SPIDERAUDIO: UNIFIED BINAURAL REPRESENTATION FOR JOINT SPATIAL SPEECH UNDERSTANDING, GENERATION AND EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in unified continuous representations for joint monaural speech understanding, generation, and even editing have shown promising results. Extending these representations to spatial speech tasks requires not only modeling the semantic and acoustic information, but also preserving spatial state information. Existing spatial representations, however, are designed for a single capability: some take hand-crafted interaural features as input and do not support waveform synthesis, while others optimize reconstruction fidelity rather than compatibility with a language model. To this end, we propose *SpiderTok*, a single all-round speech **tok**enizer that integrates semantic, acoustic, and spatial features in binaural speech for unified **sp**at**i**al understan**d**ing, gene**r**ation and **e**diting. SpiderTok simultaneously encodes 16 kHz binaural speech into a compact 50 Hz unified continuous representation. Its multi-stage training mainly includes acoustic and spatial reconstruction, semantic distillation, and joint refinement of unified representations. Building on this tokenizer, we further develop *SpiderAudio*, an all-round large speech language model that supports joint binaural speech understanding, generation, and editing within a single framework. In addition, we propose a "*Thinking-Before-Generation*" strategy to guide binaural spatial speech editing. Experiments show that SpiderTok reconstructs binaural speech with high fidelity while outperforming binaural baselines on all four spatial metrics, with up to 23% relative gain in angular cosine similarity, and that the unified representation lets SpiderAudio surpass spatial baselines on understanding across static, moving, and orbiting sources, while also supporting instruction-guided spatial editing [1].

1 INTRODUCTION

Human listeners can recognize what a voice says and hear where it comes from. Such grounding relies on acoustic cues, including interaural time differences (ITDs), interaural level differences (ILDs), and direction-dependent spectral cues introduced by the outer ears and head (Blauert, 1996). These cues reside in binaural speech. Existing speech large language models unify understanding and generation (Li et al., 2026a), and some extend further to editing (Yan et al., 2025a). However, the majority of them are built on monaural speech, which leaves spatial tasks out of reach. Many real applications require spatial processing abilities. For example, embodied human-robot interaction must recognize where a voice comes from; cross-lingual dubbing must also synthesize a voice at a position read from the original track; producing an audiobook or a radio drama must further revise a recording without disturbing where it sounds.

Existing spatial large speech language models mainly target a single capability. On the understanding side, a spatial encoder is paired with a language model to answer questions about direction and distance (Zheng et al., 2024; Biswas et al., 2026; Wilkinghoff & Tan, 2026). Such encoders take hand-crafted interaural phase differences (IPDs) as input, and their objectives ask only for labels, never for a synthesized waveform. On the generation side, latent representations are learned only for conditional spatial synthesis in the binaural domain (Zhao et al., 2026) or in first-order Ambisonics (FOA) (Lei et al., 2026). Besides, neural spatial codecs, whether binaural (Ratnarajah et al., 2025)

[1] The demo page is: `https://conf123456789.github.io/iclr2027`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

or FOA (Sudarsanam et al., 2025), optimize reconstruction at a given bitrate rather than a unified representation that could serve different spatial tasks within a speech language model.



Figure 1: Three binaural spatial speech tasks covered by SpiderAudio and their sub-tasks.

To this end, we aim to unify spatial speech understanding and generation in a single model, which in turn makes editing attainable. Since all three tasks are defined at the ears, we model binaural speech directly rather than decoding it from a scene-based format such as FOA (Zotter & Frank, 2019). (Appendix A compares the two formats in details.) **Two challenges follow**. The **first** is how to build a spatial variational autoencoder (VAE) whose latent space retains acoustic detail and interaural structure for high-fidelity reconstruction while also carrying semantic information. The **second** is how to integrate such a unified representation into a large speech language model so that a single latent sequence supports spatial speech understanding, generation, and editing simultaneously.

To address these challenges, we propose *SpiderTok*, an all-round speech tokenizer that integrates semantic, acoustic, and spatial features in binaural speech for unified spatial understanding, generation and editing. SpiderTok simultaneously encodes 16 kHz binaural speech into a compact 50 Hz unified continuous representation. Its multi-stage training combines binaural reconstruction, in which interaural phase and level differences are explicit supervision targets, with semantic distillation and joint refinement of the unified representation. Building on this tokenizer, we further develop *SpiderAudio*, an all-round large speech language model that supports joint binaural speech understanding, generation, and editing within a single framework. In addition, we propose a "*Thinking-Before-Generation*" strategy to guide binaural spatial speech editing.

Experiment shows that: **1)** on the tokenizer side, SpiderTok reconstructs binaural speech with high fidelity and outperforms the strongest binaural baseline on all four spatial metrics (angular and distance cosine similarity, interaural phase and level correlation), with the largest gain in angular cosine similarity (0.913 against 0.742). **2)** Built on this representation, SpiderAudio delivers three capabilities. **First**, it surpasses the strongest spatial understanding baseline across static, moving, and orbiting sources, while retaining ASR word error rate (WER) comparable to unified monaural models. **Second**, in spatial text-to-speech, it attains the highest angular and distance accuracy reported in this setting, while retaining competitive ASR WER and timbre similarity. **Third**, it supports instruction-guided spatial editing, where the "*Thinking-Before-Generation*" strategy improves spatial preservation over direct generation. The main contributions are summarized as follows:

1) We present **SpiderTok**, a binaural spatial speech tokenizer that encodes semantic, acoustic, and spatial information in a unified representation. To our knowledge, this is the first tokenizer to serve spatial understanding, generation, and editing within the binaural domain.

2) We develop **SpiderAudio**, a large speech language model in which this unified representation supports spatial understanding, generation and editing, trained by a understanding warm-up for cross-modal alignment followed by joint multitask training, and evaluated on all three tasks under static, moving and orbiting conditions.

3) We introduce a "*Thinking-Before-Generation*" strategy for spatial editing, which improves spatial preservation over direct generation without degrading content or prosody, and report the first spatial speech editing results in this setting.

4) We construct **SpiderData**, all-round binaural speech training data for joint binaural speech understanding, generation and editing tasks.

2 Related Work

Monaural unified representations for speech understanding and generation. Discrete neural codecs made speech tractable for language models by quantising waveforms into token sequences (Défossez et al., 2022; Kumar et al., 2023; Ji et al., 2025; Ye et al., 2025; Boson, 2025). However, quantisation is inherently lossy, and Ming-UniAudio (Yan et al., 2025a) identify the discrete nature of such tokens as the reason they remain suboptimal for understanding. The field has therefore moved towards continuous latents (Li et al., 2026a; Zhang et al., 2025). HoliTok reports that continuous representations with comparable reconstruction quality diverge sharply once generation and understanding are trained together (Li et al., 2026a), and Ming-UniAudio couples compact latents to semantic features so that one speech language model can both transcribe and synthesise (Yan et al., 2025a). Unified audio models have been scaled up (Yang et al., 2026; Li et al., 2026b; Xu et al., 2025; Kong et al., 2026), and instruction-guided editing is by now established among them (Yan et al., 2025a; Ma et al., 2026; Li et al., 2026c; Yan et al., 2025b). Motivated by the unified representation in monaural speech, we seek one for binaural speech, where the added difficulty is that the latent must also retain spatial information carried between the two channels.

Spatial audio understanding. BAT pairs a binaural spatial encoder with a language model to answer questions about the direction and distance of sound sources (Zheng et al., 2024). OWL adds geometry-aware supervision and stepwise reasoning (Biswas et al., 2026), and DSpAST disentangles the encoder's representation into components for detection, direction and distance (Wilkinghoff & Tan, 2026). SALM instead aligns audio embeddings with text by contrastive learning, using a separate branch to capture direction (Hu et al., 2025). All four compute spatial features by hand before encoding: BAT and OWL take interaural phase differences, DSpAST adds level differences and cross-correlation, and SALM takes intensity vectors from its Ambisonic input. Their objectives ask only for labels or embedding alignment, never for a synthesized waveform.

Spatial audio generation and coding. On the generation side, DualSpec (Zhao et al., 2026) compresses mel and short-time Fourier transform spectrograms with separate variational autoencoders and combines their latents for text-conditioned binaural synthesis, using the two feature types to address audio quality and directional accuracy, respectively. SwanSphere learns continuous first-order Ambisonics latents and generates with an autoregressive diffusion transformer (Lei et al., 2026). Neither is evaluated on understanding or editing. Neural spatial codecs pursue a different objective: BANC compresses binaural speech by coding content and spatial cues on separate branches (Ratnarajah et al., 2025), and FOA Tokenizer extends single-codebook tokenisation to four channels under a spatial consistency loss, additionally testing its latents on sound event localisation and detection (Sudarsanam et al., 2025). Both optimise reconstruction at a given bitrate, and neither is coupled to a language model. Our work differs from both groups. We instead build a single binaural latent sequence that a speech language model reads for understanding, predicts for generation, and modifies for editing.

3 Method

3.1 SpiderTok

3.1.1 Tokenizer Architecture

StereoEncoder The StereoEncoder mainly consists of a Waveform Embedding layer, Linear layers, and Causal Multi-head Self-attention (CausalMHA) layers. The Frame layer first splits each channel of the binaural waveform $\mathbf{X} = (\mathbf{X_L}, \mathbf{X_R}) \in \mathbb{R}^{2\times N}$ with N samples per channel into T frames, and interleaves them as a single stream $\mathbf{W} = (\mathbf{X_L^1}, \mathbf{X_R^1}, \ldots, \mathbf{X_L^T}, \mathbf{X_R^T})$. Then, $\mathbf{W}$ is fed into the StereoEncoder to obtain a 100 Hz (50 Hz for each channel) acoustic latent $\mathbf{Z}_{\text{acoustic}}$.

CausalSemantic Bottleneck. Following MingTok-Audio (Yan et al., 2025a), we introduce a semantic bottleneck that extracts semantic information from $\mathbf{Z}_{\text{acoustic}}$ to produce a 50 Hz unified representation $\mathbf{Z}_{\text{uni}}$. The CausalSemantic Bottleneck inherits the main structure of the Whisper-large-v3 encoder[2] but removes the front convolutional layers. In addition, all bidirectional attention is replaced with unidirectional causal attention to serve downstream streaming generation tasks.

StereoDecoder. The StereoDecoder inverts this process to reconstruct the two channels, and consists of a Back Projector layer, CausalMHA layers, two Linear layers and two HiFiGAN-based

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

VocoderHeads. The Back Projector first maps the $\mathbf{Z}_{\text{uni}}$ back to an interleaved sequence of $2T$ frames (50 Hz for each channel), restoring the layout produced by the Frame layer. The CausalMHA layers then operate on this sequence while the channels are still interleaved, which is where the interaural relation is resolved, since the channels no longer interact once separated. Two Linear layers subsequently de-interleave the sequence into one T-frame sequence per channel, and two VocoderHeads synthesize the left and right waveforms, giving the reconstructed binaural waveform. More details on each module and the shapes of intermediate representations are given in Appendix C.1.

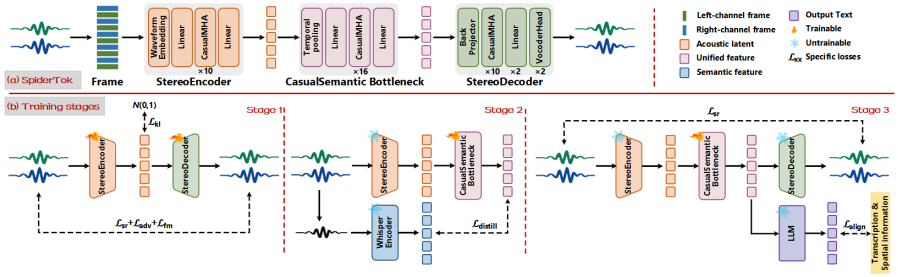


Figure 2: Architecture of SpiderTok and its three-stage training: 1) acoustic and spatial reconstruction, 2) semantic distillation, and 3) joint refinement of acoustic, semantic and spatial attributes.

3.1.2 Three-Stage Training of SpiderTok

As shown in Figure 2, three-stage training includes **1)** acoustic and spatial reconstruction training of stage 1, **2)** semantic distillation training of stage 2, and **3)** joint refinement training of acoustic, semantic and spatial attributes of stage 3.

Stage 1: Acoustic and Spatial Reconstruction. This stage involves training an binaural VAE tokenizer, which consists of a StereoEncoder and a StereoDecoder, to learn a high-quality acoustic representation of the binaural waveform. The overall loss is defined as $\mathcal{L}_1 = \lambda_{\text{sr}} \cdot \mathcal{L}_{\text{sr}} + \lambda_{\text{adv}} \cdot \mathcal{L}_{\text{adv}} + \lambda_{\text{fm}} \cdot \mathcal{L}_{\text{fm}} + \lambda_{\text{kl}} \cdot \mathcal{L}_{\text{kl}}$, Where $\lambda_{sr} = 1.0$, $\lambda_{\text{adv}} = \lambda_{\text{fm}} = 0.1$ and $\lambda_{\text{kl}} = 1e^{-6}$. $\mathcal{L}_{\text{sr}}$ denotes the spatial reconstruction loss comprising six terms as defined in Equation 1, ensuring the reconstructed binaural speech close to the original in speech quality, clarity, and spatial perception. $\mathcal{L}_{\text{adv}}$ denotes the adversarial loss and $\mathcal{L}_{\text{fm}}$ is the feature matching loss. Both $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{fm}}$ adopt the HiFi-GAN style, with the goal of remedying the speech texture distortion caused by unnatural phase recovery. $\mathcal{L}_{\text{kl}}$ is the Kullback-Leibler divergence loss that regularizes the latent space towards the prior Gaussian distribution. Except for the terms that compare the two channels, every loss is computed per channel and summed.

$$\mathcal{L}_{\text{sr}} = \lambda_{\text{mag}} \cdot \mathcal{L}_{\text{mag}} + \lambda_{\text{sc}} \cdot \mathcal{L}_{\text{sc}} + \lambda_{\text{ph}} \cdot \mathcal{L}_{\text{ph}} + \lambda_{\text{if}} \cdot \mathcal{L}_{\text{if}} + \lambda_{\text{ipd}} \cdot \mathcal{L}_{\text{ipd}} + \lambda_{\text{e}} \cdot \mathcal{L}_{\text{e}} \tag{1}$$

The first four terms act within each channel: the log-magnitude loss $\mathcal{L}_{\text{mag}}$ for envelope shifts, the spectral convergence loss $\mathcal{L}_{\text{sc}}$ for severe local distortion, the complex spectral convergence loss $\mathcal{L}_{\text{ph}}$ for absolute phase error, and the instantaneous frequency loss $\mathcal{L}_{\text{if}}$ for phase discontinuity. The remaining two act between the channels: the interaural phase difference loss $\mathcal{L}_{\text{ipd}}$ mainly constraint the interaural phase difference (IPD) and interaural time difference (ITD), while stereo energy loss $\mathcal{L}_{\text{e}}$ mainly supervises the interaural level difference (ILD) and the overall energy. For the computation of these losses, please refer to Appendix C.2.1.

Stage 2: Semantic Distillation. The CausalSemantic Bottleneck layers inherited from Whisper were pre-trained on the features produced by the convolutional front end, and now receive $\mathbf{Z}_{\text{acoustic}}$ instead. We therefore distill from the same Whisper large-v3 encoder, kept frozen, to restore their semantic behaviour under this new input. The Whisper teacher is fed the monaural speech downmixed by binaural speech. The distillation loss is computed between $\mathbf{Z}_{\text{uni}}$ and semantic teacher representations $\mathbf{Z}_{\text{semantic}}$. Details can are given in Appendix C.2.2.

