DSA-Tokenizer: Disentangled Semantic-Acoustic Tokenization via Flow Matching-based Hierarchical Fusion

Anonymous ACL submission

Abstract

Speech tokenizers are a key building block of fully discrete Speech LLMs. Existing tokenizers either prioritize semantic encoding, fuse semantic content with acoustic style inseparably, or achieve incomplete semantic-acoustic disentanglement. To achieve better disentanglement, we propose **DSA-Tokenizer**, which explicitly disentangles speech into discrete semantic and acoustic tokens via distinct optimization constraints. Specifically, semantic tokens are supervised by ASR to capture linguistic content, while acoustic tokens focus on mel-spectrograms restoration to encode style. We further introduce a hierarchical Flow Matching decoder and a joint reconstruction-context inpainting training strategy, allowing the model to support both high-fidelity reconstruction and cross-utterance voice clone. To speed up inference, we distill the DiT decoder to reduce sampling steps of inference to 4 and improve synthesis quality with GAN fine-tuning. Experiments demonstrate that DSA-Tokenizer provides strong semantic-acoustic disentanglement, reliable controllable voice cloning, and efficient high-fidelity generation with low WER/CER. Moreover, our results suggest that disentangled tokenization provides a more effective interface for downstream large-model speech generation. Audio samples are avaialble at `https://anonymous.4open.science/w/DSA_Tokenizer_demo/`.

1 Introduction

The rapid advancement of large language models (LLMs) has catalyzed a paradigm shift in speech processing, spawning Speech LLMs (KimiTeam et al., 2025; Chen et al., 2025b; Chu et al., 2024; Xu et al., 2025) that unify speech and language processing within a single framework. Among existing architectures, **fully discrete Speech LLMs** (Zhang et al., 2025; Huang et al., 2025; Zeng et al., 2024; Nguyen et al., 2024) tokenize both input and output speech, enabling end-to-end processing in a unified discrete space with seamless LLM integration—yet their performance hinges heavily on the design of the speech tokenizer (Guo et al., 2025).

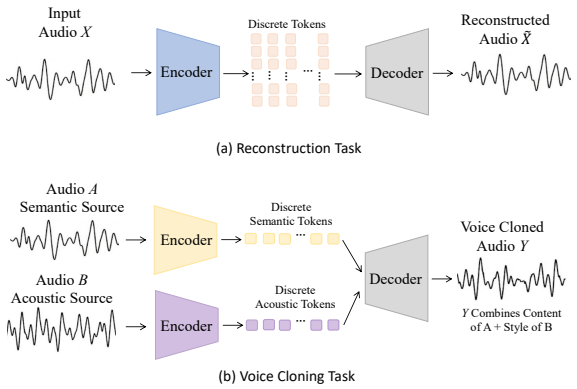

Figure 1: Illustration of (a) speech reconstruction and (b) cross-utterance voice cloning based on tokenizer

Existing speech tokenizers generally fall into three categories. **Semantic tokenizers** (Lakhotia et al., 2021; Hsu et al., 2021; Du et al., 2024) prioritize linguistic information via self-supervised learning or ASR supervision. While this facilitates integration with LLMs, they often discard essential acoustic cues like timbre. Conversely, **semantic-acoustic mixed tokenizers** (Défossez et al., 2022; Ji et al., 2025) target high-fidelity reconstruction but produce entangled representations, preventing independent attribute control. Finally, **shallowly disentangled tokenizers** (Défossez et al., 2024; Zhang et al., 2023b) attempt to decouple semantic and acoustic information atop mixed architectures. However, they often suffer from incomplete disentanglement, failing to achieve a clean separation of attributes.

To rigorously evaluate the disentanglement capability of these tokenizers, rather than relying solely on reconstruction quality or ASR performance, we argue that **cross-utterance voice cloning** (Figure 1) is a more direct evaluation task. In this setting, the semantic content is taken from one ut-

terance, while the speaker and prosodic style are taken from another. This requires the tokenizer to preserve content and style independently, making semantic-acoustic leakage immediately observable. Our experiments demonstrate that existing tokenizers exhibit substantial limitations in this task. Some models bias towards linguistic information, causing severe timbre mismatch, whereas others prioritize acoustic cues at the expense of semantic fidelity.

To address this gap, we propose **DSA-Tokenizer**. To ensure strict disentanglement, we utilize a **constrained dual-stream tokenizer** where semantic and acoustic tokens are supervised by ASR and a mel-spectrograms reconstruction objective, respectively. These streams are processed by a **flow-matching-based hierarchical fusion decoder**, which allows for high-fidelity reconstruction and cross-utterance voice cloning free from rigid length constraints. Finally, by adopting a **joint reconstruction-context inpainting training strategy** that combines self-reconstruction with contextual inpainting, we enforce the robust separation of attributes necessary for controllable speech generation.

To further improve practicality, we distill the DiT-based decoder, enabling generation in just four sampling steps, inspired by recent flow matching distillation approach ZipVoice(Zhu et al., 2025). We then apply an additional GAN fine-tuning stage on the full mel-spectrogram generated by the distilled decoder. This design is motivated by prior observations that Flow Matching in mel space, when trained with a mean squared error (MSE) objective, tends to emphasize dominant coarse structures while under-modeling perceptually salient fine-grained details, especially in low-energy regions (Yao et al., 2026). This GAN fine-tuning stage substantially improves audio quality while preserving semantic-acoustic controllability.

Experiments show that DSA-Tokenizer achieves strong reconstruction, voice cloning, and efficient high-quality 4-step decoding through GAN refinement. Further results on downstream Speech LLM tasks suggest that strict semantic-acoustic disentanglement provides a more effective token interface for controllable speech generation. Our core contributions are summarized as follows:

- We propose **DSA-Tokenizer**, a dual-stream speech tokenizer that explicitly disentangles semantic content and acoustic style while remaining compatible with fully discrete Speech LLMs.
- We introduce a DiT refinement strategy that distills the Flow-Matching decoder to perform inference in only 4 steps and improves mel-spectrogram generation with GAN fine-tuning, achieving a better audio generation quality.
- Experiments on downstream large-model speech generation show that strict semantic-acoustic disentanglement improves controllability and supports more effective speech generation.

2 Related Work

2.1 Speech Large Language Models

Recent Speech LLMs mainly follow two paradigms: thinker-talker systems that map continuous speech to discrete outputs, and fully discrete systems that tokenize both input and output speech (Chen et al., 2025b; Xu et al., 2025; Zeng et al., 2024; Zhang et al., 2025; Nguyen et al., 2024). Our work belongs to the latter line, where the capability of downstream speech generation depends heavily on the quality of the speech tokenizer.

2.2 Disentangled Speech Tokenizers

Existing speech tokenizers mainly trade off among semantic preservation, reconstruction fidelity, and semantic-acoustic disentanglement. Semantic tokenizers such as HuBERT(Hsu et al., 2021) approaches favor linguistic content but discard speaker and prosodic style. Neural codecs such as EnCodec(Défossez et al., 2022) and WavTokenizer(Ji et al., 2025) preserve rich acoustic detail but entangle content and style in shared token streams. More recent disentanglement-oriented methods, including SpeechTokenizer(Zhang et al., 2023b), DualCodec(Li et al., 2025), XY-Tokenizer(Gong et al., 2025), and SAC(Chen et al., 2025c), explicitly separate semantic and acoustic information through layered or dual-branch designs. However, these approaches still suffer from incomplete disentanglement, which limits their use in controllable speech generation.

