. To tackle the distant supervisory signal, one might introduce a token-level consistency objective for clean and noisy input audios. Yet, this leads to notoriously unstable gradients when applied to discrete codes, making the model difficult to train. The failure of these straightforward approaches underscores the need for a more integrated paradigm.

We propose StableToken, which integrates a co-designed architecture and training strategy to overcome the dual challenges of architectural fragility and distant supervision. **Architecturally**, it introduces the voting-LFQ module—a multi-branch quantizer extended from the LFQ algorithm (Yu et al., 2023), with negligible inference overhead. Its core mechanism is a differentiable bit-level majority vote. During training, this enables a more fine-grained fusion of multi-branch information, leading to more stable and robust representation learning. At inference, this same mechanism provides profound error-correction, operating at the granular bit-level rather than the coarse token-level. This distinction is critical: not only does it ensure the final token remains correct when a minority of branches err due to noise, but it can even recover the token when a majority of branches fail at the token-level, as long as the underlying bit-level errors remain sparse.

This architectural robustness is further solidified by a tailored **training strategy**. We present the model with multiple "views" of an input—a clean version to a majority of branches and a perturbed version to a random minority—to create a stable reference. A consensus loss then leverages this reference to provide the explicit, intermediate supervision. The multi-branch architecture and multi-view training strategy are thus deeply intertwined: the architecture provides the necessary structure for the training signal, and the signal in turn unlocks the architecture's full potential.

We validate StableToken through comprehensive experiments. At the tokenizer level, it achieves a new state-of-the-art in **noise robustness**, slashing the Unit Edit Distance (UED) by over 60% relative (from 26.17% to 10.17%), all while maintaining top-tier **reconstruction fidelity**. This foundational superiority translates directly to **downstream SpeechLLMs**. In speech understanding, the downstream models yield significant robustness gains that are especially pronounced under severe noise. The performance gap between StableToken and baselines widens dramatically as the noise level increases. Similarly, for speech generation, the enhanced token consistency simplifies the learning task, resulting in substantially superior synthesis quality of downstream models. These results confirm that improving tokenizer robustness is directly and highly effective for building more resilient speechLLMs.

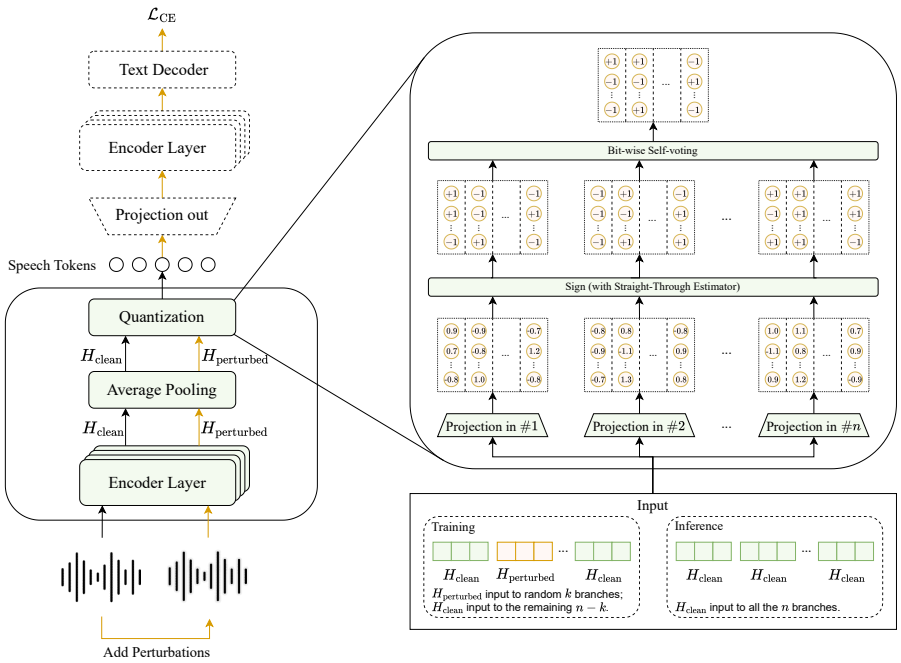


Figure 2: The architecture of StableToken. Our model replaces the standard single-path quantizer with a multi-branch **Voting-LFQ module**. The zoomed-in view shows $n$ parallel branches generating independent binary representations. A bit-wise majority vote then aggregates them into a single token. During our **Noise-Aware Consensus Training**, a randomly selected minority of branches receive perturbed inputs ($H_{\text{perturbed}}$), while the majority receive clean inputs ($H_{\text{clean}}$). A consensus loss forces the perturbed branches to align with the consensus. The yellow paths are only used during training.

## 2 METHODS

### 2.1 OVERALL STRUCTURE

Our approach, StableToken, is designed to overcome the fragility of prevailing VQ-based semantic tokenizers. These tokenizers often produce unstable token sequences in the presence of subtle noise, a vulnerability stemming from two core weaknesses: (1) a single-path architecture that lacks fault tolerance, and (2) a distant supervisory signal that fail to enforce representational invariance.

StableToken adopts the architectural paradigm, established in works like (Du et al., 2024a; Zeng et al., 2024; Du et al., 2024b) of embedding a semantic tokenizer within an end-to-end ASR model. However, our approach fundamentally enhances this design by introducing two synergistic innovations to address its inherent instabilities: (1) the Voting-LFQ Module, a multi-branch quantizer that builds in architectural robustness, and (2) Noise-Aware Consensus Training, a training strategy that explicitly enforces invariance to acoustic perturbations.

### 2.2 THE VOTING-LFQ MODULE

The foundation of StableToken is a novel quantizer architecture designed for intrinsic robustness. As shown in Figure 2, the pretrained speech encoder first processes the input speech into a sequence of hidden states. These states are then downsampled via average pooling to produce a compact representation, $\mathbf{h} \in \mathbb{R}^D$, for each time step.

While traditional quantizers map $\mathbf{h}$ to a token in a single, brittle step, our **Voting Look-up-Free Quantizer (Voting-LFQ)** is founded on redundancy and consensus. It begins by creating $n$ independent "perspectives" of the input state $\mathbf{h}$ using $n$ parallel linear projection layers. Each branch $i \in \{1, \ldots, n\}$ computes a projected vector $\mathbf{p}_i \in \mathbb{R}^d$:

$$\mathbf{p}_i = W_i\mathbf{h} + \mathbf{b}_i, \tag{1}$$

where $W_i \in \mathbb{R}^{d \times D}$ and $\mathbf{b}_i \in \mathbb{R}^d$ are the unique learnable parameters for that branch. Each projected vector is then binarized into $\mathbf{B}_i \in \{-1, +1\}^d$ using the non-differentiable sign function, i.e., $\mathbf{B}_i = \text{sign}(\mathbf{p}_i)$. We use the Straight-Through Estimator (STE) (Bengio et al., 2013) to enable end-to-end training.

During **training**, we aggregate these $n$ binary vectors in a bit-wise manner by averaging their values across branches for every dimension $j \in \{1, \ldots, d\}$, resulting in a real-valued score:

$$(\mathbf{s}_{\text{final}})_j = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{B}_i)_j. \tag{2}$$

Unlike the rigid assignment of a single bit ($+1$ or $-1$), these averaged scores can take nuanced values representing the confidence or consensus of all branches. This provides the model with richer feedback during optimization, helping it learn more robust and informative representations.

During **inference**, we perform one additional step to apply the sign function to these aggregated scores to obtain the final consensus-based binary vector:

$$(\mathbf{B}_{\text{final}})_j = \text{sign}\left((\mathbf{s}_{\text{final}})_j\right). \tag{3}$$

By using an odd number of parallel branches ($n$), we enforce a strict majority rule via a bit-wise vote, creating exceptional robustness against noise. This approach not only corrects errors when a minority of branches fail, but can also recover the true token even if a majority of branches are corrupted at the token-level. Recovery is possible as long as the underlying bit-level errors remain sparse. This resilience marks a significant advantage over fragile single-path quantizers and, in parallel, allows for more expressive representations during training.

Finally, by mapping its $-1$ and $+1$ entries to 0 and 1 respectively and treating the $\{0, 1\}$ representation as a binary number, the stabilized binary vector $\mathbf{B}_{\text{final}}$ is deterministically mapped to an integer index $k \in \{0, \ldots, 2^d - 1\}$. This index serves as the final, robust speech token. It is worth noting that our Voting-LFQ structure introduces negligible additional parameters and computational overhead during inference. A detailed complexity analysis is provided in Appendix B.6.

### 2.3 Noise-Aware Consensus Training

The Voting-LFQ architecture enables our novel training paradigm, designed to explicitly instill representational invariance. Our goal is to make the tokenizer robust to noise without degrading its performance on clean inputs.

The core mechanism works as follows: during each forward pass, for a given input audio $\mathbf{w}$, we generate a perturbed audio sample $\mathbf{w}' = \mathcal{A}(\mathbf{w})$, where $\mathcal{A}(\cdot)$ is a stochastic augmentation function applied at the waveform level (e.g., adding Gaussian noise; further details in Appendix B.3). Both $\mathbf{w}$ and $\mathbf{w}'$ are separately processed by the encoder to produce two corresponding hidden states, $\mathbf{h}$ and $\mathbf{h}'$. we then randomly select a minority subset of $k$ branches (where $k < n/2$) to receive the perturbed hidden state $\mathbf{h}'$, while the remaining $n - k$ majority branches receive the clean hidden state $\mathbf{h}$.

This setup allows the model to perform self-stabilization. To enforce this, we introduce the **consensus loss** ($\mathcal{L}_{\text{consensus}}$) which encourages all branches, whether they see a clean or noisy input, to produce similar pre-quantization representations. We compute a dynamic, "online" target $\bar{\mathbf{p}}_{\text{all}}$ by averaging the pre-quantization vectors $\mathbf{p}_i$ from **all** $n$ branches. The loss then penalizes the deviation of each branch from this global average:

$$\mathcal{L}_{\text{consensus}} = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{p}_i - \bar{\mathbf{p}}_{\text{all}}\|_2^2, \quad \text{where} \quad \bar{\mathbf{p}}_{\text{all}} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{p}_j \ . \tag{4}$$

By optimizing this objective, the clean-majority branches act as a stable anchor for the global average $\bar{\mathbf{p}}_{\text{all}}$, preventing it from being corrupted by the noisy inputs. Consequently, the noisy-minority branches are forced to learn representations that align with the clean consensus, effectively learning to ignore the perturbations. Optimizing on the continuous vectors $\mathbf{p}_i$ provides a smoother and more effective gradient signal than working with the binarized $\mathbf{b}_i$.