Stage 3: Joint Refinement of Acoustic, Semantic, and Spatial Attributes. This stage further updates the semantic bottleneck only, where we perform a joint supervised training on LLM-based semantic alignment and spatial speech reconstruction. Semantic alignment uses Qwen2.5-0.5B[3] as

[3]https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

the LLM backbone and performs on ASR and spatial understanding tasks. The loss function is $\mathcal{L}_3 = \lambda_{\text{sr}} \cdot \mathcal{L}_{\text{sr}} + \lambda_{\text{align}} \cdot \mathcal{L}_{\text{align}}$, where $\lambda_{\text{sr}} = 1.0$, and $\lambda_{\text{align}} = 2.0$. $\mathcal{L}_{\text{sr}}$ refers to Equation 1, and $\mathcal{L}_{\text{align}}$ is semantic alignment loss. More training details on three stages are given in Appendix C.2.

3.2 SPIDERAUDIO

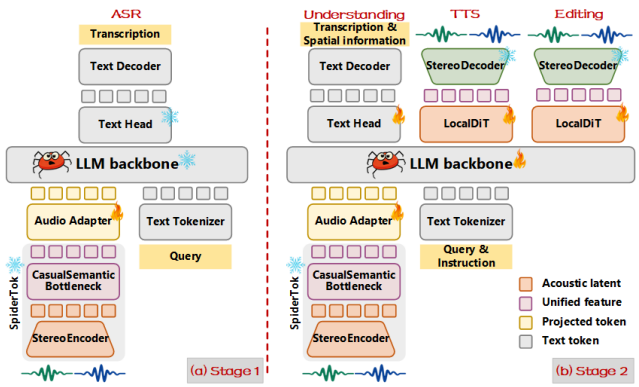


Figure 3: Architecture of SpiderAudio and its two-stage training: 1) a warm-up speech understanding training for cross-modal alignment, and 2) multi-task joint training.

3.2.1 TASK DEFINITION

The binaural input is $\mathbf{X}_{\text{in}} = (\mathbf{X}_{\text{L}}, \mathbf{X}_{\text{R}}) \in \mathbb{R}^{2\times N_1}$ with N_1 samples per channel. When used for generation or editing, it serves as a reference. Given a query or instruction prompt $\mathbf{Q}$, the end-to-end large speech language model uniformly outputs in the format of "`<text>...</text><audio>...`", and the special token `<audio>` determines the point at which decoding switches from the text decoder to the audio decoder in the streaming decoding process as illustrated in Algorithm 1. The final output text $\mathbf{Y}$ or binaural speech $\mathbf{X}_{\text{gen}} = (\hat{\mathbf{X}}_{\text{L}}, \hat{\mathbf{X}}_{\text{R}}) \in \mathbb{R}^{2\times N_2}$ can be decoded from the generated structured response streaming $\mathbf{O}$ according to the specific downstream task as follows $\mathbf{O} = [\texttt{<text>}\mathbf{Y}\texttt{</text><audio>}\mathbf{X}_{\text{gen}}] = \text{SpiderAudio}(\mathbf{X}_{\text{in}}, \mathbf{Q})$, where **1) Understanding tasks** first emit text, either a transcription or a spatial description, and then the stop head halts decoding as soon as the first audio frame would be produced, with only text retained. **2) Generation tasks** emit `</text>` immediately and produce binaural speech alone. **3) Editing tasks** emit both text and speech. The model first decodes the text that specifies the spatial state of the binaural speech, and then determines the modifications required before generating the final binaural speech. We refer to this as the "*Thinking-Before-Generation*" scheme. In addition, for the binaural speech rendering task, the monaural input is duplicated into both channels.

3.2.2 MODEL ARCHITECTURE

As illustrated in Figure 3, SpiderAudio includes four core modules: SpiderTok, Audio Adapter, LLM Backbone with the corresponding Text Tokenizer and Decoder, and DiT-based Audio Head (i.e., localDiT). **1) Audio Adapter.** The Audio adapter consists of a Pooling layer followed by two Linear layers. It projects the unified representations $\mathbf{Z}_{\text{uni}}$ into the language model's embedding space before they are concatenated with the text prompt embeddings, reducing the frame rate by a factor of five. **2) LLM Backbone.** The LLM backbone, along with its corresponding text tokenizer and text decoder, is based on Qwen2.5-7B[4]. It autoregressively predicts the text or audio tokens. **3) DiT-based Audio Head.** Since the LLM backbone cannot directly produce features of sufficient quality for waveform decoding, we follow Ming-UniAudio (Yan et al., 2025a) and SemaVoice (Wang et al., 2026) in attaching a diffusion head, and adopt a localDiT that refines the backbone's output into the unified representation $\hat{\mathbf{Z}}_{\text{uni}}$, which the StereoDecoder decodes to the binaural speech. More details on each module and the shapes of intermediate representations are given in Appendix D

3.2.3 TWO-STAGE TRAINING OF SPIDERAUDIO

The training procedure of SpiderAudio mainly consists of two stages: 1) a warm-up speech understanding training for cross-modal alignment, and 2) multi-task joint training.

[4]https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Stage 1: Cross-Modal Alignment. We train SpiderAudio on a portion of the ASR data. During this process, only the Audio Adapter updates its parameters from scratch, while the parameters of all other modules that are initialized with pretrained weights are frozen. The loss function is computed between the predictions and the ground-truth transcriptions via cross-entropy loss (i.e., $\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{ce}}$).

Stage 2: Multi-Task Joint Training. Since it is challenging for a single backbone to emit text and audio simultaneously from the outset, we divide this stage in two. The first sub-stage mixes understanding data (spatial understanding and ASR) with generation data (zero-shot and instruct spatial TTS) at a ratio of $1:3$. The second sub-stage adds spatial speech editing and mixes understanding, generation, and editing data at $1:1:2$. The loss function is defined as $\mathcal{L}_{\mathrm{II}} = \lambda_{\mathrm{ce}}\mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{flow}}\mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{stop}}\mathcal{L}_{\mathrm{stop}}$, , where $\lambda_{\mathrm{ce}} = \lambda_{\mathrm{flow}} = 1.0$ and $\lambda_{\mathrm{stop}} = 0.01$. $\mathcal{L}_{\mathrm{ce}}$ is the cross-entropy loss of the LLM for understanding task, while $\mathcal{L}_{\mathrm{flow}}$ and $\mathcal{L}_{\mathrm{stop}}$ are used for generation process. $\mathcal{L}_{\mathrm{flow}}$ is a flow matching objective. $\mathcal{L}_{\mathrm{stop}}$ teaches the model when to halt audio generation. More training details on these two training stages can be found in Appendix D.2.

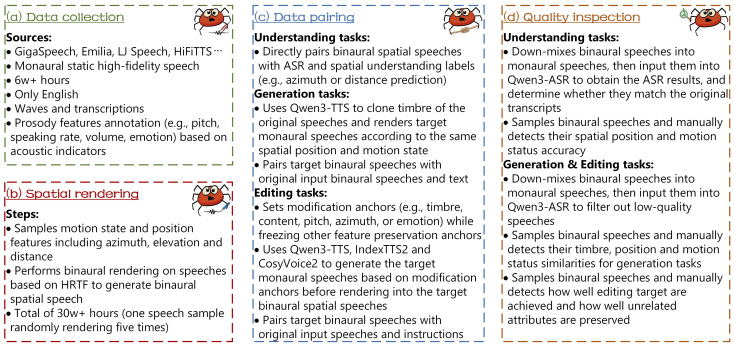


Figure 4: Illustration of data construction of SpiderData, all-round binaural speech training data for joint binaural speech understanding, generation and editing tasks.

4 SPIDERDATA

Following the previous studies (Zheng et al., 2024; Biswas et al., 2026; Wilkinghoff & Tan, 2026), we construct a unified binaural speech understanding, generation and editing training corpus, namely *SpiderData*, by rendering binaural speech that accounts for the head-related transfer function (HRTF) effects from monaural speech. Rendering uses measured KEMAR head-related impulse responses (HRIRs) stored in SOFA format (Gardner & Martin, 1995), comprising $D = 710$ measurement directions (α_j, ϵ_j) with azimuth $\alpha_j \in [0°, 355°]$ and elevation $\epsilon_j \in [-40°, 90°]$. The monaural speech before rendering is all sourced from open-source English subsets in speech corpora, including Emilia (He et al., 2024), GigaSpeech (Chen et al., 2021), LJSpeech (Negroni et al., 2025), HiFiTTS [5], LibriTTS-R (Koizumi et al., 2023), VoxPopuli (Wang et al., 2021), NonverbalTTS (Borisov et al., 2025), RAVDESS (Livingstone & Russo, 2018), and IEMOCAP (Busso et al., 2008). The total duration of these monaural data is over 60k hours.

As illustrated in Figure 4, we systematically construct task-specific inputs and labels for different downstream tasks. **1) Understanding tasks:** Spatial position and motion state parameters are first sampled as the predefined labels, based on which the corresponding input binaural speech is then rendered. **2) Generation tasks:** For zero-shot spatial TTS, we first use Qwen3-TTS[6] to clone the timbre of the reference speech, and then sample the same spatial position and motion state parameters for rendering, obtaining the reference and target binaural speech required. For instruct spatial TTS, we use Gemini-3-Flash to annotate the style parameters of the reference monaural speech before binaural speech rendering. **3) Editing tasks:** For conventional content, prosody, emotion, and paralinguistic editing, we first construct paired reference and target monaural speech, and then render them with the same sampled spatial parameters. For spatial state editing, we sample different spatial parameters, construct the instruction, and then perform rendering. For binaural speech rendering, the monaural input is copied to both channels, and the target is rendered via HRTF from the sampled spatial position and motion state. We use Qwen3-ASR (Shi et al., 2026)

[5] https://huggingface.co/datasets/MikhailT/hifi-tts

[6] https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-Base

Table 1: Reconstruction performance comparison of SpiderTok and open-source tokenizers. TPS denotes "token per second". "*" denotes reconstruction evaluation on monaural speech downmixed from the same binaural test set and open-source monaural speech models. "**" denotes binaural tokenizers retrained on our training data with unchanged architectures.

Model	TPS	Speech Quality			Spatial Fidelity			
		PESQ↑	STOI↑	UTMOS↑	ANG Cos↑	DIS Cos↑	IPD Corr↑	ILD Corr↑
Ground Truth*	-	-	-	3.916	-	-	-	-
Monaural audio tokenizer								
MiMo-Audio Tokenizer*	25	1.119	0.417	1.178	-	-	-	-
Higgs Tokenizer*	25	3.137	0.950	3.676	-	-	-	-
HoliTok*	25	**4.157**	**0.984**	**3.955**	-	-	-	-
MingTok-Audio*	50	<u>3.904</u>	<u>0.982</u>	<u>3.779</u>	-	-	-	-
X-Codec2*	50	1.999	0.902	3.742	-	-	-	-
WavTokenizer*	75	1.695	0.873	3.103	-	-	-	-
Encodec*	75	2.279	0.949	3.090	-	-	-	-
Binaural spatial audio tokenizer								
DualSpec-VAE**	15.6	1.608	0.638	2.275	0.686	0.762	0.833	0.785
BANC**	160	2.822	0.864	2.816	0.742	0.813	0.824	0.820
SpiderTok (Ours)	50	3.280	0.970	3.682	**0.913**	**0.994**	**0.975**	**0.893**

to filter out utterances with unclear pronunciation, and perform quality inspection on the generated binaural speech through manual evaluation at a preset ratio. More implementation details on the data construction of SpiderData are given in Appendix B.

5 Experiments

5.1 Experimental Setup

Evaluation Metrics. Since there is currently no open-source benchmark for spatial binaural speech understanding, generation and editing, we render each task's spatial evaluation data based on the evaluation framework of monaural models. To assess the model's capability, we render each monaural sample three times, under static, moving and orbiting conditions. For speech evaluation under spatial motion conditions (i.e., moving or orbiting), we uniformly average accuracy over the start position, end position, and motion state. During evaluation, only the spatial metrics are computed on the binaural speech and the rest use the monaural evaluation system after downmixing. In terms of the rationality of the downmixing operation, we provide quantitative results in Table 12, complemented by human evaluations (listening checks by the authors), showing that downmixing removes the spatial information, preserves the semantic content, and degrades timbre only slightly. These protocols measure the spatial capabilities of our models while verifying that basic capabilities such as ASR are retained.

Specifically, **1) Reconstruction:** The spatial reconstruction capacity of SpiderTok is scored on speech quality metrics PESQ, STOI and UTMOS, and spatial fidelity metrics ANG Cos, DIS Cos, IPD Corr and ILD Corr, following CSAVocoder (Zhu et al., 2026). The evaluation data is mainly rendered from the LibriTTS test-clean split (Zen et al., 2019). **2) Understanding:** Spatial understanding capability mainly revolves around three major aspects: spatial grounding, motion state recognition with classification accuracy, and ASR WER. The evaluation data is mainly rendered from the Librispeech test-clean split (Panayotov et al., 2015). **3) Generation:** For zero-shot spatial TTS, we report WER and TimberSIM, computed with Whisper-large-v3 and a WavLM-based model[7], together with ANG Cos and DIS Cos for the spatial state of the generated speech. For instruct spatial TTS, we report the same spatial metrics and add description-speech consistency (DSD), which checks whether the generated speech matches the instruction. The two evaluation sets are rendered from the EN splits of Seed-TTS (Anastassiou et al., 2024) and InstructTTSEval (Huang et al., 2025), respectively. **4) Editing:** For regular speech editing capability, we mainly consider the four categories in SpeechEditBench (Zhang et al., 2026): content, emotion, prosody, and paralinguistic, and adopt the stricter metric "Joint Success Rate" with the corresponding thresholds. Moreover, for spatial state editing and binaural speech rendering, we uniformly consider the editing task successful when ANG Cos is greater than 0.85 and DIS Cos is greater than 0.95. More details on the evaluation metrics can be found in Appendix C.3 and D.3.

Baselines. Since open-source work on spatial speech is scarce, we use two groups of baselines: state-of-the-art monaural models from academia and industry for the basic capabilities, and the few

[7] https://huggingface.co/bezzam/wavlm_large_finetune_seed_tts_eval

Table 2: Spatial understanding and ASR performance comparison. "-/-/-" denotes static, moving and orbiting states, respectively. "#" indicates that the model supports only static sources. "**" denotes the model retrained on our rendered binaural speech data for a fairer comparison.

Model	Spatial Understanding				ASR↓
	Direction↑	Updown↑	Distance↑	Motion↑	
Non-spatial Unified Understanding and Generation LALMs					
Audex*	-	-	-	-	3.60 / 3.17 / 3.46
Uniaudio 2.0*	-	-	-	-	3.79 / 3.78 / 4.01
Mimo-Audio*	-	-	-	-	2.61 / 2.19 / 2.72
Ming-UniAudio*	-	-	-	-	1.93 / 1.93 / 1.88
FireRedAudio*	-	-	-	-	**1.23 / 0.96 / 1.26**
Qwen2.5-Omni*	-	-	-	-	2.04 / 2.13 / 2.05
Binaural Spatial Understanding LALMs					
Random Guessing	12.5 / 12.5 / 12.5	33.3 / 33.3 / 33.3	33.3 / 33.3 / 33.3	33.3 / 33.3 / 33.3	-
OWL#	28.8 / - / -	44.0 / - / -	55.0 / - / -	-	-
BAT w/ SpatialAST#	28.5 / - / -	44.5 / - / -	53.7 / - / -	-	-
BAT w/ DSpAST#	46.5 / - / -	38.7 / - / -	57.3 / - / -	-	-
BAT w/ DSpAST + SFT**	84.7 / 58.5 / 73.7	83.0 / 54.5 / 84.3	64.0 / 58.3 / 63.7	91.8 / 89.5 / 90.3	-
SpiderAudio (Ours)	**96.5 / 70.5 / 84.0**	**92.5 / 65.0 / 93.5**	**67.5 / 60.0 / 67.0**	**100.0 / 97.8 / 98.5**	3.11 / 3.18 / 3.16

Table 3: Zero-shot and instruct spatial TTS performance comparison. "***" denotes the model supports only fixed-voice TTS without reference-voice cloning and we report only its WER.