3 Method

We propose **Disentangled Semantic-Acoustic Tokenizer** (DSA-Tokenizer), a disentangled speech tokenization framework tailored for fully discrete Speech LLMs, as shown in Figure2(a). Its goal is to achieve semantic-acoustic disentanglement

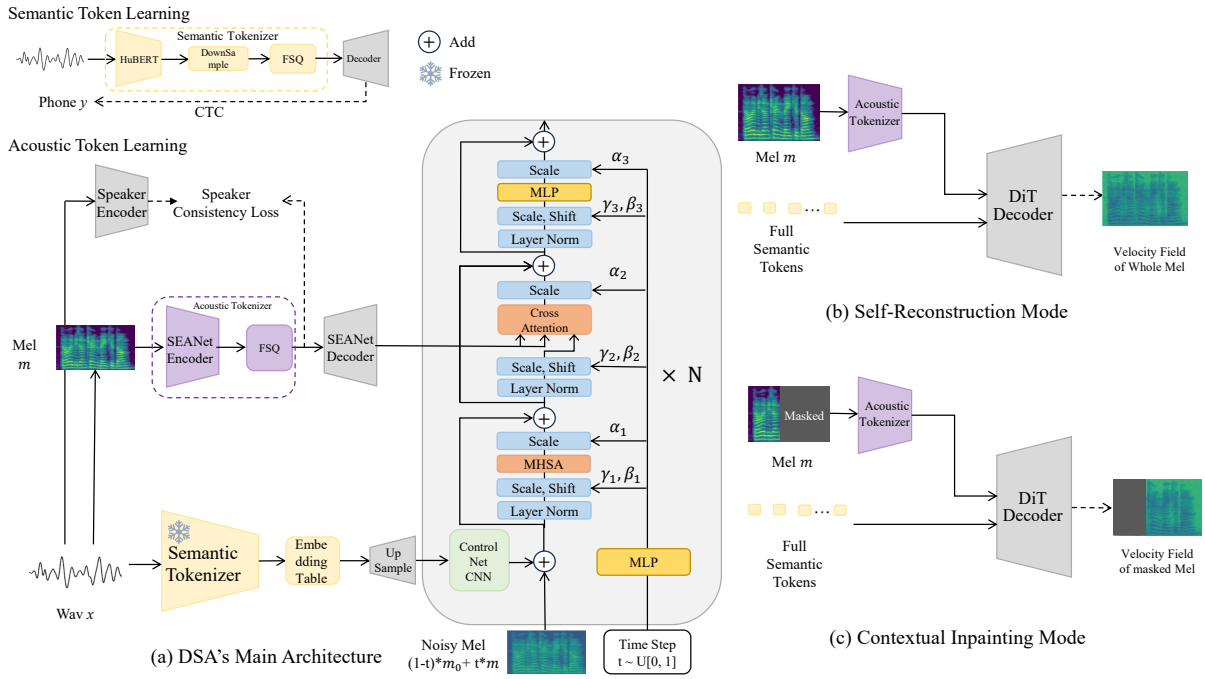

Figure 2: Overview of the proposed framework and training strategy. (a) DSA-Tokenizer framework: Input audio X is encoded into discrete semantic and acoustic tokens, which are fed into the DiT decoder for audio generation. (b) Self-Reconstruction Mode: The model learns to predict the velocity field of the full Mel-spectrogram based on the complete acoustic and semantic tokens. (c) Contextual Inpainting Mode: The model learns to predict the velocity field of the masked Mel-spectrogram region based on the acoustic tokens of the unmasked region and the full semantic tokens.

for high-fidelity speech reconstruction and cross-utterance voice cloning. To this end, the framework employs two parallel discrete token streams with orthogonal goals: semantic tokens (z_s) encoding linguistic content, and acoustic tokens (z_a) capturing style attributes. These two token streams are then fused via distinct condition injection methods in a DiT-based Flow-Matching decoder. To further enable efficient high-quality generation, we distill the decoder to 4-step inference and refine it with adversarial fine-tuning.

3.1 Dual Token Streams with Task-Specific Constraints

3.1.1 Semantic Token Learning

Following recent discrete speech tokenization work (Chen et al., 2025a; Tao et al., 2024; Discrete-Speech Team, 2025), we learn semantic tokens under ASR supervision so that they retain linguistic content while suppressing acoustic style information. Given a speech waveform x, we use a pre-trained HuBERT model as the semantic encoder, followed by a Finite Scale Quantization (FSQ) layer (Mentzer et al., 2023) to discretize the continuous representations into semantic tokens at 25 Hz:

$$z_s = \text{FSQ}\left(\text{HuBERT}(x)\right) \in \mathcal{Z}^{T_s \times D_s}$$

where $\mathcal{Z}$ denotes the discrete token space, T_s is the sequence length, and D_s is the FSQ codebook size. We enforce strict linguistic constraints by training the HuBERT-FSQ pipeline with Connectionist Temporal Classification (CTC) loss (Graves et al., 2006). After training, the auxiliary CTC decoder is discarded and the HuBERT-FSQ module is frozen as the semantic tokenizer for subsequent acoustic token learning. The semantic tokenizer uses FSQ with 6 channels and 4 levels per channel (4^6).

3.1.2 Acoustic Token Learning

Acoustic tokens are designed to capture speaker and prosodic attributes that are complementary to the linguistic content in semantic tokens. Specifically, mel-spectrograms m are extracted from raw speech x, downsampled via a SEANet-style (Tagliasacchi et al., 2020) encoder, and discretized through a FSQ quantizer to yield the acoustic tokens:

$$z_a = \text{FSQ}\left(\text{SEANetEncoder}(m)\right) \in \mathcal{Z}^{T_a \times D_a}$$

where T_a denotes the sequence length jointly determined by the CNN structure and mel-spectrogram length, and D_a is the FSQ codebook size. This design does not require T_a and T_s to have the same length, facilitating cross-utterance voice cloning.

Unlike the pre-trained semantic token stream, the acoustic tokenizer is trained **end-to-end** with the DiT decoder (Sec. 3.2). The Flow Matching loss is backpropagated through the discrete bottleneck via the straight-through estimator to optimize the SEANet encoder. This drives z_a to capture residual acoustic detail beyond the linguistic content encoded in z_s. The acoustic tokenizer uses FSQ with 8 channels and 4 levels per channel (4^8).

3.1.3 Joint Reconstruction-Contextual Inpainting Training Strategy

Training with reconstruction alone often encourages information leakage between semantic and acoustic representations and makes voice cloning harder to support. To address this issue, we adopt a joint **reconstruction-contextual inpainting** training strategy, where each batch is randomly assigned to one of two modes with equal probability.

Self-Reconstruction Mode (Figure 2(b)): The full sequences of z_s (semantic tokens) and z_a (acoustic tokens) are provided as conditions. The model learns to predict the Flow Matching velocity field of the entire mel-spectrogram for high-fidelity reconstruction.

Contextual Inpainting Mode (Figure 2(c)): we randomly sample a split point τ on the time axis and mask the mel-spectrogram after τ. The decoder is then trained to predict the masked region conditioned on the full semantic token sequence z_s and only the acoustic prefix $z_a^{<\tau}$. This design encourages the model to infer global acoustic style from partial acoustic context while following semantic guidance, thereby reducing information leakage between the two streams.

3.2 Hierarchical Flow Matching Decoder for Semantic-Acoustic Token Fusion

Fusing semantic tokens (z_s) and acoustic tokens (z_a) requires modeling two fundamentally different types of conditions: linguistic content demands strict temporal alignment, whereas acoustic style should be injected more flexibly across time. We therefore adopt a DiT-based (Peebles and Xie, 2023; Ho et al., 2020) Flow Matching (Lipman et al., 2023) decoder with a **hybrid injection strategy**. Semantic tokens are treated as the structural backbone of speech and injected as dense temporal conditions to enforce content alignment. Acoustic tokens, in contrast, are injected via **cross-attention**, allowing the decoder to retrieve speaker and prosodic cues. This design enables both high-fidelity reconstruction and cross-utterance voice cloning.

3.2.1 Semantic ControlNet-Style Injection

We first project semantic tokens z_s into continuous embeddings e_s via a learnable codebook, and then upsample them with linear interpolation to match the mel-spectrogram length T_{mel}, yielding aligned semantic features $\tilde{e}_s$. Following standard Flow Matching paradigms, at timestep $t \sim \mathrm{U}[0, 1]$, the noisy mel-spectrogram is defined as

$$m_t = (1 - t) \cdot m_0 + t \cdot m,$$

where $m_0 \sim \mathcal{N}(0, I)$ and m is the target mel-spectrogram. To enforce strict linguistic alignment, we inject semantic information as a dense structural condition. Specifically, a lightweight ControlNet-Style(Zhang et al., 2023a) CNN adapter processes $\tilde{e}_s$, and the result is directly added to m_t:

$$F_{\text{sem}} = \mathrm{CNN}(\tilde{e}_s) + m_t.$$

By treating semantic information as the structural backbone embedded into the noisy input, we ensure the model captures long-range linguistic context prior to integrating acoustic details.