### 2.4 Final Training Objective

The complete training objective for StableToken combines the ASR task loss with our consensus loss and standard LFQ regularization terms. The primary task is optimized via a Cross-Entropy loss ($\mathcal{L}_{\text{ASR}}$) on the ground-truth transcripts. Following the LFQ framework (Yu et al., 2023), we also include a **commitment loss** ($\mathcal{L}_{\text{commitment}}$) to encourage the hidden states to stay close to the quantized representations, and a **codebook entropy loss** ($\mathcal{L}_{\text{codebook}}$) to promote uniform usage of the discrete codes. The final, composite loss function is a weighted sum:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ASR}} + \lambda_1 \mathcal{L}_{\text{consensus}} + \lambda_2 \mathcal{L}_{\text{commitment}} + \lambda_3 \mathcal{L}_{\text{codebook}}, \tag{5}$$

where $\lambda_1, \lambda_2$, and $\lambda_3$ are scalar hyperparameters that balance the influence of each component.

## 3 Experimental Setup

Our experiments are structured to comprehensively validate StableToken from three perspectives. We first demonstrate its core superiority at the **tokenizer level**, establishing a new state-of-the-art in noise robustness without compromising reconstruction quality (§4.1). We then show that this fundamental stability translates directly into significant performance gains in **diverse downstream SpeechLLM tasks**, including ASR, Speech Emotion Recognition (SER), and Text-to-speech (TTS) (§4.2). Finally, we dissect the model through **ablation studies, qualitative analysis, and a case study** to verify the contribution of each design component and provide insight into its inner workings (§4.3).

**Tokenizer Training.** Our StableToken model is built upon an encoder-decoder architecture initialized from `whisper-large-v3` (Radford et al., 2023), with our Voting-LFQ module inserted into the encoder's mid-point. The tokenizer is pre-trained on a diverse 150k-hour speech corpus. Our tokenizer vocabulary size is set to 8192 (corresponding to $d = 13$), and the frame rate is 25Hz. For our main experiments, the number of voters is set to $N = 5$, a choice justified by our analysis in Section 4.3. Full details on training data hyperparameters are provided in Appendix B.

**Baseline Models.** We benchmark StableToken against a comprehensive suite of SOTA models across three categories: SSL-based, distilled, and supervised tokenizers. For all baselines, we use their officially released models. The detailed list of baseline models can be found in Appendix D.

**Tokenizer-Level Evaluation.** We assess the tokenizer's intrinsic properties. **Robustness** is measured by Unit Edit Distance (UED%, ↓) (Messica & Adi, 2024) on the FLEURS (Conneau et al., 2023) benchmark under various synthetic perturbations and real-world noise conditions, which include challenging out-of-domain (OOD) noise. A detailed description of the noise profiles is in Appendix E. **Fidelity** is measured by Word Error Rate (WER%, ↓) and Mean Opinion Score (MOS, ↑) on the LibriSpeech (Panayotov et al., 2015) and SEED (Anastassiou et al., 2024) benchmarks.

**Downstream Task Evaluation.** For downstream tasks, we follow a controlled, isogenic setup to ensure fair comparison. Each tokenizer is integrated into a SpeechLLM framework using a pre-trained `Qwen2.5-3B` (Yang et al., 2024a) backbone, which is then fine-tuned using a prompt-based paradigm (Zeng et al., 2025). We evaluate on three tasks: (1) **ASR:** Assessed on noise-augmented LibriSpeech (Panayotov et al., 2015) and the CHiME-4 (Vincent et al., 2017) benchmark using WER (%); (2) **SER:** Assessed on a noise-augmented version of the ESD (Zhou et al., 2022) test set using classification accuracy (%); (3) **TTS:** Assessed on the SEED-TTS (Anastassiou et al., 2024) benchmark using both WER (%) and MOS. The aggregated training datasets, fine-tuning hyperparameters, and prompts for each task are detailed in Appendix F.

## 4 Results

### 4.1 Tokenizer-Level Performance

#### 4.1.1 Superior Noise Robustness

As shown in Table 1, StableToken establishes a new state-of-the-art in noise robustness. It achieves an average UED of **10.17%**, a dramatic improvement over both the best supervised baseline ($\mathcal{S}^3$

Table 1: Noise robustness comparison across different semantic tokenizers. Results are reported in UED% (↓) under synthetic perturbation (Gaussian, Pink, Brown, Bit Crush) and real noise conditions. It is worth noting that a comparison is most meaningful between tokenizers of the same type. For a more comprehensive evaluation, we also include SSL and semantic distilled tokenizers as baselines.

| Model | #C | Frame Rate | Codebook Size | Gauss. Noise | Pink Noise | Brown Noise | Bit Crush | Real Noise | Real (OOD) | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| **SSL Semantic Tokenizer** | | | | | | | | | | |
| HuBERT-500 (Hsu et al., 2021) | 1 | 50Hz | 500 | 26.42 | 20.38 | 18.82 | 18.02 | 18.48 | 19.18 | 20.22 |
| NAST (Messica & Adi, 2024) | 1 | 50Hz | 200 | 18.67 | 15.78 | 15.26 | 14.95 | 18.69 | 19.07 | 17.07 |
| R-Spin (Chang & Glass, 2024) | 1 | 50Hz | 2048 | 21.56 | 17.08 | 15.47 | 14.95 | 15.08 | 14.75 | 16.48 |
| **Semantic Distilled Tokenizer** | | | | | | | | | | |
| SpeechTokenizer (Zhang et al., 2023) | 1 | 50Hz | 1024 | 37.39 | 28.05 | 28.06 | 21.38 | 22.33 | 23.09 | 26.72 |
| | 3 | 50Hz | 1024 | 55.69 | 54.90 | 59.84 | 35.29 | 33.16 | 33.67 | 45.43 |
| | 8 | 50Hz | 1024 | 72.74 | 72.72 | 75.91 | 54.01 | 48.43 | 48.63 | 62.07 |
| X-Codec (Ye et al., 2025) | 1 | 50Hz | 1024 | 53.54 | 43.85 | 40.17 | 36.95 | 27.82 | 28.78 | 38.52 |
| | 3 | 50Hz | 1024 | 71.76 | 59.95 | 57.88 | 50.26 | 41.25 | 42.44 | 53.92 |
| | 8 | 50Hz | 1024 | 84.46 | 77.31 | 76.49 | 68.47 | 59.89 | 62.28 | 71.48 |
| Mimi (Défossez et al., 2024) | 8 | 12.5Hz | 2048 | 72.68 | 59.82 | 60.19 | 43.58 | 41.66 | 42.62 | 53.43 |
| **Supervised Semantic Tokenizer** | | | | | | | | | | |
| GLM-4-Voice-Token. (Zeng et al., 2025) | 1 | 12.5Hz | 16384 | 42.44 | 32.12 | 30.22 | 25.53 | 27.67 | 28.62 | 31.10 |
| $\mathcal{S}^3$ Tokenizer (Du et al., 2024a) | 1 | 25Hz | 4096 | 35.40 | 27.09 | 25.45 | 20.64 | 23.88 | 24.58 | 26.17 |
| | 1 | 50Hz | 4096 | 46.05 | 35.90 | 33.46 | 27.20 | 27.70 | 28.21 | 33.09 |
| CosyVoice2 (Du et al., 2024b) | 1 | 25Hz | 6561 | 54.67 | 42.57 | 39.96 | 30.87 | 31.76 | 32.13 | 38.66 |
| **StableToken (Ours)** | 1 | 25Hz | 8192 | **12.93** | **9.76** | **9.37** | **7.32** | **10.65** | **10.96** | **10.17** |

Tokenizer, 26.17%) and the top-performing robust SSL-based model (R-Spin, 16.48%). Crucially, this strong performance holds even on out-of-distribution (OOD) real-world noise not seen during training, demonstrating the excellent generalization of our method. Furthermore, this outperformance is achieved using a significantly larger vocabulary than conventional tokenizers. This makes the result even more significant, as a larger vocabulary creates a finer-grained decision space, making the task of maintaining token-level invariance inherently more challenging. This substantial performance gap underscores the effectiveness of our co-designed architecture and training strategy.

#### 4.1.2 Excellent Reconstruction Quality

Table 2: Reconstruction results measured by WER (↓) and MOS (↑) on LibriSpeech (Panayotov et al., 2015) and SEED (Anastassiou et al., 2024) benchmarks.

| Model | #C | Frame Rate | BPS | WER ↓ | | | | MOS ↑ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | LS-clean | LS-other | SEED en | SEED zh | LS-clean | LS-other | SEED en | SEED zh |
| GLM-4-Voice-Token. (Zeng et al., 2025) | 1 | 12.5Hz | 175 | 4.04 | 9.33 | 3.54 | 3.23 | 4.07 | **3.99** | **4.16** | 4.10 |
| $\mathcal{S}^3$ Tokenizer (Du et al., 2024a) | 1 | 25Hz | 300 | 5.78 | 13.38 | 5.91 | 4.26 | 3.40 | 3.31 | 3.40 | 3.31 |
| CosyVoice2 (Du et al., 2024b) | 1 | 25Hz | 325 | 4.25 | 9.68 | 4.34 | 2.75 | 3.36 | 3.25 | 3.31 | 3.58 |
| **StableToken (Ours)** | 1 | 25Hz | 325 | **3.84** | **7.99** | **3.44** | **2.62** | **4.09** | 3.83 | 4.01 | **4.18** |

To evaluate reconstruction quality, we follow the methodology of Du et al. (2024a;b); Zeng et al. (2024) and train a flow matching model to synthesize audio from our speech tokens. The results, shown in Table 2, demonstrate that the leap in noise robustness does not compromise the tokenizer's fundamental quality. StableToken delivers state-of-the-art reconstruction performance, evidenced by its exceptional Word Error Rate (WER) and Mean Opinion Scores (MOS). These results validate StableToken as a versatile tokenizer that excels in both resilience and fidelity. Details on the audio reconstruction setup are provided in Appendix B.5.