Model	Zero-shot Spatial TTS				Instruct Spatial TTS		
	WER↓	TimberSIM↑	ANG Cos↑	DIS Cos↑	DSD↑	ANG Cos↑	DIS Cos↑
Non-spatial Unified Understanding and Generation LALMs							
Audex***	1.63 / 1.49 / 1.60	-	-	-	-	-	-
Uniaudio 2.0*	5.37 / 5.38 / 5.52	0.319 / 0.313 / 0.305	-	-	84.2	-	-
Mimo-Audio*	5.19 / 5.27 / 5.83	0.432 / 0.418 / 0.414	-	-	91.6	-	-
Ming-UniAudio*	2.21 / 2.23 / 2.38	0.546 / 0.537 / 0.538	-	-	-	-	-
FireRedAudio*	**1.71 / 1.71 / 1.82**	**0.637 / 0.621 / 0.620**	-	-	**97.6**	-	-
Qwen2.5-Omni*	2.53 / 2.56 / 2.61	0.616 / 0.606 / 0.609	-	-	-	-	-
Binaural Spatial Speech Generation LALMs							
SpiderAudio (Ours)	3.47 / 3.57 / 3.52	0.468 / 0.451 / 0.460	**0.790 / 0.774 / 0.789**	**0.974 / 0.973 / 0.974**	93.3	**0.845**	**0.977**

available binaural models for the spatial ones. Baselines used in our work are listed in Appendix E, as well as their categorization and detailed model descriptions.

Table 4: Spatial speech editing performance comparison.

Model	Regular				Spatial	
	Content↑	Emotion↑	Prosody↑	Paralinguistic↑	Spatial State↑	Rendering↑
Non-spatial Speech Editing LALMs						
Unison*	28.7 / 26.7 / 29.0	4.8 / 3.9 / 3.7	42.9 / 43.6 / 44.6	**76.0 / 70.5 / 74.0**	-	-
AuK*	**65.0 / 67.3 / 65.3**	8.3 / **8.8 / 8.8**	**65.4 / 64.9 / 62.8**	30.0 / 31.5 / 32.0	-	-
Ming-UniAudio*	42.0 / 47.3 / 47.0	5.5 / 5.5 / 6.0	58.1 / 56.1 / 52.9	29.0 / 31.0 / 34.5	-	-
Step-Audio-EditX*	15.7 / 18.0 / 13.3	4.7 / 4.9 / 5.5	42.7 / 45.8 / 46.9	58.5 / 64.5 / 59.0	-	-
Binaural Spatial Speech Editing LALMs						
SpiderAudio (Ours)	49.0 / 46.7 / 47.7	**9.1** / 8.7 / 8.2	60.3 / 61.0 / 59.7	52.5 / 50.0 / 53.0	**83.3 / 80.7 / 82.7**	**85.7 / 83.0 / 83.7**

5.2 Experimental Results

Binaural Reconstruction. Table 1 compares SpiderTok with monaural and binaural tokenizers. **1)** Against the two binaural systems, SpiderTok improves every spatial metric over the better of the two, with the largest relative gain on angular cosine similarity at 23%, and improves speech quality as well. Notably, these results are obtained at a lower token rate than BANC. **2)** SpiderTok trails only the two strongest monaural tokenizers, which are trained at industrial scale , and stays ahead of the rest on PESQ. A monaural tokenizer reconstructs one channel.Ours reconstructs two and their interaural relation, which may cost some speech quality.

Spatial Speech Understanding. Table 2 evaluates spatial attribute accuracy and ASR performance. SpiderAudio leads the existing spatial models by a large margin on every attribute under static, moving, and orbiting sources, and this advantage extends to distance, the hardest of the four attributes. The largest relative gain is on azimuth for moving sources, where accuracy rises from 58.5 to 70.5. On ASR performance, SpiderAudio outperforms UniAudio 2.0 and Audex, while the four stronger monaural systems remain ahead. Two factors account for this. Those systems are industrial models trained on far larger monaural corpora, and their representations carry speech alone, whereas ours must also preserve the interaural structure reported in the spatial columns.

Table 5: Ablation study on spatial reconstruction loss $\mathcal{L}_{\text{sr}}$ from SpiderTok training.

Setting	Speech Quality			Spatial Fidelity			
	PESQ↑	STOI↑	UTMOS↑	ANG Cos↑	DIS Cos↑	IPD Corr↑	ILD Corr↑
$\mathcal{L}_{\text{sr}}$	**3.280**	**0.970**	**3.682**	**0.913**	**0.994**	**0.975**	**0.893**
w/o $\mathcal{L}_{\text{mag}}$	2.163	0.882	3.046	0.862	0.936	0.931	0.842
w/o $\mathcal{L}_{\text{sc}}$	2.658	0.936	3.287	0.885	0.957	0.968	0.872
w/o $\mathcal{L}_{\text{ph}}$	2.423	0.904	3.121	0.873	0.942	0.955	0.864
w/o $\mathcal{L}_{\text{if}}$	3.006	0.962	3.576	0.894	0.976	0.963	0.891
w/o $\mathcal{L}_{\text{ipd}}$	2.971	0.950	3.442	0.725	0.873	0.792	0.730
w/o $\mathcal{L}_{\text{e}}$	2.994	0.947	3.486	0.762	0.896	0.822	0.771

Table 6: Ablation study on Stage 2 (semantic distillation) of the SpiderTok's training.

Setting	Speech Quality			Spatial Fidelity			
	PESQ↑	STOI↑	UTMOS↑	ANG Cos↑	DIS Cos↑	IPD Corr↑	ILD Corr↑
Three Stages	**3.280**	**0.970**	**3.682**	**0.913**	**0.994**	**0.975**	**0.893**
w/o Stage 2	3.034	0.922	3.275	0.907	**0.994**	0.972	0.891

Table 7: Ablation study on editing strategy used in the binaural editing task.

Setting	Regular				Spatial	
	Content↑	Emotion↑	Prosody↑	Paralinguistic↑	Spatial State↑	Rendering↑
Direct-generation	46.3 / 41.7 / 45.0	9.0 / 8.7 / 8.1	58.7 / 59.3 / 59.0	51.0 / 48.5 / 52.0	80.3 / 77.7 / 80.0	82.7 / 80.7 / 81.3
Thinking-before-generation	**49.0 / 46.7 / 47.7**	**9.1 / 8.7 / 8.2**	**60.3 / 61.0 / 59.7**	**52.5 / 50.0 / 53.0**	**83.3 / 80.7 / 82.7**	**85.7 / 83.0 / 83.7**

Spatial Speech Generation. Table 3 reports two spatial generation settings. In zero-shot spatial TTS, the model imitates the timbre and the spatial state of a binaural reference while speaking new content. In instruct spatial TTS, the target position and motion are specified in the instruction rather than taken from a reference. To our knowledge, no prior system reports angular and distance spatial metrics in either setting. On ASR WER and TimberSIM for regular monaural evaluation metrics, SpiderAudio remains competitive, similar to the understanding evaluations in Table 2.

Spatial Speech Editing. Table 4 assesses two spatial editing evaluations and in both the target position is given in the instruction. **1)** Spatial state editing requires the model to move an binaural speech to that position while preserving its content and speaker. **2)** Binaural rendering instead supplies a monaural input duplicated across both channels. To the best of our knowledge, this is the first results reported for spatial speech editing On the four regular monural evaluation attributes, SpiderAudio ranks second overall except for AuK. However, it is comparable to AuK on emotion while substantially stronger on paralinguistic editing. AuK retains an advantage on content and prosody.

5.3 Ablation Studies

Reconstruction objective. Table 5 removes each term of $\mathcal{L}_{\text{sr}}$ in turn. Removing any single term degrades reconstruction relative to the full objective, so the six components are complementary rather than redundant. This ablation suggests that further improvement is more likely to come from additional objectives designed for binaural structure.

Semantic distillation. Table 6 compares the full three-stage schedule with one that skips Stage 2. Spatial fidelity is almost unchanged, as the binaural spatial information is encoded by Stage 1 and its modules are frozen thereafter. However, speech quality degrades markedly on all three metrics.

"Thinking-Before Generating" strategy. Table 7 compares the two editing strategies. Thinking-Before-Generation improves every metric over direct generation, and the two spatial metrics gain the most. The explicit reasoning step is therefore necessary for better spatial preservation.

6 Conclusion

We present SpiderTok, an all-round speech tokenizer that encodes 16 kHz binaural speech into a compact 50 Hz unified representation carrying semantic, acoustic and spatial information, and SpiderAudio, a first all-round large speech language model that use the unified representation to jointly perform speech understanding, generation, and editing. Experiments show that SpiderTok outperforms binaural baselines, and SpiderAudio surpasses baselines on understanding under static, moving and orbiting conditions while reporting the first spatial TTS and editing results in this settings. To address the limitations discussed in Appendix F, future work will focus on HRTF personalization and extending the framework to multi-speaker scenarios.

AI USE STATEMENT

AI Use Statement. We used AI tools to assist with language polishing, and literature review. For data construction and evaluation, we used Qwen3-TTS for synthetic speech generation, Qwen3-ASR for data filtering, and Gemini-3-Flash for speech-attribute annotation and LLM-based evaluation, as detailed in the paper. The authors take full responsibility for the data construction and evaluation procedures and all reported results.

REPRODUCIBILITY STATEMENT

We support more implementation details on the training data, model architecture, loss function, training environment, and evaluation framework needed to reproduce this work in the part of Appendix. In addition, after the blind review period, we promise to open-source all SpiderTok and SpiderAudio models and their weights.

REFERENCES

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.

Subrata Biswas, Mohammad Khan, and Bashima Islam. Owl: Geometry-aware spatial reasoning for audio large language models. In *International Conference on Learning Representations*, volume 2026, pp. 20685–20710, 2026.

Jens Blauert. *Spatial hearing: the psychophysics of human sound localization*. The MIT press, 1996.

Maksim Borisov, Egor Spirin, and Daria Diatlova. Nonverbaltts: A public english corpus of text-aligned nonverbal vocalizations with emotion annotations for text-to-speech. *arXiv preprint arXiv:2507.13155*, 2025.

AI Boson. Higgs audio v2: Redefining expressiveness in audio generation, 2025.

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.

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.

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

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*, 2024.

William G Gardner and Keith D Martin. Hrtf measurements of a kemar. *The Journal of the Acoustical Society of America*, 97(6):3907–3908, 1995.

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.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026.

Jinbo Hu, Yin Cao, Ming Wu, Zhenbo Luo, and Jun Yang. Salm: Spatial audio language model with structured embeddings for understanding and editing. *arXiv preprint arXiv:2507.16724*, 2025.

Kexin Huang, Qian Tu, Liwei Fan, Chenchen Yang, Dong Zhang, Shimin Li, Zhaoye Fei, Qinyuan Cheng, and Xipeng Qiu. Instructttseval: Benchmarking complex natural-language instruction following in text-to-speech systems. *arXiv preprint arXiv:2506.16381*, 2025.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Ruiqi Li, Ziang Zhang, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93809–93826, 2025.

Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. Libritts-r: A restored multi-speaker text-to-speech corpus. *arXiv preprint arXiv:2305.18802*, 2023.

Zhifeng Kong, Sang-gil Lee, Jaehyeon Kim, Boxin Wang, Zihan Liu, Sungwon Kim, Yang Chen, Arushi Goel, Rajarshi Roy, Wenliang Dai, et al. Unified audio intelligence without regressing on text intelligence. *arXiv preprint arXiv:2607.05196*, 2026.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in neural information processing systems*, 36:27980–27993, 2023.

Ke Lei, Yu Zhang, Changhao Pan, Xueyi Pu, Wenxiang Guo, Ruiqi Li, and Zhou Zhao. Towards streaming synchronized spatial audio generation via autoregressive diffusion transformer. *arXiv preprint arXiv:2605.30940*, 2026.

Bohan Li, Shi Lian, Hankun Wang, Yiwei Guo, Yu Xi, Zhihan Li, Da Zheng, Colin Zhang, and Kai Yu. Holitok: A coutinuous holistic tokenization with robust dual capabilities of speech generation and understanding. *arXiv preprint arXiv:2605.29948*, 2026a.

Junjie Li, Xuelong Geng, Kun Xie, Feiyu Shen, Yichen Wu, Ziqi Dai, Yichen Han, Yan Jia, Kai Huang, Junjie Chen, et al. Fireredaudio: A general-purpose audio language model with decoupled continuous representations for understanding and generation. *arXiv preprint arXiv:2608.24168*, 2026b.

Zhaoqing Li, Haoning Xu, Jingran Su, Yaofang Liu, Zhefan Rao, Huimeng Wang, Jiajun Deng, Tianzi Wang, Zengrui Jin, Rui Liu, et al. Unison: A unified sound generation and editing framework via deep llm fusion. *arXiv preprint arXiv:2605.31530*, 2026c.

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.

Ziyang Ma, Zhikang Niu, Wenming Tu, Tianrui Wang, Ruiqi Yan, Junxi Liu, Yanru Huo, Nickk Huang, Yang Liu, Qicong Xie, et al. Auk technical report: An open-source foundational model for speech generation and editing. *arXiv preprint arXiv:2609.08936*, 2026.

Viola Negroni, Davide Salvi, Luca Comanducci, Taiba Majid Wani, Madleen Uecker, Irene Amerini, Stefano Tubaro, and Paolo Bestagini. Lj-tts: A paired real and synthetic speech dataset for single-speaker tts analysis. *Electronics*, 15(1):169, 2025.

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.

Anton Ratnarajah, Shi-Xiong Zhang, and Dong Yu. Banc: Towards efficient binaural audio neural codec for overlapping speech. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

Parthasaarathy Sudarsanam, Sebastian Braun, and Hannes Gamper. Foa tokenizer: Low-bitrate neural codec for first order ambisonics with spatial consistency loss. *arXiv preprint arXiv:2510.22241*, 2025.

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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 993–1003, 2021.

Huimeng Wang, Hui Lu, Jiajun Deng, Haoning Xu, Youjun Chen, Xueyuan Chen, Zhaoqing Li, Shuhai Peng, Shiyin Kang, and Xunying Liu. Semavoice: Semantic-aware continuous autoregressive speech synthesis. *arXiv preprint arXiv:2605.16964*, 2026.

Kevin Wilkinghoff and Zheng-Hua Tan. Dspast: Disentangled representations for spatial audio reasoning with large language models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 14747–14751. IEEE, 2026.

Jin Xu, Zhifang Guo, Jinzheng He, et al. Qwen2.5-omni technical report, 2025. URL `https://arxiv.org/abs/2503.20215`.

Canxiang Yan, Chunxiang Jin, Dawei Huang, Haibing Yu, Han Peng, Hui Zhan, Jie Gao, Jing Peng, Jingdong Chen, Jun Zhou, et al. Ming-uniaudio: Speech llm for joint understanding, generation and editing with unified representation. *arXiv preprint arXiv:2511.05516*, 2025a.

Chao Yan, Boyong Wu, Peng Yang, Pengfei Tan, Guoqiang Hu, Li Xie, Yuxin Zhang, Fei Tian, Xuerui Yang, Xiangyu Zhang, et al. Step-audio-editx technical report. *arXiv preprint arXiv:2511.03601*, 2025b.