3.2.2 Acoustic Injection via Cross-Attention

Unlike semantic content, acoustic style does not require strict frame-level alignment. We therefore inject acoustic information through cross-attention to provide flexible speaker and prosodic conditioning. Specifically, acoustic tokens z_a are projected to continuous embeddings via a learnable codebook and upsampled to the mel-spectrogram length T_{mel} using a SEANet-style decoder, yielding the aligned acoustic features $\tilde{e}_a$.

Within the DiT blocks, the self-attention-enhanced semantic feature F_{sem} serves as queries, while keys and values are derived from $\tilde{e}_a$ via dedicated linear projections W^K and W^V. The cross-attention fused feature is computed as:

$$F_{\text{sem-aco}} = \text{CrossAttn}\left(F_{\text{sem}}, \tilde{e}_a W^K, \tilde{e}_a W^V\right)$$

This design allows the decoder to retrieve speaker and prosodic cues flexibly without imposing rigid length constraints between semantic and acoustic tokens, thus facilitating cross-utterance voice cloning.

3.2.3 Training Objectives

We train the base decoder with a combination of Flow Matching loss for conditional mel-spectrogram generation and speaker consistency loss for acoustic style preservation.

Flow Matching Loss Following the Conditional Flow Matching (CFM) (Ho and Salimans, 2022), given the noisy mel-spectrogram

$$m_t = (1-t)m_0 + tm,$$

where $m_0 \sim \mathcal{N}(0, I)$ and $t \sim \mathrm{U}[0, 1]$, we train the decoder to predict the corresponding velocity field conditioned on the semantic and acoustic features:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{t,m_0,m}\left[\|v_t - v_\theta(m_t, t, \tilde{e}_s, \tilde{e}_a)\|^2\right].$$

Speaker Consistency Loss To encourage acoustic tokens to preserve speaker identity during the first-stage training, we extract a reference speaker embedding s_{ref} from the raw waveform using a WavLM-based speaker verification model (Chen et al., 2022) [1]. We then align it with the pooled acoustic-token representation:

$$\mathcal{L}_{\text{spk}} = 1 - \cos\left(s_{\text{ref}}, \mathrm{AttnPool}(e_a)\right)$$

where $\mathrm{AttnPool}(\cdot)$ aggregates the acoustic token embeddings $\mathbf{e}_a$ into a global vector.

The total training loss is a weighted sum: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{fm}} + \lambda_{\text{spk}}\mathcal{L}_{\text{spk}}$, where $\lambda_{\text{spk}} = 1.0$. Additional objectives used in few-step distillation and adversarial fine-tuning are introduced in the next subsection.

3.2.4 Few-step distillation and Adversarial Fine-Tuning

To improve decoding efficiency and make the end-to-end GAN fine-tuning feasible, we freeze the tokenizer and distill the pretrained Flow-Matching decoder into a 4-step student following ZipVoice. During distillation, each batch retains both self-reconstruction and contextual inpainting modes, and the student DiT decoder is trained on forward passes from both, allowing the student to preserve reconstruction fidelity and cross-utterance voice cloning capability under short-step decoding.

We then adversarially fine-tune the distilled 4-step model in mel space, while keeping the tokenizer frozen. For each batch, self-reconstruction and contextual inpainting samples are both forwarded for 4 steps to generate full mel-spectrograms. Their effective mel lengths vary across samples: self-reconstruction examples inherit the original utterance duration, while the supervised region in contextual inpainting depends on the randomly sampled mask boundary. We therefore define the adversarial training length as the shortest effective mel length t within the batch, and randomly crop one segment of length t from each fake/real pair before feeding it to a multi-scale mel-domain PatchGAN(Wang et al., 2018) discriminator. This design enables stable joint adversarial training on mixed batches without introducing additional length constraints.

To preserve speaker identity during this stage, we introduce a proxy speaker encoder trained to match the embedding space of a frozen WavLM-based speaker verification model[1]. The proxy encoder operates directly on generated mel-spectrograms and provides a differentiable mel-level speaker consistency loss, which is applied only to self-reconstruction samples:

$$\mathcal{L}_{\text{spk}}^{\text{mel}} = 1 - \cos\left(h_\phi(\hat{m}), s_{\text{ref}}\right),$$

where $\hat{m}$ is the mel-spectrogram generated by the distilled decoder, $h_\phi(\cdot)$ is the proxy speaker encoder, and s_{ref} is the same reference speaker embedding used in the first-stage speaker consistency loss. The generator is optimized with the original first-stage objective together with least-squares adversarial(Mao et al., 2017) and feature-matching losses(Salimans et al., 2016):

$$\mathcal{L}_G = \mathcal{L}_{\text{total}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}^{\text{mel}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}^{\text{mel}} + \lambda_{\text{spk}}^{\text{mel}}\mathcal{L}_{\text{spk}}^{\text{mel}}.$$

More training details are provided in Appendix C.2.

4 Experiment Setup

4.1 Evaluation Tasks

4.1.1 Reconstruction and Cross-Utterance Voice Cloning

We evaluate DSA-Tokenizer on two waveform-generation settings: **Reconstruction**, where a single utterance provides both semantic and acoustic sources, and **Cross-Utterance Voice Cloning**, where semantic and acoustic tokens are taken from different utterances. We report UTMOS (Saeki et al., 2022) for speech naturalness, WER for English and CER for Chinese for content consistency, and SIM for speaker/style preservation.

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

4.1.2 Disentanglement Probing

To assess semantic-acoustic disentanglement, we probe whether different token streams preserve only task-relevant information. ASR is used to evaluate semantic information, and speaker classification (SC) is used to evaluate speaker-related acoustic information. For single-layer baselines, ASR and SC are applied to the same token sequence; for multi-layer baselines, both are applied to Layer 0 and Layers 1–7; for DSA-Tokenizer and SAC, both are applied separately to semantic and acoustic tokens. For Facodec, ASR is applied to content code and SC is applied to other parts of Facodec.

4.1.3 Downstream Speech LLM Experiments

We further evaluate whether DSA-Tokenizer provides a better token interface for speech LLMs through two downstream tasks: **LLM-based voice cloning** and **LLM-based TTS**. For voice cloning, the autoregressive model is trained on triplets $(\mathcal{S}, \mathcal{A}, \mathcal{C})$, where $\mathcal{S}$ provides semantic content, $\mathcal{A}$ provides acoustic style, and $\mathcal{C}$ is the target speech. For zero-shot TTS, we adopt a pure LLaSA-style autoregressive architecture (Ye et al., 2025), which generates speech tokens conditioned on text and prompt speech.

4.2 Datasets

DSA-Tokenizer Training: We train the semantic tokenizer on about 4,000 hours of Chinese-English speech-text aligned data, and train the acoustic tokenizer and decoder on a cleaned 100k-hour Chinese-English subset of Emilia (He et al., 2024). Additional training, cleaning, and architecture details are provided in Appendices B, C.1, and A.

Waveform Evaluation: We use the multilingual SeedTTS dataset (Anastassiou et al., 2024) for English and Chinese reconstruction and voice cloning evaluation.

Disentanglement Probing: We use LibriSpeech (Panayotov et al., 2015) for ASR-based semantic evaluation and VoxCeleb1 (Nagrani et al., 2020) for speaker classification.

LLM-based Voice Cloning: We construct over 350,000 training triplets from LibriTTS train-960 and evaluate on over 4,000 randomly matched test pairs from LibriTTS test-clean. Target speech is synthesized by F5TTS (Chen et al., 2025d) and Cosyvoice2 (Du et al., 2024).

LLM-based TTS: We follow the same data distribution as VoxBox (Wang et al., 2025).