### 4.2 Downstream SpeechLLM Performance

The ultimate measure of a tokenizer's utility is its impact on downstream tasks. We find that StableToken's intrinsic robustness consistently translates to superior performance in ASR, SER, and TTS, especially in challenging, noisy conditions.

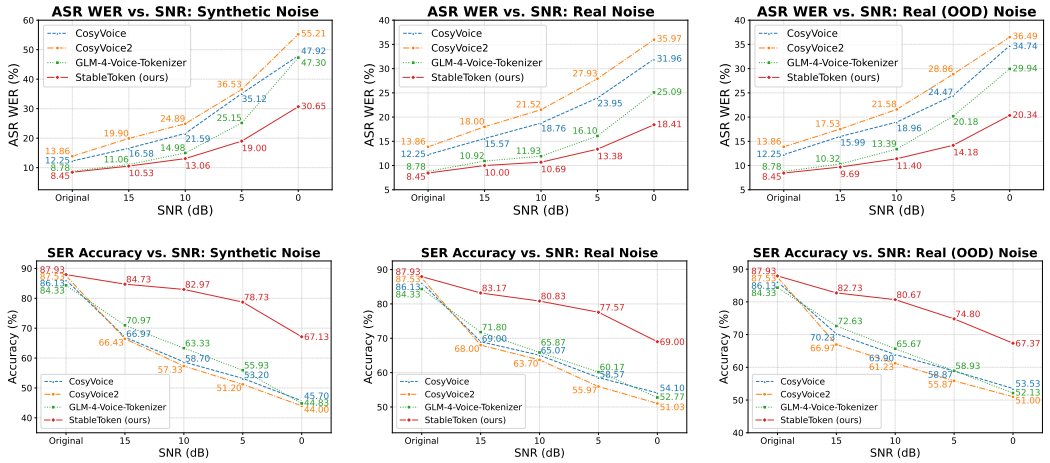


Figure 3: Performance of downstream SpeechLLMs under various noise conditions and SNR levels. **(Top Row)** ASR performance, measured in Word Error Rate (WER, ↓). **(Bottom Row)** SER performance, measured in Accuracy (↑). In both tasks, StableToken consistently demonstrates superior robustness, with the performance gap widening as noise severity increases.

Table 3: Downstream SpeechLLMs performance comparison on ASR (CHiME-4) and TTS (SEED-TTS) benchmarks. In both tasks, integrating StableToken into the downstream SpeechLLM leads to substantially improved performance, demonstrating its noise robustness and versatility.

| Tokenizer | LLM-base | ASR | | | | TTS | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Dev Set | | Test Set | | $\text{SEED-TTS}_{\text{EN}}$ | | $\text{SEED-TTS}_{\text{ZH}}$ | |
| | | Real | Sim. | Real | Sim. | WER ↓ | MOS ↑ | WER ↓ | MOS ↑ |
| CosyVoice | Qwen2.5-3B | 38.66 | 40.82 | 54.63 | 47.71 | 7.80 | 3.52 | 8.73 | 3.47 |
| CosyVoice2 | Qwen2.5-3B | 43.91 | 48.39 | 59.83 | 55.01 | 7.22 | 3.75 | 9.89 | 3.37 |
| GLM-4-Voice | Qwen2.5-3B | 36.92 | 36.38 | 51.08 | 43.09 | 6.19 | **4.19** | 5.26 | 3.85 |
| StableToken | Qwen2.5-3B | **25.56** | **25.36** | **35.90** | **30.61** | **4.43** | 4.12 | **3.02** | **4.08** |

#### 4.2.1 Robust ASR Performance

StableToken significantly contributes to a robust downstream ASR model. Figure 3 shows that while all systems perform comparably on clean audio, the performance gap widens dramatically as noise increases. Under the most severe OOD real-world noise at 0dB SNR, the model with StableToken achieves a WER of 20.34%, a relative reduction of over 30% compared to the baseline's 29.94%.

This robustness generalizes to complex acoustic scenes. On the CHiME-4 (Vincent et al., 2017) benchmark (Table 3), the StableToken-based system achieves WERs of 35.90% (test-real set) and 30.61% (test-simulated set), marking relative reductions of approximately 30% over the next-best baseline. This confirms that token-level stability is a direct driver of downstream model resilience.

#### 4.2.2 Robust SER Performance

Figure 3 shows that the StableToken-based model consistently achieves higher classification accuracy across all noise types and levels. While performance is similar across all tokenizers on clean audio, the StableToken-based model's accuracy degrades much more slowly as noise increases, demonstrating greater robustness in isolating emotional cues from corrupted audio.

#### 4.2.3 Superior TTS Performance

As shown in Table 3, StableToken delivers superior TTS performance, with significantly lower WER on both subsets, as well as an improved MOS on $\text{SEED-TTS}_{\text{ZH}}$ and competitive performance on

SEED-TTS$_{\text{EN}}$. The reduction in WER confirms that our tokenizer enables more intelligible speech synthesis that faithfully reproduces the intended text. Concurrently, the MOS score demonstrates high naturalness and auditory quality of the synthetic speech. Together, these results strongly support StableToken as an effective and information-rich representation for speech synthesis.

In summary, across understanding (ASR, SER) and generation (TTS), StableToken consistently enables stronger, more reliable downstream performance. This dual advantage affirms its effectiveness as a powerful, versatile foundation for real-world speech systems.

## 4.3 Analysis

### 4.3.1 Component Ablation Study

Table 4: Sequential ablation study of StableToken. We jointly evaluate tokenizer robustness (UED ↓) and semantic preservation (ASR WER ↓). Results show that each component contributes to robustness, with the full model providing optimal stability and semantic fidelity. ASR is measured on the validation set during tokenizer training.

| Model Configuration | Tokenizer Robustness (UED% ↓) | | | | | | ASR (WER% ↓) | |
|---|---|---|---|---|---|---|---|---|
| | Gauss. Noise | Brown Noise | Pink Noise | Bit Crush | Real Noise | Real (OOD) | LS-Clean | LS-Other |
| **StableToken (Full)** | **12.93** | **9.76** | **9.37** | **7.32** | **10.65** | **10.96** | **2.03** | **4.68** |
| ↰ *w/o Consensus Loss* | 24.80 | 19.06 | 17.81 | 14.03 | 16.97 | 17.43 | **2.03** | 4.88 |
| ↰ *w/o Noise-Aware Training* | 30.77 | 23.05 | 21.30 | 17.32 | 20.95 | 21.51 | 2.19 | 5.52 |
| ↰ *w/o Multi-Branch* | 34.53 | 25.44 | 24.58 | 19.83 | 23.68 | 24.47 | 2.39 | 5.85 |

Our sequential ablation study, presented in Table 4, confirms that each component of StableToken is critical for its performance. First, removing the *Consensus Loss* causes the significant degradation in token robustness (e.g., UED on Real OOD noise increases from 10.96% to 17.43%), which underscores the importance of enforcing explicit agreement between branches. Subsequently, removing the *Noise-Aware Training* further harms performance, particularly the preservation of semantic content (WER on LibriSpeech-Other increases from 4.88% to 5.52%). Finally, reverting to a single-branch baseline results in the poorest performance overall. This highlights the *Multi-Branch architecture*'s dual role: it is the structural enabler for our training strategy and acts as an effective ensemble at inference, mitigating quantization errors common in single-path designs (Ma et al., 2025a).

### 4.3.2 Analysis of Voter Count ($N$)

To determine the optimal number of voters that best balances performance and computational cost, we conduct preliminary training runs for each configuration. The results in Table 5 show a clear trend: increasing $N$ from 3 to 5 yields substantial improvements in both robustness and semantic preservation. However, a further increase to $N = 7$ offers only marginal gains that do not justify the added computational overhead. We therefore select $N = 5$ as the optimal configuration for all experiments. Analysis of parameters and FLOPs for different $N$ can be found in Appendix B.6.

Table 5: Impact of the number of voters ($N$) on tokenizer robustness and semantic preservation.

| Number of Voters ($N$) | Tokenizer Robustness (UED% ↓) | | | | | | ASR (WER% ↓) | |
|---|---|---|---|---|---|---|---|---|
| | Gauss. Noise | Brown Noise | Pink Noise | Bit Crush | Real Noise | Real (OOD) | LS-Clean | LS-Other |
| $N = 3$ | 20.66 | 15.42 | 14.44 | 11.55 | 14.89 | 15.27 | 2.24 | 5.47 |
| $N = 5$ | 18.68 | 13.87 | 13.11 | 10.50 | 14.06 | 14.49 | 2.22 | 5.38 |
| $N = 7$ | 18.10 | 13.35 | 12.51 | 9.84 | 13.79 | 14.11 | 2.36 | 5.52 |

Table 6: Case study on error correction via bit-wise voting.

| Output Source | Token @ Pos. 68 *(Vote on Bit #4)* | Token @ Pos. 80 *(Vote on Bit #5, #7)* | Token @ Pos. 105 *(Vote on Bit #3)* | Token @ Pos. 114 *(Vote on Bit #2, #6)* |
|---|---|---|---|---|
| Clean Reference | 5517<br>...10001101 | 3485<br>...10011101 | 2920<br>...01101000 | 6939<br>...00011011 |
| Voter 1 (Noisy) | 5533<br>...10011101 | 3485<br>...10011101 | 2920<br>...01101000 | 6939<br>...00011011 |
| Voter 2 (Noisy) | 5517<br>...10001101 | 3517<br>...10111101 | 2912<br>...01100000 | 6943<br>...00011111 |
| Voter 3 (Noisy) | 5517<br>...10001101 | 3517<br>...10111101 | 2920<br>...01101000 | 6939<br>...00011011 |
| Voter 4 (Noisy) | 5517<br>...10001101 | 3485<br>...10011101 | 2920<br>...01101000 | 7003<br>...01011011 |
| Voter 5 (Noisy) | 5533<br>...10011101 | 3357<br>...00011101 | 2920<br>...01101000 | 6939<br>...00011011 |
| Final Voted Output | 5517<br>Bit #4: 3 vs 2 → 0 | 3485<br>Bit #5: 3 vs 2 → 0<br>Bit #7: 4 vs 1 → 1 | 2920<br>4 vs 1 → 1 | 6939<br>Bit #2: 4 vs 1 → 0<br>Bit #6: 4 vs 1 → 0 |

### 4.3.3 Case Study

Table 6 provides a case study illustrating the error correction capability of the voting-LFQ module. For instance, at position 80, noise causes three voters to generate erroneous tokens. Specifically, Voters 2 and 3 flip bit #5, while Voter 5 flips bit #7. Despite **most voters** predicting incorrect tokens, the voting mechanism operating at the bit level allows for correct recovery. For bit #5, the correct value '0' wins by a 3-to-2 majority, and for bit #7, the correct value '1' wins by a 4-to-1 majority, successfully reconstructing the original token (3485). Similar corrections occur at positions 68, 105 and 114. This case study highlights a key advantage of StableToken: its resilience does not depend on every branch being perfect, but on the collective ability to override sparse bit-flip errors.