Dongchao Yang, Yuanyuan Wang, Dading Chong, Songxiang Liu, Xixin Wu, and Helen Meng. Uniaudio 2.0: A unified audio language model with text-aligned factorized audio tokenization. *arXiv preprint arXiv:2602.04683*, 2026.

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

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.

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

Hanlin Zhang, Daxin Tan, Dehua Tao, Xiao Chen, Haochen Tan, and Linqi Song. Speechedit-bench: A bilingual multi-attribute benchmark for instruction-guided speech editing. *arXiv preprint arXiv:2606.01804*, 2026.

Lei Zhao, Sizhou Chen, Linfeng Feng, Jichao Zhang, Xiao-Lei Zhang, Chi Zhang, and Xuelong Li. Dualspec: Text-to-spatial-audio generation via dual-spectrogram guided diffusion model. *IEEE Transactions on Multimedia*, 2026.

Zhisheng Zheng, Puyuan Peng, Ziyang Ma, Xie Chen, Eunsol Choi, and David Harwath. Bat: Learning to reason about spatial sounds with large language models. *arXiv preprint arXiv:2402.01591*, 2024.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 35139–35148, 2026.

Zhiyuan Zhu, Han Wang, Wenxiang Guo, Yu Zhang, Changhao Pan, Rui Yang, and Zhou Zhao. Csavocoder: A causal spatial audio vocoder towards real-time spatial audio generation. *arXiv preprint arXiv:2608.25404*, 2026.

Franz Zotter and Matthias Frank. *Ambisonics: A practical 3D audio theory for recording, studio production, sound reinforcement, and virtual reality*. Springer Nature, 2019.

A Discussion on FOA and Binaural Speech for VAE Modeling

A.1 Distinction

The fundamental difference between FOA (First-Order Ambisonics) and Binaural Speech lies in their representation domain: (1) **FOA**: A scene-centered representation that uses four channels (W, X, Y, Z) to describe the directional distribution of a three-dimensional sound field. It is independent of the listener's physiology and can be rotated mathematically. (2) **Binaural Speech**: A listener-centered representation consisting of two-channel signals. It contains spatial cues such as ITD and ILD introduced by the HRTF, and strongly depends on individual physiological characteristics.

Table 8: Key differences between FOA and Binaural Speech

Dimension	**FOA**	**Binaural Speech**
Representation domain	Scene-centered	Listener-centered
Number of channels	4 (W, X, Y, Z)	2 (left-channel waveform, right-channel waveform)
Core information	Directional distribution of sound	Interaural time and level differences
Rotation property	Lossless rotation, suitable for head tracking	Fixed to head pose, requires re-rendering
Individualization	Low, universal sound field description	High, significant HRTF individual differences
Perceptual limitation	Limited spatial resolution, blurred localization	Generic HRTF easily causes in-head localization
Data nature	Mathematical encoding of the sound field	Sound pressure signals at the eardrums

A.2 Why Binaural Speech Research Is Necessary

Although FOA can describe a sound field, the ultimate interface of human hearing is binaural. Therefore, binaural speech research is irreplaceable: (1) **Hearing is inherently binaural**: Spatial hearing and the cocktail party effect rely on differences between binaural signals; binaural listening can yield significant improvement in speech reception thresholds (spatial release from masking). (2) **Conversion from FOA to binaural is not lossless**: Virtual loudspeaker rendering degrades spatial resolution, and generic HRTFs reduce externalization and blur localization. (3) **Understanding, editing, and generation form a closed research loop**: **Understanding** investigates how the brain extracts binaural cues, and how hearing loss, hearing aids, and noise disrupt these cues, providing a foundation for hearing rehabilitation. **Editing** operates on existing binaural speech to achieve controllable modifications, such as altering source direction, distance, externalization, reverberation, or the spatial configuration between target and interfering sources, while preserving perceptual naturalness and acoustic consistency. **Generation** synthesizes binaural speech with strong externalization and spatial impression for VR, teleconferencing, and assistive listening devices, requiring solutions to challenges such as HRTF individualization, head tracking, and dynamic rendering. Together, these three tasks form a closed loop: understanding provides the perceptual and acoustic priors, editing enables flexible and controllable manipulation, and generation produces novel binaural content for real-world applications.

B Data Construction

In this section, we will provide more details on the rendering pipeline to generate spatial binaural speech from monaural speech, and the construction pipeline of SpiderData, a full-task training dataset for binaural speech understanding, generation, and editing tasks used in our work.

B.1 Binaural Speech Rendering via HRTF

B.1.1 Notation

Let $x[n]$, $n = 0, \dots, N-1$, denote a mono speech signal sampled at $f_s = 16$ kHz. The listener is fixed at the origin of a head-local frame with $+x$ pointing front, $+y$ left, and $+z$ up; the source position $\mathbf{s}[n] \in \mathbb{R}^3$ is expressed in that frame. A direction is parameterized by azimuth α, elevation ϵ, and distance r, which map to Cartesian coordinates as

$$\mathbf{p}(\alpha, \epsilon, r) = r \begin{bmatrix} \cos\epsilon \cos\alpha \\ \cos\epsilon \sin\alpha \\ \sin\epsilon \end{bmatrix}, \qquad \begin{aligned} r &= \|\mathbf{s}\|, \\ \epsilon &= \arcsin\big(s_z / r\big), \\ \alpha &= \operatorname{atan2}(s_y, s_x) \bmod 360^\circ. \end{aligned} \tag{2}$$

For a vector-valued filter $\mathbf{h} \in \mathbb{R}^{L \times 2}$ whose columns are the left and right ear responses, we write $(\mathbf{x} * \mathbf{h})[n] = \sum_l x[n-l]\,\mathbf{h}[l]$ for the two-channel convolution.

B.1.2 HRIR Acquisition and Directional Interpolation

Rendering uses measured KEMAR head-related impulse responses (HRIRs) stored in SOFA format (Gardner & Martin, 1995), comprising $D = 710$ measurement directions (α_j, ϵ_j) with $\alpha_j \in [0^\circ, 355^\circ]$ and $\epsilon_j \in [-40^\circ, 90^\circ]$. Each measured response is trimmed to its first 192 taps at the native 48 kHz rate to discard the measurement-noise tail, windowed with a 32-point Hann taper to suppress the high-frequency leakage of a rectangular truncation, and de-meaned to remove any DC offset. The responses are then resampled to f_s, yielding $\mathbf{h}_j \in \mathbb{R}^{L \times 2}$ with $L = 64$ taps. Resampling is essential: applying a response whose interaural time difference (ITD) and spectral cues are encoded at the wrong time base would displace both cues.

Because the measurement grid is discrete, an arbitrary direction is resolved by spherical barycentric interpolation. Unit vectors $\mathbf{u}_j$ are constructed from (α_j, ϵ_j) and their convex hull is computed, which coincides with the Delaunay triangulation of S^2; the resulting 1416 faces form the interpolation mesh. For a query direction, the triangle intersected by the ray $\{t\,\mathbf{u}(\alpha, \epsilon) : t > 0\}$ is located by ray–plane intersection, and the HRIR follows as

$$\mathbf{h}(\alpha, \epsilon) = \sum_{k=1}^{3} \lambda_k\, \mathbf{h}_{j_k}, \qquad \sum_{k=1}^{3} \lambda_k = 1, \;\; \lambda_k \geq 0, \tag{3}$$

where (j_1, j_2, j_3) index the triangle's vertices and λ_k are the barycentric coordinates of the intersection point. Linear interpolation is admissible here because all HRIRs share a common time origin, so the ITD encoded in the taps is preserved by the weighted sum.

B.1.3 Distance Gain

Distance is modeled by an inverse-distance law normalized to unity at the reference distance, clamped from below to avoid a singularity:

$$g(r) = \frac{r_{\text{ref}}}{\max(r,\, r_{\min})}, \qquad r_{\text{ref}} = 1\,\text{m}, \;\; r_{\min} = 0.1\,\text{m}. \tag{4}$$

B.1.4 Static Sources

For a stationary source the position is drawn once as $\alpha \sim \mathcal{U}(0^\circ, 360^\circ)$, $\epsilon \sim \mathcal{U}(-30^\circ, 45^\circ)$, and $r \sim \mathcal{U}(0.5, 5.0)$ m. The binaural output is a single convolution per ear,

$$\mathbf{y}[n] = g(r)\,(x * \mathbf{h}(\alpha, \epsilon))[n], \tag{5}$$

of length $N + L - 1$.

B.1.5 Moving and Orbiting Sources

Two dynamic conditions are defined, each parameterized by a per-sample trajectory $\mathbf{s}[n]$ with $T = N$ samples.

Linear motion (moving). Two key frames are drawn independently,

$$\begin{aligned} \alpha_0 &\sim \mathcal{U}(0^\circ, 360^\circ), \qquad \Delta\alpha \sim \pm\mathcal{U}(40^\circ, 200^\circ), \quad \alpha_1 = \alpha_0 + \Delta\alpha, \\ \epsilon_0, \epsilon_1 &\sim \mathcal{U}(-30^\circ, 45^\circ), \quad r_0, r_1 \sim \mathcal{U}(0.5, 5.0)\,\mathrm{m}, \end{aligned} \tag{6}$$

converted to $\mathbf{p}_0, \mathbf{p}_1$ via Equation 2. The position interpolates linearly in Cartesian space between the two endpoints,

$$\mathbf{s}[n] = (1 - \tau_n)\,\mathbf{p}_0 + \tau_n\,\mathbf{p}_1, \qquad \tau_n = \frac{n}{N-1}, \tag{7}$$

i.e. the source travels along the chord joining the two keyframes.

Circular motion (orbiting). The source traverses a horizontal circle about the listener,

$$\mathbf{s}[n] = \begin{bmatrix} R\cos\theta_n \\ R\sin\theta_n \\ z \end{bmatrix}, \qquad \theta_n = \sigma\,\frac{2\pi n}{f_s P} + \varphi_0, \tag{8}$$

with $R \sim \mathcal{U}(0.6, 2.5)$ m, $z \sim \mathcal{U}(-0.4, 0.6)$ m, period $P \sim \mathcal{U}(3, 8)$ s, direction $\sigma \in \{-1, +1\}$ chosen uniformly, and initial phase $\varphi_0 \sim \mathcal{U}(0, 2\pi)$.

B.1.6 Block-Wise Rendering with Cross-Faded HRIRs

Updating the HRIR at every sample would be prohibitively expensive, whereas holding it fixed within a block and switching abruptly at block boundaries injects periodic discontinuities that are audible as zipper noise. We therefore partition the signal into blocks of $B = 512$ samples (32 ms at f_s), render each block with a direction sampled at its *start*, and cross-fade consecutive blocks.

Let block i start at $n_i = iB$ and have length $m_i = \min(B, N - n_i)$. The direction is sampled at the block onset,

$$\mathbf{q}_i = \mathbf{s}[n_i], \qquad \mathbf{h}_i = \mathbf{h}\big(\alpha(\mathbf{q}_i), \epsilon(\mathbf{q}_i)\big), \qquad g_i = g\big(\|\mathbf{q}_i\|\big), \tag{9}$$

and the segment $x_i = x[n_i : n_i + m_i]$ is convolved both with the current and the previous block's HRIR,

$$\mathbf{c}_i^{\mathrm{cur}} = x_i * \mathbf{h}_i, \qquad \mathbf{c}_i^{\mathrm{prev}} = x_i * \mathbf{h}_{i-1}, \tag{10}$$

each of length $m_i + L - 1$. The two are blended by a linear ramp that reaches unity at the segment end and *remains* at unity throughout the $L - 1$ samples of convolutional tail,

$$a_i[k] = \begin{cases} \dfrac{k}{m_i - 1}, & 0 \le k \le m_i - 1, \\ 1, & m_i \le k \le m_i + L - 2, \end{cases} \qquad b_i[k] = 1 - a_i[k]. \tag{11}$$

Holding $a_i = 1$ over the tail is what makes the scheme seamless: the decay of the new HRIR propagates unattenuated into the next block, while the next block's "previous" HRIR is by construction this block's "current" HRIR. So the effective response is continuous across every boundary. The distance gain is interpolated by the same ramp, keeping the amplitude trajectory synchronized with the HRIR switch. Contributions are accumulated by overlap-add,

$$\mathbf{y}[n_i + k] \mathrel{+}= g_{i-1}\,b_i[k]\,\mathbf{c}_i^{\mathrm{prev}}[k] \;+\; g_i\,a_i[k]\,\mathbf{c}_i^{\mathrm{cur}}[k], \tag{12}$$

with the $\mathbf{c}_i^{\mathrm{prev}}$ term omitted for $i = 0$. This is the same block-cross-fading construction used by Steam Audio, Resonance Audio, and the IEM Plug-in Suite, and it underlies the binaural simulators used to build training data in prior binaural-generation work.

B.1.7 Output Binaural Speech

The convolutional tail is discarded to restore the input length, $\tilde{\mathbf{y}} = \mathbf{y}[0 : N]$, so that rendered clips remain sample-aligned with their transcripts. Loudness is protected on one side only—a clip is attenuated exactly when it would clip, leaving the relative level differences between distances and motion types intact:

$$\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}} \cdot \min\left(1,\ \frac{0.99}{\|\tilde{\mathbf{y}}\|_\infty}\right). \tag{13}$$

B.2 Data Preparation

Following previous work on spatial understanding and generation, our binaural speech training data is likewise mainly rendered from monaural speech in the open-source monaural English datasets, including Emilia (He et al., 2024), GigaSpeech (Chen et al., 2021), LJSpeech (Negroni et al., 2025), HiFiTTS [8], LibriTTS-R (Koizumi et al., 2023), VoxPopuli (Wang et al., 2021), NonverbalTTS (Borisov et al., 2025), RAVDESS (Livingstone & Russo, 2018), and IEMOCAP (Busso et al., 2008). The total duration is over 60k hours.

B.3 Data Construction for SpiderData

We render the binaural speech based on the above monaural speech via HRTF. Finally, we construct the training data needed for the binaural speech understanding, generation, and editing tasks in turn, forming the final SpiderData. The definitions and examples of the main sub-tasks considered in our work are listed in Table 9.

Table 9: Overview of specific tasks for joint binaural speech understanding, generation, and editing

Task	Definition	Example
	Binaural Speech Understanding	
Azimuth estimation	Predict the azimuth of the given reference binaural speech. The front direction is defined as $0°$, with the angle increasing counterclockwise up to $360°$.	**Input:** Reference binaural speech **Query:** In the reference speech, what is the horizontal azimuth of the speaker relative to the listener's front direction, in degrees? **Output:** $22.5°$
Elevation estimation	Predict the elevation of the given reference binaural speech. The horizontal plane at the listener's head position is defined as $0°$, with downward angles negative and upward angles positive.	**Input:** Reference binaural speech **Query:** In the reference speech, what is the vertical elevation angle of the speaker relative to the listener's head position, in degrees? **Output:** $15°$
Distance estimation	Predict the distance of the given reference binaural speech.	**Input:** Reference binaural speech **Query:** In the reference speech, what is the distance of the speaker relative to the listener, in meters? **Output:** 2.3 m
Motion state estimation	Predict the motion state of the given reference binaural speech.	**Input:** Reference binaural speech **Query:** In the reference speech, what is the motion state of the speaker, static or moving or orbiting? **Output:** Moving
ASR	Recognize the speech content of the given reference binaural speech.	**Input:** Reference binaural speech **Query:** In the reference speech, what does the speaker say? **Output:** "Have a good day!"