4.3 Baselines

We compare against representative single-layer, multi-layer, and dual-stream tokenizers, including WavTokenizer, Mimi, EnCodec, SpeechTokenizer, DualCodec, SAC, Cosy2 S^3, and Facodec. For both LLM-based voice cloning and TTS, we use Qwen3-0.6B (Yang et al., 2025) as the autoregressive backbone and expand its vocabulary with the corresponding speech tokens. Additional dataset, task-formatting, and baseline details are provided in Appendices E and D.

5 Results and Discussion

5.1 Can DSA-Tokenizer Unify Reconstruction and Voice Cloning?

DSA-Tokenizer is the only tokenizer that remains strong in both high-fidelity reconstruction and cross-utterance voice cloning. As shown in Table 1, DSA-Tokenizer achieves the highest UTMOS scores in both English and Chinese under both evaluation settings, while also maintaining very low WER/CER at the lowest bitrate. In reconstruction task, although DSA-Tokenizer does not achieve the highest SIM, we argue it is a **trade-off** under a tight bitrate budget: instead of maximizing reconstruction SIM, our model learns more usable semantic-acoustic tokens that better support Speech LLMs(Section 5.3). For voice cloning task, DSA-Tokenizer achieves the best UTMOS, SIM, and WER/CER simultaneously. In contrast, reconstruction-oriented non-disentangled baselines such as WavTokenizer can achieve strong reconstruction quality, but they do not support robust voice cloning. Among disentangled baselines, SAC remains competitive in reconstruction, yet suffers a severe degradation in voice cloning. Although FaCodec uses continuous timbre features, it still performs consistently worse than DSA-Tokenizer across both languages. We further compare DSA-Tokenizer with the CosyVoice series, which is not fully discrete, for voice cloning in Appendix F. DSA-Tokenizer outperforms all CosyVoice systems in English and achieves the best Chinese CER and UTMOS while remaining competitive in SIM.

5.2 Can Discrete Tokens Capture Sufficient Information without Leakage?

DSA-Tokenizer preserves sufficient task-relevant information in both kinds of tokens while minimizing leakage. As illustrated in Figure 3, different tokenizers exhibit distinct encoding patterns based on

Model	BitRate (kbps)	English			Chinese		
		UTMOS ↑	WER (%) ↓	SIM ↑	UTMOS ↑	CER (%) ↓	SIM ↑
Ground Truth	-	-	2.14	-	-	1.33	-
Reconstruction Task							
WavTokenizer (75 Hz)	0.90	3.92	3.23	**0.84**	2.82	5.08	0.62
Mimi (8-layer)	1.10	3.30	3.46	0.74	2.37	2.93	0.73
Encodec (2-layer)	1.50	1.56	5.40	0.61	1.35	5.19	0.62
SpeechTokenizer (2-layer)	1.00	2.12	7.57	0.36	1.75	24.39	0.33
DualCodec (12.5 Hz, 6-layer)	0.925	3.78	2.58	0.76	2.95	1.62	0.80
SAC (f_s=12.5 Hz, f_a=50 Hz)	0.875	3.88	2.03	0.83	2.99	**1.53**	**0.87**
Facodec	-	3.54	2.26	0.84	2.82	2.01	0.83
DSA-Tokenizer (f_s=25 Hz, f_a=25 Hz)	**0.70**	**4.06**	**2.00**	0.77	**3.30**	1.81	0.82
Cross-Utterance voice cloning Task							
Mimi (8-layer)	1.10	2.19	107.51	0.53	1.54	100.01	0.53
Encodec (2-layer)	1.50	1.25	98.31	0.09	1.24	68.97	0.13
SpeechTokenizer (2-layer)	1.00	1.40	12.98	0.13	1.34	123.33	0.15
DualCodec (12.5 Hz, 6-layer)	0.925	2.39	17.36	0.11	1.73	16.22	0.32
SAC (f_s=12.5 Hz, f_a=50 Hz)	0.875	1.34	90.22	0.13	1.27	71.87	0.30
Facodec	-	2.52	5.78	0.26	1.86	25.1	0.39
DSA-Tokenizer (f_s=25 Hz, f_a=25 Hz)	**0.70**	**4.16**	**2.47**	**0.61**	**3.55**	**2.16**	**0.71**

Table 1: Performance comparison of different tokenizers on speech reconstruction and cross-utterance voice cloning tasks. f_s means semantic token rate and f_a means acoustic token rate. Baseline models with similar bitrate are selected,annotated with their codebook layers or token rates.

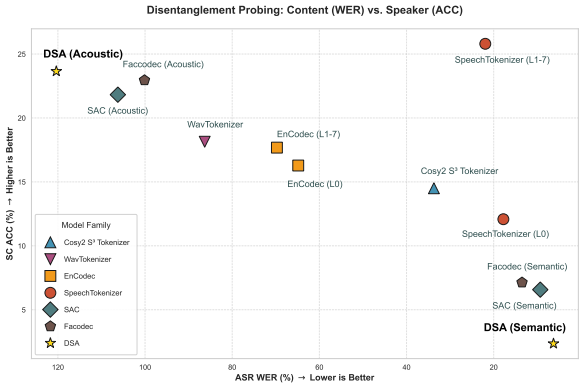

Figure 3: Disentanglement probing evaluation results. (L0) means the first layer, (L1-7) means the second to eighth layers. Tokens from multiple layers are concatenated for the experiments.

their objectives. First, Single-codebook models are highly dependent on their training objectives: the ASR-supervised CosyVoice2 S^3 tokenizer is more semantic-oriented, whereas the reconstruction-trained WavTokenizer retains more speaker-related information at the cost of much worse WER. Second, multi-layer tokenizers demonstrate varying disentanglement capabilities. EnCodec, lacking explicit constraints, shows uniform performance across layers, suggesting semantic and acoustic entangled. In contrast, SpeechTokenizer achieves shallow disentanglement by distilling semantics into layer 0, which yields significantly lower WER and SC ACC compared to layers 1–7. Third, among dual-stream baselines, both SAC and Facodec achieve partial disentanglement, but their semantic tokens still retain more speaker information and their acoustic tokens retain more semantic residue than DSA-Tokenizer. In contrast, DSA-Tokenizer attains the lowest WER on semantic tokens (6.28%) together with the lowest SC ACC (2.35%), while its acoustic tokens achieve the highest SC ACC (23.65%) and the highest WER (120.36%). These results suggest that DSA-Tokenizer provides the cleanest semantic-acoustic disentanglement among the compared tokenizers. This probing pattern is also consistent with its superior voice cloning performance in Table 2. Taken together, the probing and voice cloning results suggest that voice cloning is a **stringent test** of semantic-acoustic disentanglement, and that cleaner disentanglement in tokenizers leads to more effective voice cloning.

5.3 Does Disentanglement Benefit Downstream Speech LLMs?

Downstream speech LLM experiments suggest that cleaner semantic-acoustic disentanglement provides a more effective token interface for acoustic-related generation. In LLM-based voice cloning (Table 2), the non-disentangled WavTokenizer performs worst on all metrics and even leads to unstable generation, showing that strong reconstruction alone does not guarantee robust downstream mod-

Tokenizer	UTMOS ↑	WER (%) ↓	SIM ↑
WavTokenizer	3.75	89.44	0.28
SAC	3.35	22.89	0.38
DSA-Tokenizer	**4.10**	**16.55**	**0.48**

Table 2: LLM-based voice cloning performance.

Model	WER (%) ↓	SIM ↑	UTMOS ↑
Seed-TTS test-en			
Llasa-1B-80k	3.71	0.54	4.06
Llasa-1B-160k	3.60	0.56	4.05
Llasa-1B-250k	2.99	0.57	4.07
Spark-TTS	1.98	0.58	3.94
SAC	**1.06**	0.54	4.21
DSA	1.39	**0.60**	**4.24**
Seed-TTS test-zh			
Llasa-1B-80k	2.69	0.65	3.27
Llasa-1B-160k	2.22	0.66	3.28
Llasa-1B-250k	1.89	0.67	3.28
Spark-TTS	1.20	0.67	3.27
SAC	**0.90**	0.65	3.34
DSA	1.37	**0.68**	**3.51**

Table 3: Pure LLaSA-style autoregressive TTS performance on Seed-TTS.

eling. Compared with the partially disentangled SAC, DSA-Tokenizer achieves clearly better UTMOS and SIM while also improving WER, indicating that cleaner factorization makes cross-utterance content-style voice cloning easier for the LLM.