## 5 Related Work

**Semantic Speech Tokenizers** **SSL tokenizers** rely on self-supervised learning (SSL) to extract discrete units from audio (Chen et al., 2022; Chung et al., 2021; Conneau et al., 2021; Chiu et al., 2022), but are proven suboptimal for generative speechLLMs (Mousavi et al., 2024; Guo et al., 2025). **Hybrid approaches** combine acoustic tokenizers with semantic distillation to balance fidelity and content (Zhang et al., 2023; Ye et al., 2025; Siahkoohi et al., 2022; Yang et al., 2024b), yet their high data rate and structural incompatibility with LLMs introduce practical challenges. Recent **fully supervised methods** have proven suitable for SpeechLLMs, which produce linguistic tokens while retaining sufficient phonetic information (Zeng et al., 2025; Du et al., 2024a), supporting expressive and efficient end-to-end SpeechLLMs (Zeng et al., 2024; Ding et al., 2025; Wu et al., 2025).

**Noise Robustness** While extensive research has focused on constructing robust Automatic Speech Recognition (ASR) models (Wang et al., 2022; Tjandra et al., 2023; Eickhoff et al., 2023; Gong et al., 2023; Ahn et al., 2025), the stability of discrete speech tokens under noisy conditions has received much less attention. Recently, R-SPIN (Chang & Glass, 2024) and NAST (Messica & Adi, 2024) have begun addressing this gap, but are limited to traditional SSL-based speech tokenizers.

A more detailed discussion of related work is presented in Appendix H due to page limit.

## 6 Conclusion

We introduce StableToken, a novel tokenizer designed to solve the critical instability of existing semantic tokenizers in noisy environments. By employing a multi-branch architecture and a consensus mechanism with bitwise voting, StableToken achieves state-of-the-art token stability. This stability directly translates to significant improvements in the robustness of downstream SpeechLLMs.

## Reproducibility Statement

To facilitate reproducibility of our work, we have provided detailed descriptions of the datasets, hyperparameters, and other experimental details used in our study in Section 3 and Appendix B, E, F. Our code and model checkpoint will be released publicly upon acceptance to further support reproducibility and foster future research.

## References


Adaeze Adigwe, Noé Tits, Kevin El Haddad, Sarah Ostadabbas, and Thierry Dutoit. The emotional voices database: Towards controlling the emotion dimension in voice generation systems. *arXiv preprint arXiv:1806.09514*, 2018.

Hyebin Ahn, Kangwook Jang, and Hoirin Kim. Hubert-vic: Improving noise-robust automatic speech recognition of speech foundation model via variance-invariance-covariance regularization. *arXiv preprint arXiv:2508.12292*, 2025.

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

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*, 2019.

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

Evelina Bakhturina, Vitaly Lavrukhin, Boris Ginsburg, and Yang Zhang. Hi-fi multi-speaker english tts dataset. *arXiv preprint arXiv:2104.01497*, 2021.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*, pp. 1–5. IEEE, 2017.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

Heng-Jui Chang and James Glass. R-Spin: Efficient Speaker and Noise-invariant Representation Learning with Acoustic Pieces. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 642–662. Association for Computational Linguistics, 2024.

Heng-Jui Chang, Alexander H Liu, and James Glass. Self-supervised fine-tuning for improved content representations by speaker-invariant clustering. In *Proc. Interspeech 2023*, pp. 2983–2987, 2023.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518, 2022.

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

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 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)*, pp. 244–250. IEEE, 2021.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. 2021.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 798–805. IEEE, 2023.

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

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

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

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

Kate Dupuis and M Kathleen Pichora-Fuller. Toronto emotional speech set (tess). 2010.

Patrick Eickhoff, Matthias Möller, Theresa Pekarek Rosin, Johannes Twiefel, and Stefan Wermter. Bring the Noise: Introducing Noise Robustness to Pretrained Automatic Speech Recognition. In *Artificial Neural Networks and Machine Learning – ICANN 2023*, pp. 381–392. Springer Nature Switzerland, 2023.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*, 2024.

Qingkai Fang, Yan Zhou, Shoutao Guo, Shaolei Zhang, and Yang Feng. Llama-omni2: Llm-based real-time spoken chatbot with autoregressive streaming speech synthesis. *arXiv preprint arXiv:2505.02625*, 2025.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.

Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. *arXiv preprint arXiv:2111.09344*, 2021.

Itai Gat, Felix Kreuk, Tu-Anh Nguyen, Ann Lee, Jade Copet, Gabriel Synnaeve, Emmanuel Dupoux, and Yossi Adi. Augmentation invariant discrete representation for generative spoken language modeling. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pp. 465–477, 2023.

Yuan Gong, Sameer Khurana, Leonid Karlinsky, and James Glass. Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong General Audio Event Taggers. In *Proc. Interspeech 2023*, pp. 2358–2362, 2023. doi: 10.21437/Interspeech.2023-1511.

Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. Recent advances in discrete speech tokens: A review. *arXiv preprint arXiv:2502.06490*, 2025.

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

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

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

Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, and Qun Liu. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. In *ICLR*, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Philip Jackson and SJUoSG Haq. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, 2014.

Léo Jacqmin, Lina M Rojas-Barahona, and Benoit Favre. " do you follow me?": A survey of recent approaches in dialogue state tracking. *arXiv preprint arXiv:2207.14627*, 2022.

Jesin James, Li Tian, and Catherine Inez Watson. An open source emotional speech corpus for human robot interaction applications. In *Interspeech*, pp. 2768–2772, 2018.

Chengze Jiang, Zhuangzhuang Wang, Minjing Dong, and Jie Gui. Survey of adversarial robustness in multimodal large language models. *arXiv preprint arXiv:2503.13962*, 2025.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7669–7673. IEEE, 2020.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 119–132, 2019.

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

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue state tracking with a language model using schema-driven prompting. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4937–4949, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.404. URL https://aclanthology.org/2021.emnlp-main.404/.

Keon Lee, Kyumin Park, and Daeyoung Kim. Dailytalk: Spoken dialogue dataset for conversational text-to-speech. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. Yodas: Youtube-oriented dataset for audio and speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

Zheng Lian, Haiyang Sun, Licai Sun, Kang Chen, Mingyu Xu, Kexin Wang, Ke Xu, Yu He, Ying Li, Jinming Zhao, et al. Mer 2023: Multi-label learning, modality robustness, and semi-supervised learning. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 9610–9614, 2023.

Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and Jim Glass. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. *Advances in Neural Information Processing Systems*, 36:58346–58362, 2023.

Wenrui Liu, Zhifang Guo, Jin Xu, Yuanjun Lv, Yunfei Chu, Zemin Liu, and Junyang Lin. Analyzing and mitigating inconsistency in discrete speech tokens for neural codec language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31035–31046, 2025.

Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

Vasista Sai Lodagala, Sreyan Ghosh, and Srinivasan Umesh. Ccc-wav2vec 2.0: Clustering aided cross contrastive self-supervised learning of speech representations. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1–8. IEEE, 2023.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025a.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. Language model can listen while speaking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24831–24839, 2025b.

Shoval Messica and Yossi Adi. Nast: Noise aware speech tokenization for speech language models. In *Proc. Interspeech 2024*, pp. 4169–4173, 2024.

Pooneh Mousavi, Luca Della Libera, Jarod Duret, Artem Ploujnikov, Cem Subakan, and Mirco Ravanelli. Dasb-discrete audio and speech benchmark. *arXiv preprint arXiv:2406.14294*, 2024.

Tu Anh Nguyen, Wei-Ning Hsu, Antony d'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, et al. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *arXiv preprint arXiv:2308.05725*, 2023.

Kari Ali Noriy, Xiaosong Yang, and Jian Jun Zhang. Emns/imz/corpus: An emotive single-speaker dataset for narrative storytelling in games, television and graphic novels. *arXiv preprint arXiv:2305.13137*, 2023.

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

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*, 2021.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*, 2018.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

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

Ali Siahkoohi, Michael Chinen, Tom Denton, W Bastiaan Kleijn, and Jan Skoglund. Ultra-low-bitrate speech coding with pretrained transformers. In *Proc. Interspeech 2022*, pp. 4421–4425, 2022.

Amitay Sicherman and Yossi Adi. Analysing discrete self supervised speech representation for spoken language modeling. In *ICASSP*, 2023.

Fengping Tian, Chenyang Lyu, Xuanfan Ni, Haoqin Sun, Qingjuan Li, Zhiqiang Qian, Haijun Li, Longyue Wang, Zhao Xu, Weihua Luo, et al. Marco-voice technical report. *arXiv preprint arXiv:2508.02038*, 2025.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. deHuBERT: Disentangling Noise in a Self-supervised Model for Robust Speech Recognition. *arXiv preprint arXiv:2302.14597*, 2023.