Continued on next page

[8] https://huggingface.co/datasets/MikhailT/hifi-tts

Task	Definition	Example
Speech captioning	Describe the reference binaural speech from the perspectives of azimuth, elevation, distance, motion state, and speech content.	**Input:** Reference binaural speech **Query:** In the reference speech, what are the speaker's spatial state and spoken content? **Output:** The speaker says "Have a good day", with moving from a position with an azimuth of $22.5°$, an elevation of $15°$, and a distance of 2.3 m to a position with an azimuth of $21.8°$, an elevation of $15°$, and a distance of 2.2 m.
	Binaural Speech Generation	
Zero-shot spatial TTS	Based on the timbre and spatial state of the reference binaural speech, generate a new binaural speech with the given speech content.	**Input:** Reference binaural speech **Instruction:** Imitate the timbre and spatial state of the reference binaural speech and say "Nice to meet you!". **Output:** A new binaural speech with the same timbre and spatial state as the reference binaural speech.
Instruct spatial TTS	Generate a new binaural speech based on the given instruction on prosody or spatial state.	**Input:** None **Instruction:** At a position 0.5 m directly in front, say "Nice to meet you!" in a slow speaking pace. **Output:** A new binaural speech following the instruction.
	Binaural Speech Editing	
Spatial state editing	Change the spatial position or motion state of the given reference binaural speech.	**Input:** Reference binaural speech **Instruction:** Move the speaker's position to about 1 m directly to the left. **Output:** A new binaural speech following the instruction.
Binaural speech rendering via HRTF	Render the monaural speech into binaural audio according to the specified spatial state description.	**Input:** Reference binaural speech **Instruction:** Render the given monaural speech into spatial binaural speech according to a trajectory that moves in a straight line from 2 m directly in front to 2 m to the front right. **Output:** A new binaural speech following the instruction.
Content editing	Insert, remove or replace words in the given binaural speech.	**Input:** Reference binaural speech **Instruction:** Remove the word "always" in the given binaural speech. **Output:** A new binaural speech following the instruction.
Emotion editing	Change the emotional state of the given binaural speech.	**Input:** Reference binaural speech **Instruction:** Say the given binaural speech again in an angry tone. **Output:** A new binaural speech following the instruction.
Prosody editing	Modify the pitch, speaking rate or volume of the given binaural speech.	**Input:** Reference binaural speech **Instruction:** Speed up the given binaural speech by a factor of two. **Output:** A new binaural speech following the instruction.
Paralinguistic editing	Add or remove non-verbal vocal events like breathing, laughing or coughing.	**Input:** Reference binaural speech **Instruction:** Remove the coughing in the middle of the given binaural speech. **Output:** A new binaural speech following the instruction.

B.3.1 BINAURAL SPEECH UNDERSTANDING

Spatial understanding and ASR corpus. We randomly sample the target motion state, azimuth, elevation, and distance at first, and then render the original monaural speech into the corresponding binaural speech using the rendering method described in Section B.1. The target motion state, azimuth, elevation, and distance are regarded as the ground-truth labels. Next, based on the types of understanding questions in Table 9, we design the corresponding queries and paired labels, and use the Qwen3.5-27B model [9] to paraphrase the queries to improve the model's generalization ability.

B.3.2 BINAURAL SPEECH GENERATION

Zero-shot spatial TTS corpus. We first randomly sample the target motion state, azimuth, elevation, and distance. Then, for each reference monaural speech, we use Qwen3-TTS (Hu et al., 2026) to generate a monaural speech with the same timbre, and render these two paired monaural speech into binaural speech using the same sampling parameters as the training data for zero-shot spatial TTS.

Instruct spatial TTS corpus. We first use Gemini-3-Flash to annotate each monaural speech with a free-form natural-language caption and structured attributes covering speech attributes such as pitch, speaking rate, volume, emotion, and personality. Then, we randomly sample the target motion state, azimuth, elevation, and distance before rendering them into target binaural speech.

B.3.3 BINAURAL SPEECH EDITING

Speech editing has recently become a particularly popular topic. In addition to retaining content, emotion, prosody and paralinguistic editing from monaural speech editing tasks, we introduce two new spatial editing capabilities: spatial state editing and binaural speech rendering via HRTF.

Spatial state editing corpus. Spatial state editing requires the model to change the spatial position or motion state according to the instruction while preserving the timbre and prosody features. We first sample two different sets of spatial positions and motion states to form an editing instruction, and then render the same monaural speech with this pair of instructions to obtain the binaural reference and target speech.

Binaural speech rendering corpus. Different from spatial state editing, binaural speech rendering requires the model to directly render a monaural speech into spatial binaural speech given the spatial position and motion state. We randomly sample a set of spatial state information and use HRTF to render it into binaural speech as the target speech.

Content editing corpus. Content editing requires the model to insert, remove, or replace words while preserving speaker timber, spatial state and prosody. Starting from high-quality monaural speech, we delete, modify, or add one or more words or phrases in its transcription to obtain the target transcription after editing. We then use Qwen3-TTS to clone the timbre in the monaural speech, and employ Qwen3-ASR (Shi et al., 2026) as a quality-check model to ensure the generation quality. After pairing the monaural reference and target speech, we render the two speech signals into binaural speech using a set of randomly sampled spatial parameters, which serve as the binaural reference and target speech.

Emotion editing corpus. Emotion editing requires the model to be able to imitate the same speaker reading the same text with the prosodic conditions corresponding to different emotions. Following AuK (Ma et al., 2026), we mainly consider eight emotion categories: anger, happiness, sadness, fear, surprise, disgust, calmness, and excitement. We first use CosyVoice2 (Du et al., 2024) to generate a pair of speech with different emotional styles for the same text, and then use IndexTTS2 (Zhou et al., 2026) to generate a pair of monaural audios with the same timbre and content but different emotions, where one identical speech serves as the timbre reference and the just-generated pair serves as the emotional style reference. During this process, Qwen3-ASR continuously performs quality checking. In addition, to improve the model's robustness in real-world scenarios, we further

[9] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

incorporate the RAVDESS dataset, which was recorded by 24 professional actors (12 male and 12 female) and covers eight emotions: calm, happy, sad, angry, fearful, surprised, and disgusted, plus a neutral baseline state. The spoken text consists of two identical sentences, making it well suited as reference and target speech for emotion editing. After pairing the monaural reference and target speech, we render the two speech signals into binaural speech using a set of randomly sampled spatial parameters, which serve as the binaural reference and target speech.

Prosody editing corpus. Prosody editing requires the model to adjust prosody features such as pitch, volume, and speaking rate without significantly changing the timbre. These three metrics all have corresponding filters in FFmpeg, which can precisely control the listed multiples, dB values, and semitones. For each monaural audio, the randomly sampled speaking rate multiple is drawn from 0.5x, 0.75x, 1.25x, 1.5x, 2.0x, the volume offset is an arbitrary integer from -15 dB to 15 dB, and the pitch shift is an arbitrary integer from -5 to 5 semitones. After pairing the monaural reference and target speech, we render the two speech signals into binaural speech using a set of randomly sampled spatial parameters, which serve as the binaural reference and target speech.

Paralinguistic editing corpus. Paralinguistic editing requires the model to add or remove non-verbal vocal events, including breathing, laughing, coughing, and sighing. Starting from NonverbalTTS, we remove the non-verbal events in each speech to obtain pure linguistic content, and then use Qwen3-TTS to clone its timbre and generate speech with the pure linguistic content, forming the monaural reference and target speech. Conversely, if the editing instruction is to add non-verbal events, we simply swap the monaural reference and target speech. After pairing the monaural reference and target speech, we render the two speech signals into binaural speech using a set of randomly sampled spatial parameters, which serve as the binaural reference and target speech.

In addition, when constructing the training data for the "Thinking-Before-Generation" mode, the thinking content is divided into two parts. The first part is to caption the reference audio, which is similar to the speech captioning label in binaural speech understanding tasks. The second part is to clarify the edited target result in conjunction with the editing instruction. For example, in a content editing task, the original speech is "I do not think this matter is right," the editing instruction is "Replace the word 'right' with 'wrong'," and the clarified result is "I do not think this matter is wrong."

C SpiderTok

In this section, we provide additional implementation details about the proposed unified binaural audio tokenizer SpiderTok. Figure 2 provides an overview of the SpiderTok framework.

C.1 Model Architecture and Configuration

Table 10 provides the detailed architecture and configuration of the SpiderTok model, which has a total of 477.09M parameters. The model operates on input binaural speech tensors of size $(B, 2, N)$, whose sampling rate is 16 kHz.

The StereoEncoder (80.11M parameters) utilizes Frame and a 320-dimensional Waveform Embedding followed by two Linear layers and a stack of ten Causl Multi-head Self-Attention (CausalMHA blocks, 11 heads $\times 64$ dimensions). This yields an acoustic latent variable of shape $(B, 2T, 32)$.

The CausualSemantic Bottleneck (314.91M parameters), built upon the Whisper-large-v3 encoder, transforms the acoustic latent via Temporal Pooling and Linear layers to a dimension of 1280, aligned with Whisper-encoded semantic represeantations. It features a single CausalMHA layer with 20 heads before expanding and projecting the features through two Linear layers ($1280 \rightarrow 5120 \rightarrow 1280$), resulting in a unified feature representation of $(B, T, 1280)$, whose token-per-second (TPS) equals to 50 Hz.

Reconstruction is handled by the StereoDecoder (80.45M parameters), which first projects the unified features back to $(B, 2T, 704)$ and then processes them through ten CausalMHA layers. The final output is generated by two HiFiGAN-based VocoderHeads, which utilize upsampling rates of $(8, 5, 2, 2)$ and kernel sizes of $(16, 11, 4, 4)$ to produce the final 16 kHz binaural speech output.

Table 10: Detailed architecture and configuration of the SpiderTok model.

Module	Layer	Shape / Dimension
Input binaural speech		$(B, 2, N)$
StereoEncoder	Frame	$(B, 2, N) \rightarrow (B, 2T, 320)$
	Waveform Embedding	320
	Linear$\times 2$	$320 \rightarrow 704 \rightarrow 704$
	CausalMHA$\times 10$	704 (11 heads $\times 64$)
	Linear	$704 \rightarrow 64$
Acoustic latent		$(B, 2T, 32)$
CausalSemantic Bottleneck	Temporal Pooling	$(B, 2T, 32) \rightarrow (B, T, 64)$
	Linear	1280
	CausalMHA	1280 (20 heads $\times 64$)
	Linear$\times 2$	$1280 \rightarrow 5120 \rightarrow 1280$
Unified feature		$(B, T, 1280)$
StereoDecoder	Back Projector	$(B, T, 1280) \rightarrow (B, 2T, 704)$
	CausalMHA$\times 10$	704 (11 heads $\times 64$)
	Linear$\times 2$	$(B, 2T, 704) \rightarrow (B, T, 704) \times 2$
	VocoderHead (HiFi-GAN)$\times 2$	uprates = $(8, 5, 2, 2)$ kernels = $(16, 11, 4, 4)$
Output binaural speech		$(B, 2, N)$

C.2 Training Details of SpiderTok

As illustrated in Figure 2, the training of SpiderTok mainly includes three main stages: (1) Acoustic and spatial reconstruction training; (2) Semantic distillation training; (3) Joint reinforcement training of acoustic, semantic, and spatial attributes. Table 11 presents the implementation details for these stages including training object, training data, loss functions and learning rates. All training data originates from open-source high-quality monaural (single-channel) speech data, with being rendered into binaural spatial speech based on HRTF, encompassing three scenarios: static, linear moving, and orbiting. All experiments in this process are performed on 32 NVIDIA H800 GPUs. More detailed information about the three stages is introduced below:

C.2.1 Stage 1: Acoustic and Spatial Reconstruction Training

In the stage 1, we pay attention on training the StereoEncoder and StereoDecoder with incomplete structural symmetry on the task of binaural spatial speech reconstruction to retain acoustic and spatial attributes in acoustic latent. The loss function for this stage is calculated as follows:

$$\mathcal{L}_1 = \lambda_{\text{sr}} \cdot \mathcal{L}_{\text{sr}} + \lambda_{\text{adv}} \cdot \mathcal{L}_{\text{adv}} + \lambda_{\text{fm}} \cdot \mathcal{L}_{\text{fm}} + \lambda_{\text{kl}} \cdot \mathcal{L}_{\text{kl}} \tag{14}$$

where spatial reconstruction loss $\mathcal{L}_{\text{sr}}$ is a composite loss item, encompassing six distinct sub-items as supervision targets to ensure that the reconstructed binaural speech is close to the original binaural speech in terms of sound quality, clarity, and spatial perception. Adversarial loss $\mathcal{L}_{\text{adv}}$ and feature matching loss $\mathcal{L}_{\text{fm}}$ both adopt the HiFi-GAN style, with the goal of remedying the speech texture distortion caused by unnatural phase recovery. In addition, Kullback-Leibler divergence loss $\mathcal{L}_{\text{kl}}$ regularizes the latent space towards the prior Gaussian distribution.

Specifically, $\mathcal{L}_{\text{sr}}$ mainly includes log-magnitude loss $\mathcal{L}_{\text{mag}}$, spectral convergence loss $\mathcal{L}_{\text{sc}}$, complex spectral convergence loss $\mathcal{L}_{\text{ph}}$, instantaneous frequency loss $\mathcal{L}_{\text{if}}$, interaural phase difference loss $\mathcal{L}_{\text{ipd}}$, and stereo energy loss $\mathcal{L}_{\text{e}}$.

First, $\mathcal{L}_{\text{mag}}$ is a loss computed in the log-magnitude spectral domain. It first performs STFT on each of the two monaural speech signals and takes their magnitude spectra, then applies a logarithm to compress the dynamic range and better capture overall envelope shifts. After that, it takes the absolute error over the two signals, all time frames, and frequency bins, and averages uniformly, meaning every time-frequency bin is weighted equally. The formula can be written as:

Table 11: Implementation details during different SpiderTok training stages.

Stage	Training Object	Training Data	Loss	Learning Rate
Stage 1	StereoEncoder, StereoDecoder	Binaural spatial speeches with a duration of less than 20 seconds derived from all English samples in Emilia, GigaSpeech, LJSpeech, and Hi-FiTTS	$\mathcal{L}_{\text{sr}}, \mathcal{L}_{\text{adv}}, \mathcal{L}_{\text{fm}}, \mathcal{L}_{\text{kl}}$	Learning rate $= 3e^{-5}$; Warmup steps $=$ 7k; LR scheduler = Cosine Decay
Stage 2	CausalSemantic Bottleneck	Binaural spatial speeches with a duration of less than 20 seconds derived from all English samples in Emilia, GigaSpeech, LibriTTS-R, and VoxPopuli	$\mathcal{L}_{\text{distill}}$	Learning rate $= 3e^{-5}$; Warmup steps $=$ 2k; LR scheduler = Cosine Decay
Stage 3	CausalSemantic Bottleneck	Binaural spatial speeches with a duration of less than 12 seconds derived from all English samples in Emilia, LibriTTS-R, VoxPopuli, LJSpeech, and Hi-FiTTS	$\mathcal{L}_{\text{sr}}, \mathcal{L}_{\text{align}}$	Learning rate $= 3e^{-5}$; Warmup steps $=$ 1k; LR scheduler = Cosine Decay

$$\mathcal{L}_{\text{mag}} = \frac{1}{2}\sum_{c=1}^{2}\frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\Big|\log\big(|\mathbf{X}_c(t,f)|+\epsilon\big)-\log\big(|\hat{\mathbf{X}}_c(t,f)|+\epsilon\big)\Big| \tag{15}$$

where c denotes the c-th monaural speech signal, t and f denote the time frame and frequency bin, respectively, $\mathbf{X}_c(t,f)$ and $\hat{\mathbf{X}}_c(t,f)$ are the target and predicted complex STFTs, respectively, and ϵ is a small constant to prevent $\log(0)$.