This trend remains visible in pure LLaSA-style autoregressive TTS (Table 3). Non-disentangled tokenizers used by the LLaSA baselines and Spark-TTS can produce reasonable speech, but they remain weaker than disentangled tokenizers in the overall performance. Among disentangled tokenizers, SAC attains the lowest WER, suggesting that partial disentanglement already helps content modeling in this setting. DSA-Tokenizer, however, achieves the best UTMOS and SIM in both English and Chinese while keeping WER competitive. Taken together, these results suggest that **cleaner semantic-acoustic disentanglement provides a better token interface for Speech LLMs.**

5.4 What Design Choices Are Essential?

We ablate both the base tokenizer design and the short-step decoder refinement. We first examine the core design choices of the original DSA-Tokenizer (before distillation and GAN fine-tuning) in Table 4. Removing the speaker loss ($\mathcal{L}_{\text{spk}}$) has little effect on reconstruction UTMOS and WER, but causes a substantial drop in SIM. Its impact is much larger

	Reconstruction			**Voice Cloning**		
Model	UTMOS	WER	SIM	UTMOS	WER	SIM
DSA-Tokenizer	3.78	2.16	0.73	4.03	2.58	0.60
w/o $\mathcal{L}_{\text{spk}}$	3.81	2.16	0.47	3.83	7.83	0.20
w/o Contextual Inpainting	3.80	2.25	0.71	2.33	134.85	0.21

Table 4: Ablation study of speaker loss and recombination mode on reconstruction and voice cloning tasks.

Model	**Steps**	**Reconstruction**			**Voice Cloning**		
		UTMOS	WER	SIM	UTMOS	WER	SIM
Original Decoder	16	3.78	2.16	0.73	4.03	2.58	0.60
4-step Distilled	4	3.87	2.18	0.70	4.04	2.49	0.56
4-step Distilled + GAN	4	**4.06**	**2.00**	**0.77**	**4.16**	**2.47**	**0.61**

Table 5: Ablation of short-step distillation and GAN refinement.

in voice cloning, where UTMOS, WER, and SIM all degrade markedly, showing that $\mathcal{L}_{\text{spk}}$ is important not only for preserving speaker identity but also for stable cross-utterance style transfer. Removing contextual inpainting has only a mild effect on reconstruction, but causes voice cloning to collapse, showing that contextual inpainting is the key training mode that enables robust voice cloning.

We further study the second-stage decoder refinement in Table 5. Distilling the original decoder from 16 steps to 4 steps preserves the overall reconstruction and voice cloning performance remarkably well: UTMOS and WER remain highly comparable to the original model, with only a moderate drop in SIM. Applying GAN fine-tuning on top of the distilled decoder further improves all metrics across both tasks, yielding the best UTMOS, WER, and SIM. In particular, the gain is not limited to perceptual quality: content-related metrics are also improved, indicating that adversarial refinement enhances synthesis fidelity without weakening semantic-acoustic controllability.

6 Conclusion

We present DSA-Tokenizer, a dual-stream speech tokenizer that explicitly disentangles speech into discrete semantic and acoustic tokens. With task-specific supervision, contextual inpainting training, DSA-Tokenizer supports both high-fidelity reconstruction and cross-utterance voice cloning. We further distill the decoder to 4-step inference and refine it with GAN fine-tuning for efficient high-quality generation. Experiments show that DSA-Tokenizer provides the cleanest semantic-acoustic disentanglement, the best overall balance between reconstruction and voice cloning at low bitrate, and a more effective token interface for Speech LLMs.

Limitations

Despite its strong performance, DSA-Tokenizer still has several limitations. First, although decoder distillation reduces inference to 4 steps, the model is still not as lightweight as one-shot codec or GAN-based generators, which may limit deployment in latency-sensitive settings. Second, our training and evaluation focus on Chinese-English speech, cross-utterance voice cloning, and LLM-based TTS; broader language coverage and more diverse downstream speech LLM tasks remain unexplored. Third, our disentanglement analysis mainly probes semantic information and speaker-related acoustic information through ASR and speaker classification, and does not fully measure other factors such as prosody or emotion. We leave these directions to future work.

References

Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, and 1 others. 2024. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, and 27 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *Preprint*, arXiv:2406.02430.

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

Hervé Bredin and Antoine Laurent. 2021. End-to-end speaker segmentation for overlap-aware resegmentation. In *Proc. Interspeech 2021*, Brno, Czech Republic.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, and 1 others. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Kai Chen, Yunhao Gou, Runhui Huang, Zhili Liu, Daxin Tan, Jing Xu, Chunwei Wang, Yi Zhu, Yihan Zeng, Kuo Yang, and 1 others. 2025a. Emova: Empowering language models to see, hear and speak with vivid emotions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5455–5466.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, and 17 others. 2025b. Minmo: A multimodal large language model for seamless voice interaction. *Preprint*, arXiv:2501.06282.

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

Wenxi Chen, Xinsheng Wang, Ruiqi Yan, Yushen Chen, Zhikang Niu, Ziyang Ma, Xiquan Li, Yuzhe Liang, Hanlin Wen, Shunshun Yin, Ming Tao, and Xie Chen. 2025c. Sac: Neural speech codec with semantic-acoustic dual-stream quantization. *Preprint*, arXiv:2510.16841.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2025d. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *Preprint*, arXiv:2410.06885.

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

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. *Preprint*, arXiv:2108.06209.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech 2020*, pages 3830–3834.

DiscreteSpeech Team. 2025. Dstk: Discrete speech tokenization toolkit. `https://huggingface.co/DiscreteSpeech/DSTK`.

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

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou.

2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *Preprint*, arXiv:2412.10117.

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

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

Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. 2021. Adversarially adaptive normalization for single domain generalization. *Preprint*, arXiv:2106.01899.

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

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. 2021. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):652–662.

Yitian Gong, Luozhijie Jin, Ruifan Deng, Dong Zhang, Xin Zhang, Qinyuan Cheng, Zhaoye Fei, Shimin Li, and Xipeng Qiu. 2025. Xy-tokenizer: Mitigating the semantic-acoustic conflict in low-bitrate speech codecs. *Preprint*, arXiv:2506.23325.

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

Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. 2025. Recent advances in discrete speech tokens: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *Preprint*, arXiv:2407.05361.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *Preprint*, arXiv:2207.12598.

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

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, Peng Liu, Ruihang Miao, Wang You, Xi Chen, Xuerui Yang, Yechang Huang, Yuxiang Zhang, Zheng Gong, Zixin Zhang, and 126 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *Preprint*, arXiv:2502.11946.

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

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2024. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995. IEEE.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, and 21 others. 2025. Kimi-audio technical report. *Preprint*, arXiv:2504.18425.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, and Emmanuel Dupoux. 2021. Generative spoken language modeling from raw audio. *Preprint*, arXiv:2102.01192.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow matching for generative modeling. *Preprint*, arXiv:2210.02747.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. *Preprint*, arXiv:1611.04076.

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

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2020. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux. 2024. Spirit lm: Interleaved spoken and written language model. *Preprint*, arXiv:2402.05755.

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

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. *Preprint*, arXiv:2212.09748.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *Preprint*, arXiv:2204.02152.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Preprint*, arXiv:1606.03498.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

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

Dehua Tao, Daxin Tan, Yu Ting Yeung, Xiao Chen, and Tan Lee. 2024. Toneunit: A speech discretization approach for tonal language speech synthesis. *arXiv preprint arXiv:2406.08989*.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. *Preprint*, arXiv:1711.11585.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, Weizhen Bian, Zhen Ye, Sitong Cheng, Ruibin Yuan, Zhixian Zhao, Xinfa Zhu, Jiahao Pan, Liumeng Xue, Pengcheng Zhu, and 6 others. 2025. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *Preprint*, arXiv:2503.01710.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025. Qwen3-omni technical report. *Preprint*, arXiv:2509.17765.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zehui Yang, Yifan Chen, Lei Luo, Runyan Yang, Lingxuan Ye, Gaofeng Cheng, Ji Xu, Yaohui Jin, Qingqing Zhang, Pengyuan Zhang, and 1 others. 2022. Open source magicdata-ramc: A rich annotated mandarin conversational (ramc) speech dataset. *arXiv preprint arXiv:2203.16844*.