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

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2017.

Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech & Language*, 46:535–557, 2017.

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

Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *European conference on computer vision*, pp. 700–717. Springer, 2020.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024.

Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. Wav2vec-Switch: Contrastive Learning from Original-noisy Speech Pairs for Robust Speech Recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7632–7636. IEEE, 2022.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025.

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

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

Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. 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, 2024b.

Wanqi Yang, Yanda Li, Meng Fang, Yunchao Wei, Tianyi Zhou, and Ling Chen. Who can withstand chat-audio attacks? an evaluation benchmark for large language models. *arXiv preprint arXiv:2411.14842*, 2024c.

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

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

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

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, Yuxiao Dong, Jie Tang, et al. Scaling speech-text pre-training with synthetic interleaved data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6182–6186. IEEE, 2022.

Jian Zhang, Linhao Zhang, Bokai Lei, Chuhan Wu, Wei Jia, and Xiao Zhou. Wildspeech-bench: Benchmarking audio llms in natural speech conversation. *arXiv preprint arXiv:2506.21875*, 2025.

JTFLM Zhang and Huibin Jia. Design of speech corpus for mandarin text to speech. In *The blizzard challenge 2008 workshop*, 2008.

Linhao Zhang and Houfeng Wang. Using bidirectional transformer-crf for spoken language understanding. In *CCF international conference on natural language processing and chinese computing*, pp. 130–141. Springer, 2019.

Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Houfeng Wang. Graph lstm with context-gated mechanism for spoken language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 9539–9546, 2020.

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

Jinming Zhao, Tenggan Zhang, Jingwen Hu, Yuchen Liu, Qin Jin, Xinchao Wang, and Haizhou Li. M3ed: Multi-modal multi-scene multi-label emotional dialogue database. *arXiv preprint arXiv:2205.10237*, 2022.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Emotional voice conversion: Theory, databases and esd. *Speech Communication*, 137:1–18, 2022.

# A LARGE LANGUAGE MODEL (LLM) USAGE STATEMENT

In accordance with the conference policies on Large Language Model (LLM) usage, we hereby disclose the following: After completing the initial draft of this paper, we utilized an LLM to enhance grammar and polish the writing of this manuscript. No new research ideas, experimental designs, or scientific content were generated by the LLM. All scientific contributions, analyses, and conclusions presented in this work are solely those of the authors. We take full responsibility for the content of this paper, including all sections that have been revised or improved with LLM assistance. The LLM is not an author and did not contribute to the research ideation or substantive scientific writing.

This statement is provided to ensure transparency and compliance with the conference's policies on LLM usage.

# B DETAILS OF STABLETOKEN

## B.1 TRAINING DATASETS FOR STABLETOKEN

We train our StableToken model on hundreds of thousands of hours of both open-source data and in-house data. All open-source datasets used in this work are listed in Table 7.

Table 7: Summary of datasets used for training StableToken

| **Dataset** | **Duration (#hours)** | **Task** | **Language(s)** |
|---|---|---|---|
| LibriSpeech (Panayotov et al., 2015) | 960 | ASR | English |
| Multilingual LibriSpeech (Pratap et al., 2020) | 27,322 | ASR | English |
| The People's Speech (Galvez et al., 2021) | 5,568 | ASR | English |
| GigaSpeech (Chen et al., 2021) | 10,000 | ASR | English |
| Yodas (Li et al., 2023) | 29,155 | ASR | English |
| Hi-Fi TTS (Bakhturina et al., 2021) | 292 | ASR | English |
| VCTK (Veaux et al., 2017) | 44 | ASR | English |
| LibriTTS (Zen et al., 2019) | 586 | ASR | English |
| VoiceAssistant-400K (Xie & Wu, 2024) | 679 | ASR | English |
| AISHELLL-1 (Bu et al., 2017) | 150 | ASR | Chinese |
| WenetSpeech (Zhang et al., 2022) | 10,005 | ASR | Chinese |
| Common Voice (Ardila et al., 2019) | 2,133 | ASR | English, Chinese |
| Emilia (He et al., 2024) | 96,750 | ASR | English, Chinese |

## B.2 TRAINING HYPERPARAMETERS FOR STABLETOKEN

Table 8 summarizes the main hyperparameters used throughout StableToken training.

Table 8: Hyperparameters used for training StableToken

| **Hyperparameter** | **Value** |
|---|---|
| optimizer_type | AdamW |
| lr_scheduler | OneCycleLR |
| max_lr | 1.5e-5 |
| warmup_steps | 1000 |
| weight_decay | 0.01 |
| grad_clip | 1.0 |
| consensus_loss_weight ($\lambda_1$) | 0.25 |
| commitment_loss_weight ($\lambda_2$) | 0.25 |
| codebook_entropy_loss_weight ($\lambda_3$) | 1.0 |

### B.3 Details of Stochastic Perturbations During Training

We detail the construction and parameterization of the stochastic augmentation function $\mathcal{A}(\cdot)$ as introduced in Section 2.3. For each sample, one perturbation type is randomly selected from the following five categories: Gaussian Noise, Pink Noise, Brown Noise, Bit Crush Distortion, and Real-world Noise. The intensity of the selected perturbation is then uniformly sampled from a predefined range specific to each type, as summarized in Table 9. For real-world noise, an additional random selection of a noise audio clip is performed. The pool of noise clips consists of samples from the AudioCaps (Kim et al., 2019), FSD50k (Fonseca et al., 2021), and ESC-50 (Piczak, 2015) datasets. Notably, the ESC-10 subset of ESC-50 is excluded from training and reserved exclusively for evaluation as out-of-domain real-world noise.

Table 9: Perturbation types and their corresponding intensity ranges utilized during training.

| **Perturbation Type** | **Intensity Range** |
|---|---|
| Gaussian Noise | $16 \leq \text{SNR} \leq 30$ |
| Pink Noise | $16 \leq \text{SNR} \leq 24$ |
| Brown Noise | $12 \leq \text{SNR} \leq 24$ |
| Bit Crush Distortion | $8 \leq \text{Bit Depth} \leq 14$ |
| Real-world Noise | $12 \leq \text{SNR} \leq 24$ |

### B.4 Discussion of Consensus Objective Loss Choices

We choose the $L_2$ loss (Mean Squared Error Loss) for the consensus objective in Eq. 4 due to several considerations:

First, $L_2$ loss offers a simple and direct way to minimize the differences among multiple branches. At the same time, its form is also naturally compatible with the existing commitment loss in quantization-based models, thus facilitating stable optimization and consistent gradient behavior across objectives. Furthermore, by averaging representations from all branches, the resulting target inherently incorporates a form of confidence weighting, as outlier branch results are diluted in the mean.

Second, since all branches originate from the same underlying input, with only noise perturbations in minority branches, the goal is not to increase inter-class separation as in contrastive learning, but to enforce similarity across the noisy and clean versions. The L2 loss directly minimizes the Euclidean distance between the branches and their consensus, effectively encouraging robust invariance without introducing additional factors or requiring extra sampling. Moreover, cosine similarity is less sensitive to the number of bit flips in high-dimensional binary codes, especially when representations are already close, such as the flip of a single bit, which corresponds to small changes in angle and often results in diminished gradient signals and less effective correction of localized errors.

### B.5 Audio Reconstruction Details

Table 10: Summary of datasets used for training the flow matching model

| **Dataset** | **Language(s)** |
|---|---|
| Librilight (Kahn et al., 2020) | English |
| WenetSpeech (Zhang et al., 2022) | Chinese |
| Yodas2 (Li et al., 2023) | English, Chinese |
| Emilia (He et al., 2024) | English, Chinese |

Following the framework of CosyVoice (Du et al., 2024a) and GLM-4-Voice (Zeng et al., 2025), we train a flow matching model to reconstruct audio from speech tokens. The model takes as input the speech token representations and produces Mel spectrograms. Finally, a HiFi-GAN vocoder (Kong et al., 2020) converts the generated Mel spectrograms into the speech waveforms. The datasets for training the flow matching model is listed in Table 10 (excluding our in-house datasets, which comprise both English and Chinese speech data), and the training hyperparameters in Table 11.

Table 11: Hyperparameters used for training the flow matching model

| Hyperparameter | Value |
|---|---|
| optimizer_type | Adam |
| lr_scheduler | WarmupLR |
| learning_rate | 3.0e-4 |
| warmup_steps | 25000 |
| grad_clip | 1.5 |
| batch_type | dynamic |
| max_frames_in_batch | 10000 |

### B.6 Analysis on Computational Efficiency

#### B.6.1 Computational Overhead of Different Voter Counts

To further explore the computational overhead brought by increasing the number of voters ($N$), we present the model parameter counts and floating-point operations (FLOPs) for models with $N = 1$, 3, 5, and 7 in Table 12. Both metrics are measured using the THOP library[1]. For FLOPs calculation, we use an input audio with a duration of **30 seconds**.

The results show that as $N$ increases, the model parameters increase linearly, but the increment between adjacent $N$ values is relatively small (about 0.033M parameters). The FLOPs for different $N$ are also very close, indicating that enlarging $N$ does not introduce significant extra computational cost. This suggests that increasing the number of voters achieves better performance and robustness with only minimal impact on model size and inference efficiency.

Table 12: Model parameters and inference FLOPs for different voter counts $N$ on a 30-second input audio.

| Number of Voters ($N$) | #Parameters (M) | #FLOPs (G) |
|---|---|---|
| $N = 1$ | 320.261 | 480.978 |
| $N = 3$ | 320.294 (↑ 0.010%) | 481.003 (↑ 0.005%) |
| $N = 5$ | 320.328 (↑ 0.021%) | 481.028 (↑ 0.010%) |
| $N = 7$ | 320.361 (↑ 0.031%) | 481.053 (↑ 0.016%) |

#### B.6.2 Empirical Inference Efficiency Benchmark

From an architectural standpoint, the overhead introduced by the multi-branch design is negligible. The core design of StableToken confines the multi-branch architecture to the lightweight quantization stage only, while the computationally expensive Transformer Encoder remains a single-path process. Since the parallel quantization branches can be executed concurrently with high efficiency on modern hardware, this design introduces almost no additional end-to-end latency compared to conventional single-path tokenizers.