Second, $\mathcal{L}_{\text{sc}}$ is good at capturing severe local distortion: when a single bin of the two monaural speech signals blows up, it gets amplified by the squaring. The formula can be written as:

$$\mathcal{L}_{\text{sc}} = \frac{1}{2}\sum_{c=1}^{2}\frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\Big[|\mathbf{X}_c(t,f)|-|\hat{\mathbf{X}}_c(t,f)|\Big]^2 \tag{16}$$

where c denotes the c-th monaural speech signal, t and f denote the time frame and frequency bin, respectively, and $\mathbf{X}_c(t,f)$ and $\hat{\mathbf{X}}_c(t,f)$ are the target and predicted complex STFTs.

Third, $\mathcal{L}_{\text{ph}}$ takes the absolute value of the per-frame phase difference between the two monaural speech signals, and is good at capturing absolute phase recovery errors. The formula can be written as:

$$\mathcal{L}_{\text{ph}} = \frac{1}{2}\sum_{c=1}^{2}\frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\Big|\angle\mathbf{X}_c(t,f)-\angle\hat{\mathbf{X}}_c(t,f)\Big| \tag{17}$$

where c denotes the c-th monaural speech signal, t and f denote the time frame and frequency bin, respectively, $\angle\mathbf{X}_c(t,f)$ and $\angle\hat{\mathbf{X}}_c(t,f)$ are the target and predicted phases, and the absolute value is taken per frame.

Forth, $\mathcal{L}_{\text{if}}$ focuses on the inter-frame phase difference between the two monaural speech signals, and is good at capturing phase discontinuity errors. The formula can be written as:

$$\mathcal{L}_{\text{if}} = \frac{1}{2}\sum_{c=1}^{2}\frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\left|\Delta\angle\mathbf{X}_c(t,f) - \Delta\angle\hat{\mathbf{X}}_c(t,f)\right| \tag{18}$$

where $\Delta\angle\mathbf{X}_c(t,f) = \angle\mathbf{X}_c(t+1,f) - \angle\mathbf{X}_c(t,f)$ and $\Delta\angle\hat{\mathbf{X}}_c(t,f) = \angle\hat{\mathbf{X}}_c(t+1,f) - \angle\hat{\mathbf{X}}_c(t,f)$ denote the inter-frame phase differences of the target and predicted signals, respectively.

Finally, $\mathcal{L}_{\text{ipd}}$ mainly supervises the interaural phase difference (IPD) and interaural time difference (ITD), while $\mathcal{L}_{\text{e}}$ mainly supervises the interaural level difference (ILD) as well as the overall energy:

$$\mathcal{L}_{\text{ipd}} = \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\left|(\angle\mathbf{X}_1(t,f) - \angle\mathbf{X}_2(t,f)) - \left(\angle\hat{\mathbf{X}}_1(t,f) - \angle\hat{\mathbf{X}}_2(t,f)\right)\right| \tag{19}$$

$$\mathcal{L}_{\text{e}} = \frac{1}{2}\sum_{c=1}^{2}\frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\left[\log\big(|\mathbf{X}_c(t,f)| + \epsilon\big) - \log\big(|\hat{\mathbf{X}}_c(t,f)| + \epsilon\big)\right]^2 \tag{20}$$

where $\angle\mathbf{X}_c(t,f)$ and $\angle\hat{\mathbf{X}}_c(t,f)$ are the target and predicted phases, $\mathbf{X}_c(t,f)$ and $\hat{\mathbf{X}}_c(t,f)$ are the target and predicted complex STFTs, and ϵ prevents $\log(0)$.

During this training process, except for the losses that measure the difference between the two ears, all other losses computed on monaural speech are calculated separately for the left and right channels and then summed. We use the binaural spatial speeches with a duration of less than 20 seconds derived from all English samples in Emilia, GigaSpeech, LJSpeech, and HiFiTTS. Futhermore, we set the loss combination weights λ_{sr}, λ_{adv}, λ_{fm} and λ_{kl} to be 1.0, 0.1, 0.1, and $1e^{-6}$, respectively. The learning rate is linearly warmed up from 0 to a peak of $3e^{-5}$ over the first 7k steps, then smoothly decayed to nearly $3e^{-6}$ following a cosine curve. We train the StereoEncoder and StereoDecoder for two epochs to ensure the acoustic and spatial reconstruction capability.

C.2.2 Stage 2: Semantic Distillation Training

We employ a CausalSemantic Bottleneck as a semantic bottleneck to extract rich semantically relevant features from the acoustic representations output by the StereoEncoder. Following MingUniAudio, this bottleneck adopts the remaining modules of the Whisper-large-v3 encoder with the Convolutional layers removed, while replacing its bidirectional attention with causal attention. To preheat the CausalSemantic Bottleneck for the joint reinforcement training of acoustic, semantic and spatial attributes in the third stage, we first distill the capability of the pretrained Whisper-large-v3 encoder to extract semantic information. Since the Whisper encoder only accepts monaural (single-channel) input, we first downmix the binaural (two-channel) speech to monaural speech and then employ Whisper encoder to extract the target semantic representations. This downmixing process mainly discards spatial cues (e.g., ITD/ILD), while largely preserving semantic content as shown in Table 12, though it may introduce mild phase cancellation. The loss function for this stage is calculated as follows:

$$\mathcal{L}_2 = \mathcal{L}_{\text{distill}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{Z}_{\text{uni}}^{(i)} - \mathbf{Z}_{\text{semantic}}^{(i)}\right\|_2^2 \tag{21}$$

where the distillation loss $\mathcal{L}_{\text{distill}}$ is computed as the mean squared error (MSE) between the unified features $\mathbf{Z}_{\text{uni}}$ and the Whisper representations of identical shape $\mathbf{Z}_{\text{semantic}}$, so as to encourage the unified features to absorb as much semantically relevant information as possible. N denotes the batch size.

Table 12: Effect of the downmixing method on semantic and acoustic attributes. Spatial binaural speech rendered from the LibriSpeech test-clean is adopted for evaluation. "WER" measures whether the downmixed monaural speech incurs semantic loss with respect to the binaural speech, while "TimberSIM" to the orignial reference speech measures whether the downmixed monaural speech incurs timbre loss with respect to the binaural speech.

Input Speech	**WER**	**TimberSIM (vs. Ref)**
Left-channel	3.15 / 3.11 / 3.11	0.602 / 0.672 / 0.611
Right-channel	2.91 / 3.16 / 3.18	0.597 / 0.672 / 0.617
Downmixed monaural	2.93 / 3.18 / 3.16	0.583 / 0.660 / 0.581

During this training process, we use Binaural spatial speeches with a duration of less than 20 seconds derived from all English samples in Emilia, GigaSpeech, LibriTTS-R, and VoxPopuli as the distillation data. The the peak learning rate is also set to $3e^{-5}$. Since the StereoEncoder has already been initialized in the first stage, the number of warmup steps in the second stage can be reduced to 2k steps. We freeze the weights of the StereoEncoder and only train the CausalSemantic Bottleneck for approximately one epoch.

C.2.3 Stage 3: Joint Reinforcement Training of Acoustic, Semantic, and Spatial Attributes

We perform a joint supervised training on LLM-based semantic alignment and spatial speech reconstruction in the third stage. During this stage, we update only the parameters of CausalSemantic Bottleneck, thereby forcing the unified features to simultaneously reinforce semantic, spatial and acoustic attributes. We pre-align the unified representation with downstream ASR and spatial understanding tasks using an audio adapter and a smaller Qwen LLM backbone [10]. The loss function for this stage is calculated as follows:

$$\mathcal{L}_3 = \lambda_{\text{sr}} \cdot \mathcal{L}_{\text{sr}} + \lambda_{\text{align}} \cdot \mathcal{L}_{\text{align}} \tag{22}$$

where the semantic alignment loss $\mathcal{L}_{\text{align}}$ is used to map the unified features into the representation space of the language model in advance.

Considering training efficiency, we restrict the third stage to binaural spatial speeches with a duration of less than 12 seconds derived from all English samples in Emilia, LibriTTS-R, VoxPopuli, LJSpeech, and HiFiTTS. We freeze the weights of the StereoEncoder and StereoDecoder, while training the CausalSemantic Bottleneck for approximately one epoch. After the joint reinforcement in the third stage, the unified features contain rich semantic, acoustic, and spatial information to support downstream binaural speech understanding, generation and editing tasks.

C.3 Evaluation Metrics

To evaluate the reconstruction performance of SpiderTok, we render spatial binaural speech from the out-of-domain LibriTTS test-clean data, considering three motion states: static, moving, and orbiting. As is shown in Table 10, we comprehensively compare the reconstruction capability of SpiderTok and the baselines from two perspectives, namely speech quality and spatial fidelity, using a total of seven metrics, including PESQ, STOI, UTMOS, ANG Cos, DIS Cos, IPD Corr, and ILD Corr. For PESQ, STOI, and UTMOS evaluation, we uniformly downmix the binaural audio to monaural audio before computation, even though this may introduce a slight quality degradation. The other four metrics all quantify the spatial fidelity of binaural audio reconstruction from both perceptual and physical perspectives. The first two (ANG Cos and DIS Cos) are based on the pretrained spatial audio encoder Spatial-AST and measure the consistency of the learned direction/distance representations; the latter two (IPD Corr and ILD Corr) operate directly on interpretable physical cues, including binaural phase difference and binaural level difference, and are used to localize which specific cue degrades. All these four metrics are computed between the reconstructed binaural speech and the original input binaural speech, and higher values are better.

[10] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

C.3.1 PESQ

Perceptual evaluation of speech quality (PESQ) is an objective speech quality assessment method whose core goal is to predict the subjective mean opinion score (MOS). By simulating the auditory characteristics of the human ear, it compares the original reference signal with the distorted signal after transmission or processing, and finally outputs an objective MOS score.

The calculation process consists of five steps.

First, level alignment and filtering are performed: the reference and degraded signals are adjusted to a uniform standard listening level (79 dB SPL) and then filtered by an IRS-like filter that simulates a standard telephone handset. Second, time alignment is conducted: because signals are delayed during transmission, PESQ aligns the two signals in time by segmenting them and performing fine delay estimation and compensation for each segment to handle delay variations. Third, an auditory transform is applied: the aligned signals are passed through a psychoacoustic model and transformed into the time-frequency domain, which involves computing the short-time power spectrum (e.g., 32 ms frames with 50% overlap), applying frequency and gain compensation, and mapping the spectrum to loudness density on the Bark scale. Fourth, distortion parameters are extracted by computing the difference in loudness density between the reference and degraded signals, yielding two core parameters: symmetric disturbance (d_{SYM}), which is sensitive to all differences between the two signals, and asymmetric disturbance (d_{ASYM}), which is mainly sensitive to added interference components in the degraded signal that are absent from the reference. Finally, the two distortion parameters are aggregated over time and frequency and mapped to the final PESQ score (i.e., the objective MOS) via a linear regression model:

$$\mathrm{PESQ} = 4.5 - 0.1 \times d_{\mathrm{SYM}} - 0.0309 \times d_{\mathrm{ASYM}}, \tag{23}$$

where d_{SYM} and d_{ASYM} are the aggregated symmetric and asymmetric disturbance terms, respectively. The resulting PESQ score is typically reported in the range $[-0.5, 4.5]$, with higher values indicating better perceptual quality; in practice, it is often clipped to $[1, 4.5]$ when interpreted as an objective MOS.

C.3.2 STOI

Short-time objective intelligibility (STOI) directly predicts speech intelligibility, which refers to whether a listener can correctly recognize the speech content.

The calculation procedure consists of six steps.

First, silent frame removal is performed: frames with excessively low energy (e.g., 40 dB below the highest-energy frame) are removed from the input signal, since they contribute little to intelligibility. Second, time-frequency decomposition is conducted: the clean and degraded speech signals are decomposed into the time-frequency domain via the short-time Fourier transform (STFT), typically using a 25 ms Hanning window and a 256-channel frequency band division covering 0–5 kHz. Third, one-third octave band energy extraction is applied: the band energies are further merged into one-third octave bands, which is a key step in simulating the frequency resolution of the human auditory system; STOI mainly focuses on the low-frequency range from 150 Hz to 4.3 kHz. Fourth, short-time envelope normalization and clipping are carried out: the short-time spectral envelope of the degraded signal is normalized to eliminate the effect of overall gain differences, and the normalized envelope is clipped to limit its range of variation, avoiding the interference of extreme values in the correlation computation. Fifth, correlation computation is performed: for each one-third octave band and each short-time segment (about 386 ms), the correlation coefficient between the clean speech envelope and the clipped degraded speech envelope is computed, which represents the intelligibility contribution of that segment in that band. Finally, score aggregation is conducted: the correlation coefficients across all bands and all time segments are averaged to obtain the final STOI score:

$$\mathrm{STOI} = \frac{1}{J \cdot M} \sum_{j=1}^{J} \sum_{m=1}^{M} d_{j,m}, \tag{24}$$

where J is the number of one-third octave bands, M is the number of speech frames, and $d_{j,m}$ is the correlation coefficient at the j-th band and the m-th frame, which measures the envelope similarity between the clean and degraded speech in that local time-frequency unit. The resulting STOI score lies in $[0, 1]$, with higher values indicating better intelligibility.

Interpretation. The STOI score typically ranges from 0 to 1: 1 indicates that, from the perspective of intelligibility, the degraded signal is indistinguishable from the clean signal and is fully intelligible, while 0 indicates that it is completely unintelligible.

C.3.3 UTMOS

UTMOS differs fundamentally from PESQ and STOI in design philosophy: it does not rely on signal processing models, but instead uses deep learning to directly "learn" human subjective ratings. The core of UTMOS is ensemble learning, which comprises strong and weak learners. The strong learner serves as the main component of the model and is fine-tuned from a self-supervised learning (SSL) model (e.g., wav2vec 2.0). Its input includes not only the speech waveform but also multiple auxiliary information sources to improve prediction accuracy: a data domain ID, which identifies the dataset from which the speech originates and helps the model adapt to data from different sources; a listener ID, which is the most important input feature and is used to model individual preferences and biases of different listeners in subjective rating; and phoneme and reference sequences, which help the model assess pronunciation accuracy by comparing the phonemes of the synthesized speech with the real phonemes. The weak learner, in contrast, uses traditional machine learning methods (e.g., SVM) to predict scores from SSL features, thereby increasing model diversity. The final UTMOS score is obtained by ensembling the predictions of the strong and weak learners, and higher values indicate better predicted speech quality.

C.3.4 ANG Cos and DIS Cos

Following CSAVocoder (Zhu et al., 2026), ANG Cos and DIS Cos are mainly used in our work to measure the consistency between the reconstructed binaural audio and the ground truth in terms of distance and direction, including azimuth and elevation.

The procedure consists of five steps.