Zengwei Yao, Wei Kang, Han Zhu, Liyong Guo, Lingxuan Ye, Fangjun Kuang, Weiji Zhuang, Zhaoqing Li, Zhifeng Han, Long Lin, and Daniel Povey. 2026. Flow2GAN: Hybrid flow matching and GAN with multi-resolution network for few-step high-fidelity audio generation. In *The Fourteenth International Conference on Learning Representations*.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, Hongzhan Lin, Jianyi Chen, Xingjian Du, Liumeng Xue, Yunlin Chen, Zhifei Li, Lei Xie, Qiuqiang Kong, Yike Guo, and Wei Xue. 2025. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *Preprint*, arXiv:2502.04128.

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

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, and 1 others. 2022. 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)*, pages 6182–6186. IEEE.

Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, and 1 others. 2025. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023a. Adding conditional control to text-to-image diffusion models. *Preprint*, arXiv:2302.05543.

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

Han Zhu, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhaoqing Li, Weiji Zhuang, Long Lin, and Daniel Povey. 2025. Zipvoice: Fast and high-quality zero-shot text-to-speech with flow matching. *Preprint*, arXiv:2506.13053.

A Model Architecture

A.1 SEANetEncoder

For the SEANetEncoder, we employ a four-layer architecture with dimensions of [512, 1024, 1024, 1024]. The downsampling ratios are set to [2, 1, 1] and [2, 2, 1] for acoustic token rates of 50Hz and 25Hz, respectively.

A.2 SEANetDecoder

For the SEANetDecoder, we utilize a four-layer architecture with dimensions of [1024, 1024, 1024, 1024]. The upsampling ratios are set to [1, 1, 2] and [1, 2, 2] for acoustic token rates of 50Hz and 25Hz, respectively.

A.3 DiT blocks

For the DSA-Tokenizer, we utilize a stack of 22 DiT blocks with a hidden dimension of 1024. We incorporate AdaNorm (Fan et al., 2021) for both multi-head self-attention and multi-head cross-attention mechanisms. The FFN consists of two linear layers with an inner dimension set to 4096. And RoPE(Su et al., 2023) is set as the position embedding for both multi-head self-attention and multi-head cross-attention.

A.3.1 ControlNet-Style CNN Adapter

For the ControlNet-Style CNN Adapter, we employ a lightweight stack of three 1D convolutional layers to extract contextual information while projecting the feature dimension from 512 to 1024, as the embedding dim of semantic token is 512. Each layer utilizes a kernel size of 3 and a padding of 1 to preserve the temporal sequence length.

A.4 Attention pooling layer

Let $e_a \in \mathbb{R}^{T\times D}$ denote the input acoustic sequence, where T is the sequence length and $D = 1024$ is the feature dimension. To aggregate the variable-length sequence into a fixed-size embedding, we employ a Masked Attentive Statistics Pooling (ASP) layer followed by a projection adapter.

First, the input sequence is transposed to $\mathbf{X} \in \mathbb{R}^{D\times T}$. A channel-wise attention mechanism computes the importance scores $\mathbf{S} \in \mathbb{R}^{D\times T}$ for each time step and feature dimension:

$$\mathbf{S} = \mathbf{W}_2\left(\tanh\left(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1\right)\right) + \mathbf{b}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{D_{attn}\times D}$, $\mathbf{W}_2 \in \mathbb{R}^{D\times D_{attn}}$ are convolution weights with kernel size 1, and D_{attn} is the bottleneck dimension, $\mathbf{b_1} \in \mathbb{R}^D$ and $\mathbf{b}_2 \in \mathbb{R}^D$ are bias weights.

To handle variable sequence lengths, we apply a temporal mask $M \in \{0, -\infty\}^T$ based on the valid length of the sequence. The normalized attention weights $\boldsymbol{\alpha} \in \mathbb{R}^{D\times T}$ are obtained via a masked softmax operation along the temporal dimension:

$$\alpha_{d,t} = \frac{\exp(S_{d,t} + M_t)}{\sum_{\tau=1}^{T}\exp(S_{d,\tau} + M_\tau)},$$

where $d \in [1, D]$ and $t \in [1, T]$.

Using these weights, we calculate the weighted mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and the weighted standard deviation vector $\boldsymbol{\sigma} \in \mathbb{R}^D$:

$$\boldsymbol{\mu}_d = \sum_{t=1}^{T}\alpha_{d,t}\cdot X_{d,t},$$

$$\boldsymbol{\sigma}_d = \sqrt{\sum_{t=1}^{T}\alpha_{d,t}\cdot(X_{d,t} - \boldsymbol{\mu}_d)^2 + \epsilon}$$

where ϵ is a small constant for numerical stability.

Finally, the statistics are concatenated and projected to the target dimension $D_{out} = 256$:

$$\mathbf{h} = \mathbf{W}_{proj}[\boldsymbol{\mu};\boldsymbol{\sigma}] + \mathbf{b}_{proj},$$

where $[\cdot;\cdot]$ denotes concatenation resulting in a $2D$-dimensional vector (2048), and $\mathbf{W}_{proj} \in \mathbb{R}^{D_{out}\times 2D}$, $\mathbf{b}_{proj} \in \mathbb{R}^{2D}$. The resulting $\mathbf{h}$ serves as the pooled acoustic token embeddings.

A.5 Proxy Speaker Encoder

The encoder adopts a hybrid ECAPA-Transformer architecture(Desplanques et al., 2020). The input mel is first normalized with InstanceNorm1d, followed by a Conv1d front-end and three SE-Res2 blocks blocks with dilation rates 2, 3, and 4(Gao et al., 2021). Their outputs are concatenated and mixed by 1×1 convolutions, and augmented with sinusoidal positional embeddings. We then apply 6 Transformer encoder blocks, and combine

it with the ECAPA branch through a residual connection. Finally, masked attentive statistics pooling produces a pooled feature.

The Masked attentive statistics pooling is used to aggregate variable-length frame-level features into a fixed-dimensional utterance representation. Given the input sequence and its valid lengths, a temporal mask is first constructed to exclude padded frames from attention and pooling. The global mean and standard deviation over valid frames are computed and concatenated with the original features. The attention branch then applies two 1×1 convolution layers with a tanh activation to generate frame-wise attention weights, followed by masked softmax normalization over time. Finally, the module computes a weighted mean and weighted standard deviation and concatenates them to form the pooled feature.

A.6 Codebook Size

The acoustic tokenizer uses FSQ with 8 channels and 4 levels per channel, corresponding to a codebook size of 65536. The semantic tokenizer uses FSQ with 6 channels and 4 levels per channel, corresponding to a codebook size of 4096.

B Dataset Cleaning

We find that some samples of Emilia Dataset (He et al., 2024) contains more than two speakers. As we have Recombination (Contextual Inpainting) Mode, samples contains more than two speakers will hurt the performance of our model, therefore, we utilize speaker-diarization model [2] (Bredin and Laurent, 2021) to remove those dirty samples.

C Training Details

C.1 Training Details of DSA-Tokenizer

All DSA-Tokenizer are trained witicdynamic batching with 30k frames of speech per batch. We train the tokenizer for 400k steps using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 7.5×10^{-5} and 32k warmup steps. The dataset used for the training of the semantic tokenizer is sampled from LibriSpeech (Panayotov et al., 2015), GigaSpeech (Chen et al., 2021), Libri-Heavy (Kang et al., 2024), CommonVoice (Ardila et al., 2020), AISHELL-2 (Du et al., 2018), WenetSpeech (Zhang et al., 2022), MagicData-RAMC (Yang et al., 2022), and People's Speech (Galvez et al., 2021). The DSA-Tokenizer has a parameter size of approximately 430M. We trained the models on the Ascend platform for 15 days.