To rigorously evaluate the practical inference efficiency of StableToken, we conducted a comprehensive benchmark test measuring latency, Real-Time Factor (RTF), throughput, and memory footprint. We compared StableToken against a strong baseline, GLM-4-Voice-Tokenizer (Zeng et al., 2025), under identical hardware conditions using both an NVIDIA H20 GPU and an AMD EPYC 9K84 96-Core Processor.

The tests covered various batch sizes and audio durations. The detailed results are summarized in Table 13 (GPU) and Table 14 (CPU).

Based on these results, we observe the following: (1) **Latency, RTF, and Throughput:** The empirical data aligns with our theoretical analysis. The end-to-end latency and RTF of StableToken are nearly identical to the baseline, showing a slight advantage at larger batch sizes (e.g., 2.0% faster at a batch

[1] https://github.com/Lyken17/pytorch-OpCounter

Table 13: Inference Efficiency and Resource Usage on a single NVIDIA H20 GPU.

| Batch Size | Duration (s) | Metric | StableToken (25 tokens/s) | GLM-4-Voice (12.5 tokens/s) | Difference (%) |
|---|---|---|---|---|---|
| 1 | 1 | Latency (ms) | 9.11 | 9.00 | +1.22% |
| | | RTF | 0.0091 | 0.0090 | +1.22% |
| | | Throughput (s/s) | 109.77 | 111.11 | -1.21% |
| | | Memory (MB) | **1330** | 1538 | **-13.5%** |
| 1 | 30 | Latency (ms) | 63.17 | 63.19 | -0.03% |
| | | RTF | 0.0021 | 0.0021 | -0.03% |
| | | Throughput (s/s) | 474.92 | 474.79 | +0.03% |
| | | Memory (MB) | **1693** | 1853 | **-8.6%** |
| 16 | 10 | Latency (ms) | 235.96 | 239.57 | -1.5% |
| | | RTF | 0.0015 | 0.0015 | -1.5% |
| | | Throughput (s/s) | 678.08 | 667.87 | +1.5% |
| | | Memory (MB) | **2173** | 2328 | **-6.7%** |
| 32 | 60 | Latency (ms) | **1622.86** | 1656.49 | **-2.0%** |
| | | RTF | **0.0008** | 0.0009 | **-2.0%** |
| | | Throughput (s/s) | **1183.10** | 1159.08 | **+2.1%** |
| | | Memory (MB) | 13978 | 14230 | -1.8% |

Table 14: Inference Efficiency and Resource Usage on an AMD EPYC 9K84 96-Core Processor.

| Batch Size | Duration (s) | Metric | StableToken (25 tokens/s) | GLM-4-Voice (12.5 tokens/s) | Difference (%) |
|---|---|---|---|---|---|
| 1 | 1 | Latency (ms) | **104.27** | 116.62 | **-10.6%** |
| | | RTF | **0.1043** | 0.1166 | **-10.6%** |
| | | Throughput (s/s) | **9.59** | 8.57 | **+11.9%** |
| | | Memory (MB) | **2349** | 2727 | **-13.9%** |
| 1 | 30 | Latency (ms) | 1790.86 | 1809.86 | -1.0% |
| | | RTF | 0.0597 | 0.0603 | -1.0% |
| | | Throughput (s/s) | 16.75 | 16.58 | +1.0% |
| | | Memory (MB) | **2357** | 2739 | **-13.9%** |
| 16 | 10 | Latency (ms) | **5287.85** | 5687.53 | **-7.0%** |
| | | RTF | **0.0330** | 0.0355 | **-7.0%** |
| | | Throughput (s/s) | **30.26** | 28.13 | **+7.6%** |
| | | Memory (MB) | **2566** | 2982 | **-14.0%** |
| 32 | 60 | Latency (ms) | **52021.89** | 55549.44 | **-6.4%** |
| | | RTF | **0.0271** | 0.0289 | **-6.4%** |
| | | Throughput (s/s) | **36.91** | 34.56 | **+6.8%** |
| | | Memory (MB) | **2957** | 3117 | **-5.1%** |

size of 32 on GPU). Throughput matches or slightly exceeds the baseline across all tested scenarios. (2) **Memory Footprint:** StableToken consistently consumes less memory than the baseline, with savings of up to 13.5% on GPU and 14.0% on CPU. We attribute this efficiency to the use of Lookup-Free Quantization (LFQ), which is inherently more resource-efficient than the Vector Quantization (VQ) employed by the baseline.

In summary, the benchmarks demonstrate that StableToken achieves improved token stability without compromising inference efficiency. Its latency and throughput are competitive with strong baselines, while offering a lower memory footprint.

### B.7 Analysis of Token Distribution and Cross-Lingual Efficiency

In this section, we investigate the fundamental properties of the token distribution generated by our multi-branch architecture. Specifically, we analyze the entropy, vocabulary efficiency, and language-specificity of the learned representations. We conduct a statistical analysis on the Chinese and English token distributions to verify whether the architecture effectively learns distinct phonetic representations without introducing significant bias or inefficiency.

Our analysis focuses on two main aspects: (1) Evaluating the efficiency of vocabulary space utilization for individual languages. (2) Assessing the degree of distributional overlap between Chinese and English to confirm the capture of language-specific features.

#### B.7.1 Vocabulary Utilization and Distribution Entropy

We first analyze the vocabulary usage of StableToken on the LibriSpeech-test (English) and AISHELL (Chinese) datasets. As shown in Table 15, both languages utilize a substantial portion of the available vocabulary. The high entropy values observed for both languages indicate that the token distributions are rich and dispersed, rather than collapsing into a small subset of tokens. This suggests that the tokenizer maintains high representational capacity for both languages.

Table 15: Vocabulary Utilization Statistics on LibriSpeech and AISHELL

| Metric | English | Chinese |
|---|---|---|
| Total Tokens | 969,440 | 906,250 |
| Unique Tokens Used | 8,139 (99.35%) | 7,565 (92.35%) |
| Token Entropy | 12.43 | 11.68 |

#### B.7.2 Cross-Lingual Token Specificity

To assess the language specificity of the learned codebook, we examine the overlap of high-frequency tokens between languages. We identify the top-frequency tokens for one language and calculate their occurrence frequency in the other.

Tables 16 and 17 present these comparisons. The results indicate that tokens with high frequency in one language are extremely rare or non-existent in the other. This disjointed usage pattern demonstrates that the model has successfully learned specialized, language-specific phonetic units, allocating distinct sub-spaces of the codebook to different languages.

Table 16: Frequency Analysis of Top 5 English Tokens

| Token ID | Freq. (English) | Freq. (Chinese) | Rank in Chinese |
|---|---|---|---|
| 1877 | 0.4803% | 0.0119% | 2455 |
| 3813 | 0.3693% | **0.00%** | N/A |
| 3809 | 0.3377% | **0.00%** | N/A |
| 3812 | 0.3090% | **0.00%** | N/A |
| 3808 | 0.2957% | 0.0001% | 7320 |

Table 17: Frequency Analysis of Top 5 Chinese Tokens

| Token ID | Freq. (Chinese) | Freq. (English) | Rank in English |
|---|---|---|---|
| 3810 | 0.7531% | 0.0042% | 6276 |
| 3811 | 0.7530% | 0.0002% | 8045 |
| 7910 | 0.7392% | 0.0027% | 6850 |
| 3815 | 0.6933% | 0.0001% | 8104 |
| 3234 | 0.6815% | 0.0001% | 8085 |

### B.7.3 Quantitative Distributional Divergence

To formally quantify the difference between the token distributions of the two languages, we calculate the Kullback-Leibler (KL) Divergence. As shown in Table 18, the significant non-zero KL Divergence values confirm that the model maintains distinct probability distributions for each language.

Table 18: KL Divergence between Language Token Distributions

| Metric | Value |
|---|---|
| KL Divergence (English ‖ Chinese) | **2.81** |
| KL Divergence (Chinese ‖ English) | **2.05** |

Collectively, these analyses demonstrate that the multi-branch voting mechanism robustly allocates specific regions of the codebook to different languages, capturing unique phonetic characteristics while maintaining efficient vocabulary utilization.

## B.8 Long Audio Processing and Boundary Stability

In this section, we clarify the model's configuration regarding input duration and analyze the stability of tokenization at segment boundaries. This analysis is crucial for understanding the model's behavior in streaming scenarios or when processing long audio via chunking.

### B.8.1 Model Configuration and Segment Length

StableToken inherits the architectural specifications and processing conventions of its backbone, `whisper-large-v3`. Consequently, the model is configured as follows: (1) **Maximum Duration:** The tokenizer processes a maximum input audio duration of 30 seconds per inference pass, consistent with the Whisper context window. (2) **Training Constraint:** During training, all audio segments were constrained to this 30-second window. Audio files exceeding this duration in the training dataset were truncated to the first 30 seconds. (3) **Inference Strategy:** For audio exceeding 30 seconds, StableToken employs a standard chunking strategy. The input is segmented into 30-second (or smaller) chunks which are processed independently. The resulting token sequences are then concatenated to form the final output.

### B.8.2 Analysis of Boundary Stability

A potential concern with chunk-based processing is whether token stability degrades at the boundaries of audio segments (i.e., the beginning and end of a chunk), which could lead to defects when chunks are concatenated together.

To investigate this, we analyzed the distribution of tokenization inconsistencies across different temporal regions of the input audio under Gaussian noise perturbation. We segmented the test audio clips into three regions: (1) **Start:** The first 15% of the audio duration; (2) **Middle:** The central 70% of the audio duration; and (3) **End:** The final 15% of the audio duration. We calculated the Unit Edit Distance (UED) independently for each region. The results are presented in Table 19.