First, the audio is resampled to 32 kHz and normalized to -14 dBFS to align with the Spatial-AST (Zheng et al., 2024) training distribution. Second, it is segmented into 1-second chunks (zero-padded if shorter than 1 second); since Spatial-AST is only reliable for static sound sources, the source is assumed to be approximately stationary within 1 second. Third, four-channel input features are extracted for each segment: binaural log-mel spectrograms (`logmel_L`, `logmel_R`) and the cosine and sine of the binaural phase difference (`IPD_cos`, `IPD_sin`). The log-mel is obtained via STFT with

$$\texttt{n_fft} = 1024, \qquad \texttt{hop} = 320,$$

and a 128-dimensional mel filter bank covering 50–14 kHz, while the IPD is given by the phase difference between the left and right channel STFTs. Fourth, the features are passed through Spatial-AST (an 88M-parameter ViT), from which `doa_token` and `dis_token` (each 768-dimensional, after LayerNorm) are taken. Fifth, for each second, the cosine similarity between the recon and ground-truth tokens is computed (dot product after L2 normalization), then averaged over seconds:

$$\text{ANG Cos} = \frac{1}{S}\sum_{s=1}^{S}\frac{\mathbf{e}^{\text{doa}}_{\text{recon},s}\cdot\mathbf{e}^{\text{doa}}_{\text{gt},s}}{\left\|\mathbf{e}^{\text{doa}}_{\text{recon},s}\right\|\left\|\mathbf{e}^{\text{doa}}_{\text{gt},s}\right\|} \tag{25}$$

$$\text{DIS Cos} = \frac{1}{S}\sum_{s=1}^{S}\frac{\mathbf{e}^{\text{dis}}_{\text{recon},s}\cdot\mathbf{e}^{\text{dis}}_{\text{gt},s}}{\left\|\mathbf{e}^{\text{dis}}_{\text{recon},s}\right\|\left\|\mathbf{e}^{\text{dis}}_{\text{gt},s}\right\|} \tag{26}$$

where S is the number of seconds in the sample, $\mathbf{e}^{\text{doa}}_{\text{recon},s}$ and $\mathbf{e}^{\text{doa}}_{\text{gt},s}$ are the LayerNorm-normalized `doa_token` embeddings of the reconstruction and the ground truth for the s-th second, respectively, and $\mathbf{e}^{\text{dis}}_{\text{recon},s}$ and $\mathbf{e}^{\text{dis}}_{\text{gt},s}$ are the corresponding `dis_token` embeddings.

Interpretation. ANG_{Cos} and DIS_{Cos} are bounded in $[0, 1]$, and larger values are preferred, with 1 indicating complete consistency in the direction/distance representation.

C.4 IPD Corr

Interaural phase difference correlation (IPD Corr) is the primary cue for low-frequency grounding, but it is a per-frequency angular field (with $\pm\pi$ wrapping) rather than a scalar. To avoid errors caused

by phase wrapping, we compare the phase spectra of the reconstruction (recon) and the ground truth (gt) on the reliable frequency bands of IPD (300–1400 Hz) on a per-second basis using circular statistics.

The procedure consists of five steps.

First, the binaural STFT is computed with

$$\texttt{nperseg} = 512, \qquad \texttt{noverlap} = 384, \qquad f_s = 16 \text{ kHz},$$

and the per-frame, per-frequency IPD is calculated as

$$\phi(f,t) = \angle\left(L(f,t)\,\overline{R(f,t)}\right), \tag{27}$$

where $L(f,t)$ and $R(f,t)$ are the complex STFT coefficients of the left and right channels, respectively, $\overline{R(f,t)}$ denotes the complex conjugate of $R(f,t)$, and $\angle(\cdot)$ returns the phase angle in radians.

Second, the 300–1400 Hz band is selected, which is the reliable region where IPD is not severely wrapped. Let K be the number of frequency bins in this band.

Third, within each second, the IPD of all energy-gated frames in that second is circularly averaged to obtain the IPD spectral vector for that second,

$$\boldsymbol{\phi}_s \in \mathbb{R}^K,$$

where K is the number of frequency bins in the band. The energy gate is defined as

$$E(f,t) > 0.05 \max_{f,t} E(f,t),$$

or equivalently, frames whose energy is greater than 5% of the peak energy are retained. The circular average is computed as

$$\bar{\phi}_s(f) = \angle\left(\frac{1}{N_s}\sum_{t\in s} e^{\mathrm{j}\phi(f,t)}\right), \tag{28}$$

where N_s is the number of energy-gated frames in the s-th second, and $\mathrm{j} = \sqrt{-1}$.

Fourth, for each second, the circular correlation coefficient between the two angle vectors of the reconstructed binaural speech and the ground-truth binaural speech is computed. The circular correlation ρ_s^{circ} first subtracts the respective circular mean and then normalizes, correctly handling $\pm\pi$ wrapping:

$$\rho_s^{\text{circ}} = \frac{\sum_k \sin(\alpha_k - \bar{\alpha})\,\sin(\beta_k - \bar{\beta})}{\sqrt{\sum_k \sin^2(\alpha_k - \bar{\alpha})\;\sum_k \sin^2(\beta_k - \bar{\beta})}}, \tag{29}$$

where

$$\alpha = \bar{\boldsymbol{\phi}}_s^{\text{recon}}, \qquad \beta = \bar{\boldsymbol{\phi}}_s^{\text{gt}},$$

and $\bar{\alpha}, \bar{\beta}$ are the circular means of α and β, respectively. This is equivalent to correlating the centered unit complex vectors.

Finally, the results are averaged over seconds:

$$\text{IPD Corr} = \frac{1}{S}\sum_{s=1}^{S} \rho_s^{\text{circ}}, \tag{30}$$

where S is the total number of seconds.

Interpretation. The value lies in $[-1, 1]$, with higher values being better; a value close to 1 indicates that the phase spectra are highly consistent.

C.4.1 ILD CORR

Interaural level difference correlation (ILD Corr) is the primary cue for high-frequency grounding (the head shadow effect) and is a real-valued sequence in dB. It is complementary to the IPD/ITD cues (low-frequency temporal/phase cues).

The procedure consists of three steps.

First, the binaural STFT is computed with the same parameters as for IPD:

$$\texttt{nperseg} = 512, \qquad \texttt{noverlap} = 384, \qquad f_s = 16\ \text{kHz},$$

and the per-frame ILD is calculated in the high-frequency band 2000–8000 Hz, where the head shadow is most pronounced:

$$\mathrm{ILD}(t) = 20 \log_{10} \frac{\frac{1}{|\mathcal{M}|} \sum_{f \in \mathcal{M}} |L(f,t)|^2}{\frac{1}{|\mathcal{M}|} \sum_{f \in \mathcal{M}} |R(f,t)|^2} \quad (\mathrm{dB}), \tag{31}$$

where $\mathcal{M}$ denotes the set of frequency bins in the 2000–8000 Hz band, and $|\mathcal{M}|$ is its cardinality.

Second, for each second, the median of the ILD over the energy-gated frames in that second is taken to obtain the ILD scalar for that second. Since ILD is aggregated by the real-valued median, circular statistics are unnecessary.

Third, for each second s, the Pearson correlation coefficient $\rho_s^{\mathrm{Pearson}}$ between the two ILD scalar sequences of the reconstructed binaural speech and the ground-truth binaural speech is computed:

$$\rho_s^{\mathrm{Pearson}} = \frac{\sum_{t \in \mathcal{T}_s} \left(\mathrm{ILD}^{\mathrm{recon}}(t) - \overline{\mathrm{ILD}_s^{\mathrm{recon}}} \right) \left(\mathrm{ILD}^{\mathrm{gt}}(t) - \overline{\mathrm{ILD}_s^{\mathrm{gt}}} \right)}{\sqrt{\sum_{t \in \mathcal{T}_s} \left(\mathrm{ILD}^{\mathrm{recon}}(t) - \overline{\mathrm{ILD}_s^{\mathrm{recon}}} \right)^2} \sqrt{\sum_{t \in \mathcal{T}_s} \left(\mathrm{ILD}^{\mathrm{gt}}(t) - \overline{\mathrm{ILD}_s^{\mathrm{gt}}} \right)^2}}, \tag{32}$$

where $\mathcal{T}_s$ is the set of frame indices within the s-th second, and $\overline{\mathrm{ILD}_s^{\mathrm{recon}}}$ and $\overline{\mathrm{ILD}_s^{\mathrm{gt}}}$ are the means of the frame-level ILD values within the s-th second for the reconstruction and the ground truth, respectively.

The final ILD Corr is then averaged over seconds:

$$\text{ILD Corr} = \frac{1}{S} \sum_{s=1}^{S} \rho_s^{\mathrm{Pearson}}, \tag{33}$$

where S is the total number of seconds.

Interpretation. The value lies in $[-1, 1]$, with higher values being better. The ILD itself is affected by speech content and is relatively noisy; a value above 0.85 is therefore regarded as well preserved.

C.5 Effectiveness of Training Stage 2: Semantic Distillation Training

As is illustrated in Table 6 and Figure 5, the importance of preheating the CausalSemantic Bottleneck in this stage greatly facilitates the rapid convergence of the semantic alignment in the third stage and improves the perceptual speech quality like PESQ, STOI and UTMOS of the final reconstructed binaural speech. The two runs follow the same three-phase trajectory: a steep descent through roughly the first 25k steps (during which the baseline completes 51% of its total fall but the cold start only 34%), a slower decline to about 75k, and a plateau thereafter. Because it starts higher (3.64 vs. 2.80) and descends more slowly, the gap is non-monotonic: it widens to a peak of $+1.49$ at around 15k steps, then narrows steadily to a persistent $+0.18$ across the final 50k (range 0.170–0.192), where both curves are flat. Skipping the middle stage therefore costs a slower start plus a small permanent offset, not a different convergence level.

Table 13: SpiderAudio architecture overview.

Module	Layer	Shape / Dimension
Input binaural speech		$(B, 2, N_1)$
SpiderTok Encoder	StereoEncoder	$(B, 2, N_1) \rightarrow (B, 2T_1, 32)$
	CausalSemantic Bottleneck	$(B, 2T_1, 32) \rightarrow (B, T_1, 1280)$
Unified feature		$(B, T_1, 1280)$
Audio Adapter	Pooling + Linear*2	$(B, T_1, 1280) \rightarrow (B, \lfloor T_1/5 \rfloor, 3584)$
Projected feature		$(B, \lfloor T_1/5 \rfloor, 3584)$
LLM Backbone	Qwen2.5-7B	3584 ($28 heads * 128$)
Output hidden states		$(B, L, 3584)$
Text Decoder	Text Head	$(B, L_t, 3584) \rightarrow (B, L_t, 152064)$
	Text Decoder	$(B, L_t, 152064) \rightarrow (B, L_t)$
Output text		(B, L_t)
Audio Decoder	DiT-base Audio Head	$(B, L_a, 3584) \rightarrow (B, T_2, 1280)$
	Stop Head	$(B, L_a, 3584) \rightarrow (B, T_2)$
	StereoDecoder	$(B, T_2, 1280) \rightarrow (B, 2, N_2)$
Output binaural speech		$(B, 2, N_2)$

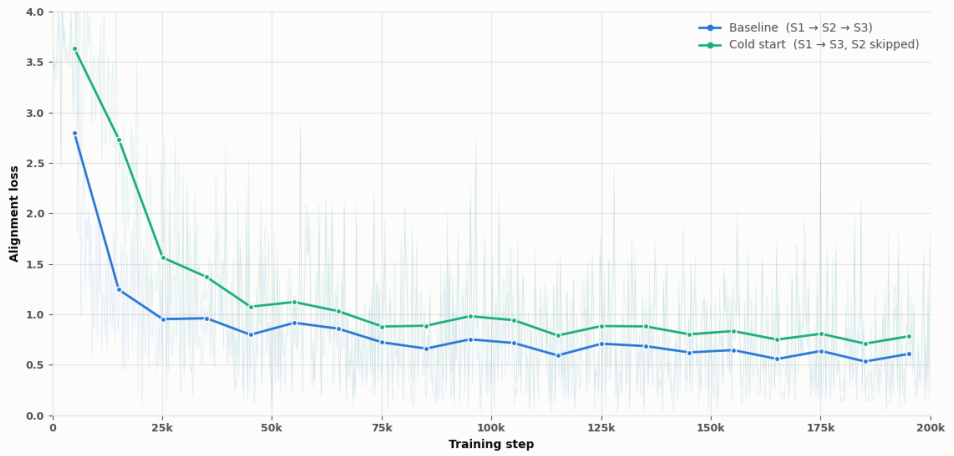


Figure 5: Comparison of the alignment loss in Stage 3 with and without the semantic distillation training stage in stage 2. "Baseline" means that semantic distillation is used before training the third stage, whereas "cold start" means it is not used. Marker represents mean alignment over each 10k step bin (21 bins). Translucent trace behind represents every raw value the runs logged, one point per 50 steps.

D SPIDERAUDIO

In this section, we provide additional implementation details about the proposed SpiderAudio. Figure 3 provides an overview of the SpiderAudio framework and training strategy.

D.1 MODEL ARCHITECTURE AND CONFIGURATION

Table 13 shows the detailed architecture and configuration of the SpiderAudio framework, which has a total of 8.15B parameters.

The unified representation with a shape of $(B, T_1, 1280)$ is downsampled by a factor of 5 and projected into the Qwen2.5 text representation space, resulting in a shape of $(B, \lfloor T_1/5 \rfloor, 3584)$. After being concatenated with the query or instruction text embedding, it is fed into the Qwen2.5-7B model, yielding a hidden state output of shape $(B, L, 3584)$.

In addtion, we uniformly adopt the template "`<text><text>...</text><audio>...`" as the output label, where the output text and binaural speech fill in the corresponding slots. As depicted in Algorithm 1, SpiderAudio sequentially decodes the overall hidden states using the text decoder and the audio decoder to text and speech output, which is guided by speical tokens `<text>`, `</text>` and `<audio>`. Briefly, the hidden states are first decoded into text by the text decoder; once `<audio>` is emitted, decoding switches to the audio decoder to produce the speech latent and finally the binaural audio. In speech understanding tasks, the stop head terminates decoding once the first audio latent is predicted, and thus the model produces only the understood text. In speech generation tasks, the model predicts `</text>` immediately following `<text>`, so only the generated binaural speech is produced. Only under the editing mode with the "Thinking-Before-Generation" scheme does the model first decode the thinking process before generating the corresponding binaural speech.

Algorithm 1 Sequential Text–Speech Decoding for SpiderAudio

Require: input binaural waveform $\mathbf{W}_{\text{ref}}$; instruction u; stop threshold τ; minimum speech steps m; step caps $N_{\text{txt}}, N_{\text{aud}}$; flow matching steps S; audio frames per chunk L_p

Ensure: output text $\mathbf{y}$; output binaural waveform $\hat{\mathbf{W}}$

Phase 0: Embedding

$\mathbf{E}_{\text{ref}} \leftarrow \textsc{ExtractAudioFeature}(\mathbf{W}_{\text{ref}})$ ▷ $\mathbf{E}_{\text{ref}}$: reference embeddings

$\mathbf{h} \leftarrow \textsc{Concat}\big(\textsc{Embed}\big(\mathbf{p}(u)\big),\ \mathbf{E}_{\text{ref}}\big)$ ▷ $\mathbf{h}$: LLM input embeddings; $\mathbf{p}$: prompt token ids

$\kappa \leftarrow \varnothing;\quad \mathcal{H} \leftarrow \mathbf{0}$ ▷ κ: semantic-branch KV cache; $\mathcal{H}$: flow latent history

Phase 1: Text decoding

$\mathbf{y} \leftarrow [\,];\quad y \leftarrow \varepsilon$ ▷ $\mathbf{y}$: emitted text; y: current token

while $y \neq \langle\text{audio}\rangle$ **and** $|\mathbf{y}| < N_{\text{txt}}$ **do**

  $y \leftarrow \textsc{Sample}\big(\text{LLM}(\mathbf{h})_{\text{logits}}\big)$

  $\mathbf{y} \leftarrow \mathbf{y} \,\|\, [y];\quad \mathbf{h} \leftarrow \mathbf{h} \,\|\, \textsc{Embed}(y)$

end while

Phase 2: Speech decoding

$\mathbf{Z} \leftarrow [\,]$ ▷ $\mathbf{Z}$: accumulated audio latent chunks

while $|\mathbf{Z}| < N_{\text{aud}}$ **do**

  $\mathbf{c} \leftarrow \text{LLM}(\mathbf{h})_{\text{last}}$ ▷ $\mathbf{c}$: flow condition (last hidden state)

  $\mathbf{z} \leftarrow \textsc{FlowSample}(\mathbf{c}, \mathcal{H}; S)$ ▷ $\mathbf{z}$: one sampled latent chunk

  $\mathbf{Z} \leftarrow \mathbf{Z} \,\|\, [\mathbf{z}];\quad \mathcal{H} \leftarrow \mathbf{z}$

  $(\mathbf{u}, \kappa) \leftarrow \textsc{LatentToUniStreaming}(\mathbf{z}, \kappa)$ ▷ $\mathbf{u}$: unified features

  $\mathbf{h} \leftarrow \mathbf{h} \,\|\, \textsc{Proj}\big(\textsc{Mean}(\mathbf{u})\big)$

  if $|\mathbf{Z}| > m$ **and** $\textsc{StopHead}(\mathbf{c}) > \tau$ **then**

    break

  end if

end while

$\hat{\mathbf{W}} \leftarrow \textsc{DecodeFromLatent}\big(\textsc{Concat}(\mathbf{Z})\big)$ ▷ $\hat{\mathbf{W}}$: binaural waveform

return $\mathbf{y}$, $\hat{\mathbf{W}}$

D.2 Training Details of SpiderAudio

As illustrated in Figure 3, the training of SpiderAudio mainly includes two main stages: (1) Cross-modal alignment, (2) Multi-task joint training. Table 14 presents the implementation details for these stages including training object, training data, loss functions and learning rates. All training data originates from open-source high-quality monaural (single-channel) speech data, with being rendered into binaural spatial speech based on HRTF, encompassing three scenarios: static, linear moving, and orbiting. All experiments in this training process are performed on 128 NVIDIA H100 GPUs. More detailed information about the two stages is introduced below:

Table 14: Training stages and implementation details.