C.2 Training Details of Short-Step Distillation and GAN Fine-Tuning

During both short-step distillation and GAN fine-tuning, the semantic and acoustic tokenizers are frozen, and only the decoder-side modules are updated. Following ZipVoice-style flow distillation, we initialize the student from the pretrained Flow-Matching decoder and distill it into a 4-step generator. Each batch retains the same two training modes as in the first stage, namely self-reconstruction and contextual inpainting, and both modes are forwarded during distillation. The learning rate of the first distillation stage is set to 7.5e-5. The learning rate of the second distillation stage is set to 1.5e-5. The number of training steps for each distillation stage is 10k.

For adversarial refinement, we operate directly in mel space with a multi-scale, multi-band 2D PatchGAN discriminator, rather than a waveform discriminator. The input is the log-mel spectrogram, and each sample is normalized independently before being fed into the discriminator. To improve robustness to both local spectral detail and broader time-frequency structure, we construct four frequency views: the full band $[0, 100]$ and three overlapping sub-bands $[0, 32]$, $[24, 72]$, and $[64, 100]$. Each view is further evaluated at three temporal scales $\{1, 2, 4\}$, yielding 12 mel-domain sub-discriminators in total. We use least-squares GAN (LSGAN) losses together with feature matching losses. Since self-reconstruction samples have different utterance durations and the supervised regions of contextual inpainting samples depend on the randomly sampled mask boundary, the effective mel lengths vary across a mixed batch. We therefore define the adversarial crop length as the shortest effective mel length t in the batch, and randomly crop one segment of length t from each real/generated pair before feeding it to the discriminator.

To provide differentiable speaker supervision on generated mel-spectrograms, we train a proxy speaker encoder to match the embedding space of the frozen WavLM-based speaker verification model[1]. The resulting mel-level speaker consistency loss is applied only to self-reconstruction

[2] https://huggingface.co/pyannote/speaker-diarization

samples. The proxy speaker encoder used in GAN fine-tuning is trained as a mel-to-speaker embedding regressor, rather than a speaker classifier. Given an input mel-spectrogram, the model predicts a speaker embedding aligned to the embedding space of the same frozen WavLM-based speaker verification model[1].

The proxy speaker encoder is trained with AdamW using a learning rate of 1.2×10^{-4}. The training objective combines cosine, MSE, and contrastive distillation terms:

$$\mathcal{L}_{\text{proxy}} = \mathcal{L}_{\text{cos}} + 0.1\,\mathcal{L}_{\text{mse}} + 0.1\,\mathcal{L}_{\text{nce}},$$

where $\mathcal{L}_{\text{cos}} = 1 - \cos(\hat{s}, s)$, $\mathcal{L}_{\text{mse}}$ is the MSE between normalized predicted and target embeddings, and $\mathcal{L}_{\text{nce}}$ is an InfoNCE loss.

For each discriminator branch D_k, the discriminator loss is:

$$\mathcal{L}_D^{(k)} = \mathbb{E}\left[(1 - D_k(m_{\text{real}}))^2\right] + \mathbb{E}\left[D_k(m_{\text{fake}})^2\right].$$

The generator-side adversarial and feature-matching losses are defined as:

$$\mathcal{L}_{\text{adv}}^{\text{mel}} = \sum_k \mathbb{E}\left[(1 - D_k(m_{\text{fake}}))^2\right],$$

$$\mathcal{L}_{\text{feat}}^{\text{mel}} = \sum_k \sum_l \left\| f_k^{(l)}(m_{\text{fake}}) - f_k^{(l)}(m_{\text{real}}) \right\|_1,$$

where $f_k^{(l)}(\cdot)$ denotes the l-th intermediate feature of the k-th sub-discriminator. The mel-level speaker consistency loss is

$$\mathcal{L}_{\text{spk}}^{\text{mel}} = 1 - \cos\left(h_\phi(\hat{m}), s_{\text{ref}}\right),$$

where $\hat{m}$ is the mel-spectrogram generated by the 4-step decoder, $h_\phi(\cdot)$ is the proxy speaker encoder, and s_{ref} is the same reference speaker embedding used in the first-stage speaker consistency loss. The final generator objective during GAN fine-tuning is

$$\mathcal{L}_G = \mathcal{L}_{\text{total}} + 0.025\,\mathcal{L}_{\text{adv}}^{\text{mel}} + 0.05\,\mathcal{L}_{\text{feat}}^{\text{mel}} + 0.5\,\mathcal{L}_{\text{spk}}^{\text{mel}}.$$

The learning rate of discriminator is set to 1.0e-4. The learning rate of generator is set to 1.5e-5. The total step of gan finetuning stage is 10k.

C.3 Training Details of Disentanglement Probing

In the disentanglement probing experiments, tokens extracted from the tokenizers are first mapped to 128-dimensional embeddings, followed by processing via a two-layer CNN and a 256-dimensional bidirectional LSTM. For the scenario where multi-layer tokens are used as input, the token embeddings of all layers are first summed before being processed by the CNN and LSTM.

For the ASR experiment, English character sequences (including space symbols) are used as training targets, and the sequential output of the LSTM is optimized against these sequences using CTC loss. In the speaker classification experiment, the LSTM output is aggregated by a pooling layer and then mapped to speaker IDs through an MLP layer, with optimization performed using cross-entropy loss.

For dataset configurations: the LibriSpeech train-960 subset is used for training, dev-clean for validation, and test-clean for testing in the ASR experiment. In the speaker classification task, the VoxCeleb1 train, dev, and test subsets serve as the training, validation, and testing sets, respectively. All models are trained for 30 epochs, with the model achieving the lowest validation WER (for ASR) or the highest speaker classification accuracy (for speaker classification) selected for testing and metric calculation. The LibriSpeech training set contains nearly 1000 hours of speech data, while VoxCeleb1 includes the same 1251 speakers in both its training and testing sets.

C.4 Training Details of LLM-based Voice Cloning

In LLM-based voice cloning tasks, Qwen3-0.6B serves as the LLM backbone. Speech symbols that depend on the tokenizer's codebook size and number of layers are first integrated into the LLM's vocabulary, with adjustments made to the LLM's embedding table and prediction head. We train the LLM using the supervised fine-tuning (SFT) paradigm, where only the loss of the response part is computed. The model is trained for 5 epochs with a learning rate of 1e-5 and a warmup ratio of 0.1.

C.5 Training Details of LLM-based TTS

In LLM-based voice cloning tasks, Qwen3-0.6B serves as the LLM backbone. Speech symbols that depend on the tokenizer's codebook size and number of layers are first integrated into the LLM's vocabulary, with adjustments made to the LLM's embedding table and prediction head. We train the LLM using the supervised fine-tuning (SFT) paradigm, where only the loss of the response part

is computed. The model is trained for 1 epochs with a learning rate of 5e-5 and a warmup ratio of 0.03.

D Baseline Details

Four representative types of baseline models are selected for fair comparison: WavTokenizer (single-layer, non-decoupled), SAC and DualCodec (dual-branch, decoupled), Mimi and Encodec (multi-layer, non-decoupled), and SpeechTokenizer (multi-layer, decoupled) Cosy2 S^3 (ASR-supervised captures semantics).

WavTokenizer is built on the VQ-GAN framework and utilizes a single VQ layer with a codebook size of 4096. Operating at a frame rate of 75 Hz, it achieves a bitrate of 0.9 kbps.

SAC is also built on the VQ-GAN framework. It uses a pretrained semantic tokenizer, while utilizing the DAC framework for the acoustic stream with a total frame rate of 62.5 Hz. The semantic codebook size is 16,384 and the acoustic codebook size is 16,384, achieving a bitrate of 0.875 kbps.

DualCodec is a semantic-enhanced tokenizer that directly encodes semantic-rich SSL features (Chung et al., 2021) into its first layer. In our experimental setting, for a fair comparison, we select the model with six codebook layers that operates at the frame rate of 12.5Hz. The first codebook contains 16,384 entries, while the remaining five each contain 4,096 entries, achieving a bitrate of 0.925 kbps.