Table 19: Distribution of UED across different temporal regions of audio clips under Gaussian noise.

| **Region** | **Inconsistencies** | **Ref. Tokens** | **UED** | **% of Total Inconsistencies** |
|---|---|---|---|---|
| Start (0-15%) | 16,585 | 119,770 | 13.85% | 16.29% |
| Middle (15-85%) | 69,333 | 550,069 | 12.60% | 68.11% |
| End (85-100%) | 15,884 | 116,016 | 13.69% | 15.60% |
| **Overall** | 101,802 | 785,855 | 12.95% | 100.00% |

The empirical results indicate that UED is consistent across all regions. The UED in the middle region (12.60%) is only marginally lower than that at the start (13.85%) and end (13.69%). Furthermore, the

proportion of total inconsistencies occurring in the start (16.29%) and end (15.60%) regions closely aligns with their respective 15% share of the audio duration.

This uniformity demonstrates that there is no significant degradation in token stability at the segment boundaries. The consensus mechanism employed by StableToken maintains robustness consistently throughout the audio clip, supporting the effectiveness of the chunking strategy for long audio processing.

## C Ablation Study on Architectural Hyperparameters

In this section, we investigate the impact of two key architectural hyperparameters on model performance: the depth at which the quantizer is placed within the encoder, and the ratio of clean to perturbed branches during training. We compare our proposed configuration (**L16 + 3:2**, which means the quantizer is placed at Layer 16 with a 3:2 clean-to-noisy branch ratio) against three variants: (1) **Shallower Quantization (L8 + 3:2)**: The quantizer is placed at Layer 8; (2) **Deeper Quantization (L24 + 3:2)**: The quantizer is placed at Layer 24; and (3) **Reduced Noise Ratio (L16 + 4:1)**: The ratio of clean to perturbed branches is set to 4:1.

Table 20 summarizes the performance of each model checkpoint at 100k training steps across tokenizer robustness (UED), Automatic Speech Recognition (ASR), and Speech Emotion Recognition (SER) tasks.

Table 20: Ablation study on quantizer placement and perturbation ratio. UED is calculated under Gaussian Noise. LibriSpeech WER is reported for test-clean / test-other. SER is evaluated on the ESD dataset.

| **Configuration** | **Robustness** | **ASR (WER%, ↓)** | | | **SER** |
|---|---|---|---|---|---|
| (Layer + Clean:Noisy) | (UED%, ↓) | LibriSpeech | WenetSpeech | KeSpeech | (Acc%, ↑) |
| L16 + 3:2 (Ours) | 18.68 | **2.22 / 5.38** | 10.91 | **11.00** | **67.38** |
| L8 + 3:2 | 22.05 | 2.39 / 5.82 | 11.84 | 11.24 | 66.90 |
| L24 + 3:2 | **13.65** | 2.52 / 5.96 | **10.16** | 11.03 | 59.51 |
| L16 + 4:1 | 20.94 | 2.40 / 5.91 | 11.00 | 11.57 | 62.14 |

### C.1 Effect of Quantizer Placement

The results demonstrate a clear trade-off associated with the quantizer's depth: (1) **Deep Quantization (L24):** Placing the quantizer at the deep layer forces it to operate on highly abstract, semantic features. This yields the highest tokenizer robustness (lowest UED) and strong ASR performance on challenging datasets like WenetSpeech. However, at this depth, much of the fine-grained acoustic information (e.g., prosody and timbre) has been abstracted away, resulting in a significant degradation in SER performance (59.51% vs. 67.38%). (2) **Shallow Quantization (L8):** Operating on shallower features at Layer 8 retains acoustic details but lacks sufficient semantic abstraction. This leads to both poorer tokenizer robustness and degraded ASR performance compared to the deeper configurations. (3) **Balanced Configuration (L16):** Our proposed placement at Layer 16 strikes an effective balance. It captures sufficiently robust semantic content for high-performance ASR while retaining enough acoustic detail to excel in paralinguistic tasks like SER.

### C.2 Effect of Perturbation Ratio

Comparing the 3:2 and 4:1 clean-to-noisy ratios reveals the importance of the training signal difficulty: The 3:2 ratio results in significantly better tokenizer robustness compared to the 4:1 ratio (18.68% vs. 20.94% UED). A higher proportion of perturbed branches creates a more challenging training objective for the consensus mechanism.

This enhanced robustness translates directly to downstream tasks. We hypothesize that the more diverse training signal forces the model to learn representations that are truly invariant to perturbations rather than relying on the clean majority. This richer representation benefits both semantic (ASR) and paralinguistic (SER) performance.

In conclusion, these ablation studies validate the design choices of the proposed model. The combination of Layer 16 placement and a 3:2 noise ratio optimizes the trade-off between semantic robustness and acoustic feature preservation.

## D BASELINE MODELS

The baseline models considered in our study are as follows: (1) **HuBERT-500** (Hsu et al., 2021), a self-supervised speech representation model that leverages iterative offline clustering to produce pseudo-labels and employs a masked prediction loss; we use the official checkpoint with 500 clusters[2]; (2) **NAST** (Messica & Adi, 2024), a noise-aware speech tokenization approach comprising a predictor, residual encoder, and decoder, in which the predictor representations of clean speech and augmented speech are explicitly aligned; (3) **R-Spin** (Chang & Glass, 2024), which enhances the robustness of speech representations by learning discrete, speaker- and noise-invariant acoustic units through a prediction-based training objective; (4) **SpeechTokenizer** (Zhang et al., 2023), which introduces a hierarchical encoder-decoder framework with residual vector quantization (RVQ) to unify semantic and acoustic tokens; (5) **X-Codec** (Ye et al., 2025), which augments the RVQ backbone with semantic features from a pre-trained semantic encoder and applies a semantic reconstruction loss to achieve higher fidelity in audio generation; (6) **Mimi** (Défossez et al., 2024), a neural audio codec using RVQ to convert audio into discrete tokens, where the first quantization level is distilled to capture semantic information; (7) **CosyVoice ($\mathcal{S}^3$ Tokenizer)** (Du et al., 2024a), which extracts supervised semantic tokens from a multilingual speech recognition encoder for LLM-based TTS, thereby improving content consistency and speaker similarity in voice cloning; (8) **CosyVoice2** (Du et al., 2024b), which introduces finite-scalar quantization (FSQ) to improve the codebook utilization and updates the model architecture for streaming synthesis capabilities; (9) **GLM-4-Voice** (Zeng et al., 2025), which fine-tunes a pre-trained ASR model by including a pooling layer and a vector quantization layer, producing discrete tokens that strongly preserve semantic information at low frame rates.

## E NOISE PROFILES

In tokenizer-level evaluation, we augment the FLEURS (Conneau et al., 2023) benchmark with a variety of synthetic perturbations, including Gaussian Noise, Pink Noise, Brown Noise and Bit Crush Distortion, as well as real-world noise samples from the ESC-50 (Piczak, 2015) dataset. Specifically, the ESC-10 (Piczak, 2015) subset is used as out-of-domain (OOD) real-world noise and is excluded from our StableToken training pipeline, while the remaining 40 noise categories from ESC-50 are incorporated into the training process and thus are considered in-domain real-world noise.

We carefully adjusted the noise level to ensure that the added noise does not obscure the semantic content of the original audio and does not affect human perception of the speech. A summary of all perturbation types and their corresponding intensity is provided in Table 21.

Table 21: Details of synthetic and real-world perturbations used for noise augmentation.

| Perturbation Type | Intensity Value |
|---|---|
| Gaussian Noise | SNR = 25 |
| Pink Noise | SNR = 22 |
| Brown Noise | SNR = 16 |
| Bit Crush Distortion | Bit Depth = 10 |
| Real-world Noise | SNR = 16 |
| OOD Real-world Noise | SNR = 16 |

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/hubert

# F Details of Downstream Task Evaluation

## F.1 Training Datasets for SpeechLLMs

In this section, we summarize the speech datasets employed for training SpeechLLMs in Table 22, covering various tasks including Automatic Speech Recognition (ASR), Speech Emotion Recognition (SER), Text-to-Speech (TTS), and Speech Next Token Prediction (SNTP).

Table 22: Summary of datasets used for training SpeechLLMs

| **Dataset** | **Task** | **Language(s)** |
|---|---|---|
| LibriSpeech (Panayotov et al., 2015) | ASR | English |
| Multi-Lingual Librispeech (Pratap et al., 2020) | ASR | English |
| TESS (Dupuis & Pichora-Fuller, 2010) | SER | English |
| SAVEE (Jackson & Haq, 2014) | SER | English |
| RAVDESS (Livingstone & Russo, 2018) | SER | English |
| MELD (Poria et al., 2018) | SER | English |
| MEAD (Wang et al., 2020) | SER | English |
| JL-Corpus (James et al., 2018) | SER | English |
| IEMOCAP (Busso et al., 2008) | SER | English |
| Expresso (Nguyen et al., 2023) | SER | English |
| EmoV-DB (Adigwe et al., 2018) | SER | English |
| EMNS (Noriy et al., 2023) | SER | English |
| Dailytalk (Lee et al., 2023) | SER | English |
| CREMA-D (Cao et al., 2014) | SER | English |
| CASIA (Zhang & Jia, 2008) | SER | Chinese |
| M3ED (Zhao et al., 2022) | SER | Chinese |
| MER2023 (Lian et al., 2023) | SER | Chinese |
| CSEMOTIONS (Tian et al., 2025) | SER | Chinese |
| ESD (Zhou et al., 2022) | SER | English, Chinese |
| Hi-Fi TTS (Bakhturina et al., 2021) | TTS | English |
| VCTK (Veaux et al., 2017) | TTS | English |
| LibriTTS (Zen et al., 2019) | TTS | English |
| GigaSpeech (Chen et al., 2021) | SNTP | English |
| VoxPopuli (Wang et al., 2021) | SNTP | English |
| MagicData[3] | TTS | Chinese |
| AISHELL-1 (Bu et al., 2017) | TTS | Chinese |
| WenetSpeech (Zhang et al., 2022) | SNTP | Chinese |

## F.2 Training Hyperparameters for SpeechLLMs

Table 23 summarizes the main hyperparameters used throughout downstream SpeechLLM training. Unless otherwise specified, these settings are uniformly adopted for all tasks and models.

## F.3 Prompts Used in Downstream Tasks

This appendix contains the complete lists of textual prompts used for Automatic Speech Recognition (ASR), Speech Emotion Recognition (SER), and Text-to-Speech (TTS) tasks. For both fine-tuning and inference, a prompt was randomly selected from the corresponding set for each sample.

[3] https://www.openslr.org/68/

Table 23: Hyperparameters used for training all downstream SpeechLLMs.

| Hyperparameter | Value |
|---|---|
| optimizer_type | Adam |
| lr_scheduler | Cosine |
| learning_rate | 4.0e-4 |
| min_lr | 4.0e-5 |
| lr_decay_ratio | 0.75 |
| weight_decay | 0.1 |
| grad_clip | 1.0 |

### F.3.1 ASR TRANSCRIPTION PROMPTS

```
Please transcribe the following audio content into text.
Please convert the following recording into text.
Please transcribe this audio recording into text.
Transcribe the following audio content into text.
Convert the following recording into text.
This audio recording needs to be transcribed into text.
Recognize and convert the following speech content into text.
Turn the following audio file into text.
Transcribe this recording into text.
Transcribe the following audio file into text.
Convert this speech recording into text.
Recognize the following audio content and convert it into text.
Transcribe the following recording into text.
```

### F.3.2 SER EMOTION PROMPTS

```
What is the emotion of this text?
Analyze the sentiment of the following sentence.
Identify the feeling expressed in this audio.
Is the tone of this message positive or negative?
Detect the emotion in the user’s feedback.
What emotion is being conveyed here?
Classify the emotion of this statement.
Tell me the emotional state of the speaker.
Analyze the emotional content of this speech.
```

### F.3.3 TTS SYNTHESIS PROMPTS

```
Please synthesize the following text into speech.
Convert the following text to speech.
Transform the following text into speech.
This text needs to be synthesized into speech.
Synthesize the following text into speech.
Turn the following text into speech.
Generate speech from the following text.
Convert the text below into speech.
Create speech from the following text.
Produce speech from the following text.
Render the following text as speech.
```

## G Full Reconstruction Results

The comprehensive results for the tokenizer-level reconstruction quality evaluation are provided in Table 24. Note that SSL-based semantic tokenizers are not included in this comparison, as there are no publicly available decoders for reconstructing audio from their generated tokens.

Table 24: WER (↓) and MOS (↑) on LibriSpeech (Panayotov et al., 2015) and SEED (Anastassiou et al., 2024). StableToken combines strong noise robustness with competitive reconstruction quality. It is worth noting that a comparison is most meaningful between tokenizers of the same type.

| Model | #C | Frame Rate | BPS | WER ↓ | | | | MOS ↑ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | LS-clean | LS-other | SEED en | SEED zh | LS-clean | LS-other | SEED en | SEED zh |
| **Semantic Distilled tokenizer** | | | | | | | | | | | |
| SpeechTokenizer (Zhang et al., 2023) | 1 | 50Hz | 500 | 4.77 | 16.06 | 10.37 | 74.98 | 2.51 | 2.49 | 2.51 | 2.44 |
| | 3 | 50Hz | 1500 | 4.03 | 10.72 | 4.93 | 7.81 | 3.00 | 2.89 | 2.89 | 3.06 |
| | 8 | 50Hz | 4000 | 3.21 | 6.58 | 2.77 | 2.25 | 3.32 | 3.10 | **3.22** | **3.44** |
| X-Codec (Ye et al., 2025) | 1 | 50Hz | 500 | 3.98 | 9.02 | 4.72 | 5.96 | 3.17 | 3.04 | 3.05 | 3.18 |
| | 3 | 50Hz | 1500 | 3.16 | 6.11 | 2.74 | 2.24 | 3.43 | 3.17 | 3.19 | 3.38 |
| | 8 | 50Hz | 4000 | **3.09** | **5.49** | **2.25** | **1.74** | **3.47** | **3.19** | 3.19 | 3.33 |
| Mimi (Défossez et al., 2024) | 8 | 12.5Hz | 1100 | 4.65 | 9.84 | 3.86 | 2.81 | 3.26 | 3.06 | 3.15 | 3.19 |
| **Supervised Semantic tokenizer** | | | | | | | | | | | |
| GLM-4-Voice-Token. (Zeng et al., 2025) | 1 | 12.5Hz | 175 | 4.04 | 9.33 | 3.54 | 3.23 | 4.07 | **3.99** | **4.16** | 4.10 |
| $\mathcal{S}^3$ Tokenizer (Du et al., 2024a) | 1 | 25Hz | 300 | 5.78 | 13.38 | 5.91 | 4.26 | 3.40 | 3.31 | 3.40 | 3.31 |
| CosyVoice2 (Du et al., 2024b) | 1 | 25Hz | 325 | 4.25 | 9.68 | 4.34 | 2.75 | 3.36 | 3.25 | 3.31 | 3.58 |
| **StableToken (Ours)** | 1 | 25Hz | 325 | **3.84** | **7.99** | **3.44** | **2.62** | **4.09** | 3.83 | 4.01 | **4.18** |

## H Related Work

**Semantic Speech Tokenizers** The evolution of LLMs has driven the transition of spoken dialogue models from traditional pipelines to end-to-end SpeechLLMs (Zhang & Wang, 2019; Zhang et al., 2020; Jacqmin et al., 2022; Lee et al., 2021; Fang et al., 2024; Défossez et al., 2024; Wang et al., 2024), with semantic tokenizers becoming increasingly crucial. The design of semantic tokenizers has evolved through several distinct paradigms. Early approaches utilized self-supervised learning (SSL) to derive discrete units from unlabeled data (Hsu et al., 2021; Baevski et al., 2020; Chen et al., 2022; Chung et al., 2021; Conneau et al., 2021; Chiu et al., 2022; Baevski et al., 2019; Liu et al., 2023; Gat et al., 2023; Huang et al., 2022; Lodagala et al., 2023; Chang et al., 2023). The vast majority of tokens produced by these methods are designed for discriminative tasks. It is reported that discretized SSL tokens primarily encode phonetic information, causing high Gross Pitch Error (GPE) when paired with a vocoder for audio generation, making them unsuitable for end-to-end SpeechLLMs (Sicherman & Adi, 2023; Polyak et al., 2021; Mousavi et al., 2024; Guo et al., 2025).

A second category employs a hybrid approach, enhancing an acoustic tokenizer with semantic distillation to balance acoustic fidelity and semantic content (Zhang et al., 2023; Ye et al., 2025; Défossez et al., 2024; Siahkoohi et al., 2022; Yang et al., 2024b). This design enables strong performance on both generative and discriminative tasks. However, their integration with downstream large language models (LLMs) is hampered by several significant challenges. First, to preserve high fidelity, these methods tend to encode excessive acoustic details, which results in a high bits-per-second (BPS) rate. This high data rate generates longer token sequences, thereby increasing the computational load and impairing training efficiency. Furthermore, their reliance on Residual Vector Quantization (RVQ) produces hierarchical tokens that are inherently incompatible with the flat input structure expected by most LLMs. Collectively, the high data rate, the structural mismatch, and the overhead of processing superfluous acoustic information present substantial obstacles to their application in modern SpeechLLMs.

More recently, a third and more direct paradigm has gained traction: fully supervised training. Given that the primary goal is to capture semantic and phonetic information, this approach directly uses an Automatic Speech Recognition (ASR) objective for supervision. The process involves quantizing the

intermediate representations of a powerful ASR encoder and optimizing the model with an ASR loss, ensuring the resulting tokens directly represent linguistic units (Zeng et al., 2025; Du et al., 2024a;b). Subsequently, a downstream vocoder is trained to convert these discrete tokens into mel-spectrograms for speech synthesis. This tokenizer design is foundational to the current state-of-the-art end-to-end SpeechLLMs, underscoring its effectiveness and growing adoption. Interestingly, research has revealed that while the ASR objective targets linguistic content, the resulting tokens retain sufficient extra-phonetic information (e.g., prosody). This is likely because the ASR encoder implicitly learns to model prosodic features as they serve as valuable auxiliary cues for achieving high transcription accuracy. This retained information allows an integrated LLM to generate highly expressive synthesis and convey complex emotions. Consequently, this design's ability to support expressive generation has made it a foundational choice for state-of-the-art SpeechLLMs (Zeng et al., 2024; Ding et al., 2025).

**Noise Robustness** Ensuring the stability of discrete speech tokens in the presence of noise is critical for the performance of modern Speech Language Models (SLMs). However, this issue has been largely overlooked compared to the extensive research focused on improving the robustness of the Automatic Speech Recognition (ASR) model itself (Wang et al., 2022; Tjandra et al., 2023; Eickhoff et al., 2023; Gong et al., 2023; Ahn et al., 2025). Recently, two studies have begun to address this gap by investigating the noise robustness of traditional SSL-based speech tokenizers.

R-SPIN (Chang & Glass, 2024) addresses this by learning speaker- and noise-invariant discrete units through a data-efficient self-supervised framework. It extends the speaker-invariant clustering of Spin by using an additional noise-perturbed view of the input and an auxiliary loss that predicts "acoustic pieces," which are phoneme-aligned pseudo-labels, to prevent model collapse and ensure the resulting discrete units represent pure linguistic content . In contrast, NAST (Messica & Adi, 2024) proposes an architecture designed explicitly for robust tokenization, consisting of a predictor, a residual encoder, and a decoder. Its training is governed by a combination of a reconstruction loss, a diversity loss to encourage codebook usage, and a crucial robustness loss that penalizes changes in the predicted token distribution between clean and noise-augmented versions of the same utterance, thereby directly optimizing for token-level stability. Liu et al. (2025) introduce slice-consistency and perturbation-consistency constraints to mitigate discrete representation inconsistency, but their approach targets acoustic tokenizers (rather than semantic tokenizers), which prioritize audio detail reconstruction. Therefore, noise invariance is less meaningful in their context, making their work fundamentally different from ours.