Mimi utilizes features from WavLM for semantic distillation. It employs eight codebooks, each of size 2,048, at a 12.5 Hz frame rate, resulting in a bitrate of 1.1 kbps.

Encodec is an RVQ-based neural audio codec operating at a frame rate of 75 Hz. In our experimental setting, for a fair comparison, we use only the first two codebook layers, yielding a bitrate of 1.5 kbps.

SpeechTokenizer shares a similar architecture with Mini but enhances the semantic richness of the first-layer tokens via semantic distillation using HuBERT. In our experimental setting, for a fair comparison, we use only the first two codebook layers, yielding a bitrate of 1 kbps.

CosyVoice2 S^3 Tokenizer inserts an FSQ module into the encoder of the SenseVoice-Large ASR model (An et al., 2024), which discretizes intermediate continuous representations into discrete tokens with textual supervision. This FSQ module consists of a single-layer FSQ codebook with 8 channels and 3 levels per channel, resulting in a total of 6561 code entries. With a token rate of 25 Hz, this tokenizer achieves a bitrate of approximately 0.317 kbps.

Facodec introduces a novel factorized vector quantization (FVQ) mechanism to decompose complex speech waveforms into distinct, disentangled subspaces representing content, prosody, timbre, and acoustic details. It enhances the disentanglement of these specific attributes through targeted techniques including information bottlenecks, auxiliary supervision, gradient reversal layers, and detail dropout. In our experimental setting, we utilize a total of 6 codebook layers—specifically allocating 2 for content, 1 for prosody, and 3 for acoustic details—with a codebook size of 1024, and the timbre vector.

E Additional Experiment Setup Details

E.1 Dataset Details for Downstream Speech LLMs

For LLM-based voice cloning, we construct more than 350,000 training triplets from the training set of LibriTTS, using f5-tts and cosyvoice2. In each triplet $(\mathcal{S}, \mathcal{A}, \mathcal{C})$, $\mathcal{S}$ and $\mathcal{A}$ are taken from different utterances, and $\mathcal{C}$ is synthesized by two TTS models as the supervised target. For evaluation, we derive more than 4,000 randomly matched test pairs from LibriTTS test-clean.

For LLM-based TTS, we follow the same data distribution as VoxBox (Wang et al., 2025). This setting is used to study whether different tokenization schemes provide more effective interfaces for autoregressive speech generation.

E.2 Downstream Speech LLM Task Formatting

For LLM-based voice cloning, each speech utterance is represented by semantic tokens and acoustic tokens. The autoregressive model is trained on triplets $(\mathcal{S}, \mathcal{A}, \mathcal{C})$, where $\mathcal{S}$ provides semantic content, $\mathcal{A}$ provides acoustic style, and $\mathcal{C}$ denotes the target cloned speech. The model takes the [SEP]-separated concatenation of $\mathcal{S}$ and $\mathcal{A}$ as input and

Model	UTMOS	WER	SIM	UTMOS_{zh}	CER_{zh}	SIM_{zh}
CosyVoice1	3.55	3.24	0.40	3.00	2.94	0.60
CosyVoice2	3.93	3.48	0.52	3.28	2.91	0.72
CosyVoice3	3.73	2.74	0.56	3.10	2.21	**0.75**
DSA-Tokenizer	**4.16**	**2.47**	**0.61**	**3.55**	**2.16**	0.71

Table 6: Additional comparison with the CosyVoice series on cross-utterance voice cloning.

predicts the target token sequence $\mathcal{C}$. During inference, the model generates token sequences for unseen $(\mathcal{S}', \mathcal{A}')$ pairs, which are then decoded into waveform by the tokenizer decoder.

For TTS, we adopt a pure LLaSA-style autoregressive architecture (Ye et al., 2025). The model generates semantic and acoustic token sequences conditioned on the input text prompt together with the semantic and acoustic tokens extracted from the prompt speech. The generated token sequences are then decoded into speech waveform by the tokenizer decoder.

For both downstream tasks, the semantic and acoustic tokens of DSA-Tokenizer and SAC are interleaved in the same sequence.

E.3 Additional Baseline Notes for Downstream Speech LLMs

For both LLM-based voice cloning and TTS, we use Qwen3-0.6B (Yang et al., 2025) as the autoregressive backbone and expand its vocabulary with the corresponding speech tokens.

In LLM-based voice cloning, we compare WavTokenizer, SAC, and DSA-Tokenizer to analyze how non-disentangled and disentangled token interfaces affect cross-utterance generation.

In LLM-based TTS, the compared systems include LLaSA-1B-80k, LLaSA-1B-160k, LLaSA-1B-250k, Spark-TTS, SAC, and DSA-Tokenizer. Among them, the LLaSA and Spark-TTS systems use non-disentangled tokenizers, while SAC and DSA-Tokenizer represent disentangled tokenization approaches.

F Additional Comparison with the CosyVoice Series

We additionally compare DSA-Tokenizer with the CosyVoice series as strong system-level voice cloning baselines.

Table 6 shows that, using continuous features, the CosyVoice systems are still strong baselines. DSA-Tokenizer nevertheless outperforms all three CosyVoice systems on English UTMOS, WER, and SIM. On Chinese, it achieves the best CER and UTMOS while remaining competitive in SIM.

Model	ASR WER (%) ↓	SC ACC (%) ↑
Cosy2 S^3 Tokenizer	33.72	14.50
WavTokenizer	86.31	18.14
EnCodec		
Layer 0	64.85	16.28
Layer 1-7	69.74	17.68
SpeechTokenizer		
Layer 0	17.78	12.08
Layer 1-7	21.93	25.80
SAC		
Semantic token	9.31	6.57
Acoustic token	106.24	21.82
Facodec		
Semantic token	13.5	7.13
Acoustic token	100.13	22.95
DSA-Tokenizer		
Semantic token	6.28	2.35
Acoustic token	120.36	23.65

Table 7: Disentanglement probing evaluation results

G Disentanglement Probing Result

The detailed result of disentanglement probing is listed in Table 7.

H Evaluation tools

The UTMOS score is computed using the pretrained model available at [3]. Speaker similarity (SIM) scores are calculated with a speaker encoder based on WavLM, fine-tuned for the speaker verification task1. For English speech, the word error rate (WER) is evaluated with Whisper-large-V3[4], and for Chinese speech, the character error rate (CER) is evaluated with Paraformer[5].

I Broader Impact

This work introduces a novel dual-stream audio tokenizer capable of disentangling semantic and acoustic information. Our research holds significant potential for positive societal impact, particularly in the development of Speech LLMs. The disentangled representation offers fine-grained control for speech. However, the capability to clone voices with high similarity raises concerns regarding misuse for deepfakes, unauthorized voice imperson-

[3] https://huggingface.co/spaces/sarulab-speech/UTMOS-demo/tree/main

[4] https://huggingface.co/openai/whisper-large-v3

[5] https://huggingface.co/funasr/paraformer-zh

ation, and misinformation.To prevent abuse, we recommend developing robust deepfake detection tools, speech watermarking, and clear reporting mechanisms.

J License Discussion

The speaker-diarization model(Bredin and Laurent, 2021) and the model[3] used to calculate the UTMOS score are released under the MIT license. The speaker verification fine-tuned WavLM1 is released under the CC BY-SA 3.0 license. Whisper-large-V3[4] is released under the Apache 2.0 license. And the model[5] calculating CER is released under model-license license.

The Emilia dataset, LibriSpeech, MagicData-RAMC and LibriTTS are under the license of cc-by-4.0. GigaSpeech, Libri-Heavy, AISHELL-2, WenetSpeech are released under the Apache 2.0 License. CommonVoice is released under the MPL-2.0 license. MagicData-RAMC and VoxCeleb1 are under the license CC BY-NC-ND 4.0. People's Speech is under the license of CC-BY-SA and CC-BY 4.0.

K AI Usage

We used LLMs for grammatical checking and polishing in Sections 1, section 5 and the Appendix.