Stage	Training Object	Training Data	Loss	Implementation Detail
Stage 1	Audio Adapter	ASR data in SpiderData	$\mathcal{L}_{ce}$	Learning rate = $1e^{-4}$ Warmup step = 5k LR scheduler = Linear decay
Stage 2	All Modules except SpiderTok	All data in SpiderData	$\mathcal{L}_{ce}, \mathcal{L}_{flow}, \mathcal{L}_{stop}$	Learning rate = $1e^{-5}$ Warmup step = 5k LR scheduler = Linear decay

D.2.1 Stage 1: Cross-Modal Alignment Training

To achieve cross-modal alignment and preserve the intelligence of the LLM backbone, we first train an Audio Adapter on the simplest ASR task to map the unified speech representation into the Qwen text representation space. During this process, only the parameters of the Audio Adapter are trainable, while those of all other modules are frozen. We train on the ASR portion of the constructed full-task dataset SpiderData, with an initial learning rate of $1e^{-4}$, a warmup of 5k steps, linear decay, and a total of half epoch. The loss function for this stage is calculated as follows:

$$\mathcal{L}_1 = \mathcal{L}_{\text{ce}} \tag{34}$$

where the $\mathcal{L}_{\text{ce}}$ here is the cross-entropy loss computed between the predicted transcription and the ground-truth transcription.

D.2.2 Stage 2: Multi-Task Joint Training

As it is highly challenging for a single LLM backbone to generate both text content and audio latent simultaneously from the beginning using both the cross-entropy loss and flow matching loss, we further divide the second stage into two sub-stages. In the first sub-stage, we mix at a ratio of $1:3$ the training data from binaural speech understanding tasks, including spatial grounding and ASR, and generation tasks, including zero-shot spatial TTS and instruct spatial TTS. In the second sub-stage, we further incorporate the spatial speech editing task, which improves the ability of SpiderAudio to first think and then generate. During the second sub-stage, we mix understanding, generation, and editing tasks at a ratio of $1:1:2$. The loss function for this stage is calculated as follows:

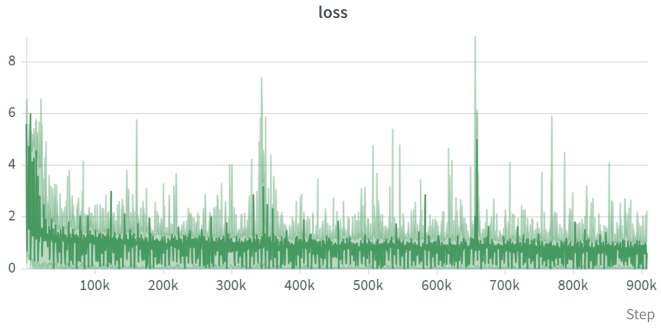
$$\mathcal{L}_2 = \lambda_{\text{ce}} \cdot \mathcal{L}_{\text{ce}} + \lambda_{\text{flow}} \cdot \mathcal{L}_{\text{flow}} + \lambda_{\text{stop}} \cdot \mathcal{L}_{\text{stop}} \tag{35}$$

where $\mathcal{L}_{\text{ce}}$ is the general cross-entropy loss of the LLM for understanding task, while $\mathcal{L}_{\text{flow}}$ and $\mathcal{L}_{\text{stop}}$ are the main losses in the generation process. $\mathcal{L}_{\text{flow}}$ is a training objective commonly used in generative models, especially in diffusion-related fields, whose core idea is to let the neural network learn a velocity field. $\mathcal{L}_{\text{stop}}$ is employed to teach SpiderAudio when to halt generation; otherwise, the speech would be generated endlessly. Moreover, $\mathcal{L}_{\text{flow}}$ and $\mathcal{L}_{\text{stop}}$ are zero for understanding tasks, and $\mathcal{L}_{\text{ce}}$ remains zero for generation tasks. Only in editing tasks, all losses are all non-zero. In addition, the wights λ_{ce}, λ_{flow} and λ_{stop} are set to 1.0, 1.0 and 0.01, respectively.

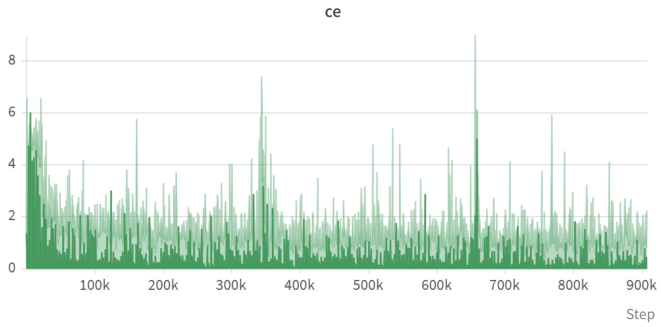
Figure 6 illustrates the training loss curves in the second stage, where sub-figures (a), (b), (c), and (d) correspond to the overall loss, cross-entropy loss, flow matching loss, and stop loss, respectively. The loss decreased monotonically throughout training (median from 1.29 to 0.75 by -41.9%), with the stop-prediction head converging within 80k steps, the cross-entropy loss within 300k, and the flow-matching loss within 800k; no objective exhibited overfitting.

D.3 Evaluation Metrics

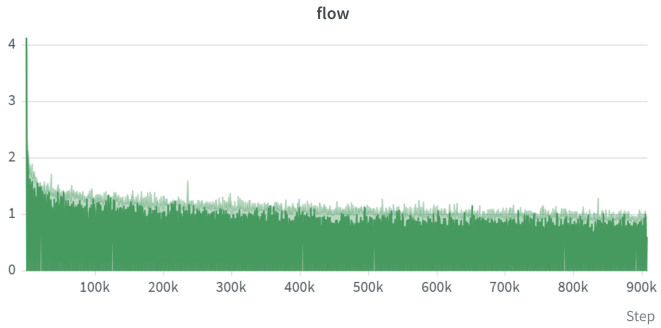
To evaluate the all-around capabilities of SpiderAudio in binaural speech understanding, generation, and editing from multiple perspectives, we build a comprehensive evaluation framework suite for each task, as shown in Table 2, Table 3 and Table 4.



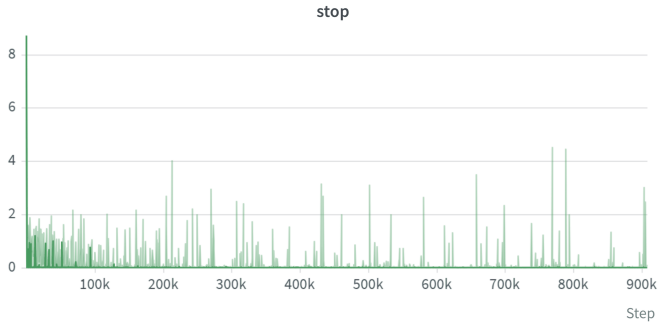
(a) Total loss



(b) Cross-entropy loss



(c) Flow matching loss



(d) Stop loss

Figure 6: The loss curves for the first 800K steps of the second training stage for SpiderAudio. The shaded bars in the figure represent noisy data from extreme outliers.

Table 15: Number distribution of eight-class direction in the evaluation set.

Class	Static	Moving Startpoint	Moving Endpoint	Orbiting Startpoint	Orbiting Endpoint
Front	331	320	315	338	345
Front-left	325	327	312	340	317
Left	318	330	347	324	340
Back-left	341	338	322	349	361
Back	320	311	345	329	311
Back-right	351	339	305	313	337
Right	327	310	323	279	332
Font-right	307	345	351	348	277
Total	2620	2620	2620	2620	2620

Table 16: Number distribution of three-class updown in the evaluation set.

Class	Static	Moving Startpoint	Moving Endpoint	Orbiting Startpoint	Orbiting Endpoint
Below	1224	1254	1218	842	842
Level	709	702	712	1463	1463
Above	687	664	690	315	315
Total	2620	2620	2620	2620	2620

D.3.1 Binaural Speech Understanding

The capability of binaural speech understanding considered in our work mainly includes spatial understanding and binaural speech recognition. We evaluate it using binaural speech rendered from the out-of-domain LibriSpeech test-clean data, using a generation process similar to training data for binaural understanding tasks in SpiderData.

In terms of spatial understanding, we mainly examine the models' capabilities in spatial grounding and motion state recognition. Spatial grounding mainly includes azimuth, elevation, and distance estimation, while motion state recognition mainly involves identifying whether the sound source is static, moving linearly, or orbiting. Specifically, following BAT (Zheng et al., 2024), we use an eight-class classification scheme for azimuth to evaluate its accuracy, including front, back, left, right, front-left, back-left, front-right, and back-right. Adjacent classes are separated by the bisector in between and the number distribution in the evaluation set is shown in Table 15. As for elevation, we divide the region from bottom to top into three classes: below, level, and above, corresponding to less than -10°, -10° to 10°, and greater than 10°, and the number distribution in the evaluation set is shown in Table 16. Distance prediction is classified as near ($<$ 1.0m), mid (1.0–3.0 m), and far ($>$ 3.0m), as shown in Table 17. For the moving and orbiting motion states, we uniformly use the average of the static results at the first and last endpoints as the classification criterion.

In terms of binaural speech recognition, since binaural speech recognition has rarely been explored in prior work, we compare against the approach of downmixing binaural speech to monaural speech and then using state-of-the-art (SOTA) non-spatial unified understanding and generation LALMs for ASR comparison.

D.3.2 Binaural Speech Generation

We mainly evaluate two common generation capabilities, zero-shot spatial TTS and instruct spatial TTS, using the English subsets of Seed-TTS (Anastassiou et al., 2024) and InstructTTSEval DSD split (Huang et al., 2025) as rendering targets, respective.

In terms of zero-shot spatial TTS, we provide SpiderAudio with a binaural speech as the reference audio, and it imitates the timbre and spatial state while generating the specified content. To demonstrate its all-around capability, we first downmix the generated binaural speech into monaural speech, and then evaluate its content accuracy and speaker similarity to the reference audio, with WER and TimberSIM evaluated via Whisper-large-v3 and WavLM-base evaluation model [11], respectively. As

[11] https://huggingface.co/bezzam/wavlm_large_finetune_seed_tts_eval

Table 17: Number distribution of three-class distance in the evaluation set.

Class	Static	Moving Startpoint	Moving Endpoint	Orbiting Startpoint	Orbiting Endpoint
Near	823	894	873	928	882
Mid	860	876	845	870	875
Far	937	850	902	822	863
Total	2620	2620	2620	2620	2620

Table 18: Distribution of data across different attributes and motion conditions.

Motion State	Content	Emotion	Prosody	Paralinguistic	Spatial State	Rendering
Static	300	750	300	200	300	300
Moving	300	750	300	200	300	300
Orbiting	300	750	300	200	300	300
Total	900	2250	900	600	900	900

for the spatial similarity between the generated binaural audio and the reference audio, we adopt the ANG Cos and DIS Cos metrics for evaluation.

In terms of instruct spatial TTS, following AuK (Ma et al., 2026), we additionally assess the description-speech consistency (DSD) besides spatial state accuracy. First, DSD mainly measures the model's ability to accurately translate abstract, unstructured style descriptions into actual speech features. In our work, Gemini-3-flash is used as the judge model. Since there is no reference audio in this task, we downmix the generated binaural speech to monaural audio, and then re-render the downmixed monaural audio into binaural speech with the ground-truth spatial state according to the spatial position and motion state specified in the instruction, which serves as the reference audio for evaluating ANG Cos and DIS Cos.

D.3.3 Binaural Speech Editing

We mainly evaluate six editing capabilities of SpiderAudio, including regular editing (i.e., content, emotion, prosody, and paralinguistic editing) and spatial editing (i.e., spatial state editing and binaural speech rendering via HRTF). We use the Content, Emotion, Prosody, and Paralinguistic subsets of SpeechEditBench (Zhang et al., 2026) as the raw monaural data to be rendered for evaluation. In addition, the raw monaural data for evaluating spatial state editing and binaural speech rendering capabilities is based on the Content and Prosody subsets, respectively. In addition, to ensure that performing a given editing instruction does not affect other attributes of the speech (e.g., removing a word should not severely alter the timbre), we adopt the stricter "Joint Success" evaluation protocol from SpeechEditBench. During the evaluation of editing capability, all metrics requiring LLM-as-a-judge evaluation are assessed using Gemini-3-Flash. Moreover, for spatial state editing and binaural speech rendering, we uniformly consider the editing task successful when ANG Cos is greater than 0.85 and DIS Cos is greater than 0.95. The final number distribution of each sub-task is shown in Table 18.

D.4 Effectiveness of Spatial Grounding

To more intuitively demonstrate SpiderAudio's capability on the spatial grounding tasks, in addition to the coarse-grained classification accuracy reported in the main text, we further provide the mean absolute error (MAE) of azimuth, elevation, and distance predictions on spatial binaural speech here, as shown in Table 19.

The experimental results robustly demonstrate the strong spatial grounding capability of the current model, particularly highlighting its exceptional distance estimation performance. While the MAE results for azimuth and elevation naturally increases under dynamic conditions (moving and orbiting) due to the inherent challenges of tracking motion, the model maintains remarkable stability in distance prediction, with errors staying consistently low between 1.11 and 1.25 across all motion states. Furthermore, even in the most challenging "Moving" scenario, the model achieves a highly

Table 19: MAE results of spatial grounding metrics across different motion conditions.

	Azimuth MAE	Elevation MAE	Distance MAE
Static	2.76	2.18	1.11
Moving	8.99	6.88	1.25
Orbiting	8.16	4.83	1.15

respectable elevation MAE of 6.88, proving its reliable spatial awareness and robustness against dynamic environmental changes.

E BASELINES

Since open-source work on spatial speech is scarce, we use two groups of baselines: state-of-the-art monaural models from academia and industry for the basic capabilities, and the few available binaural models for the spatial ones.

First, in terms of the reconstruction capability, we use the monaural speech tokenizers, including MiMo-Audio Tokenizer (Zhang et al., 2025), Higgs Tokenizer (Boson, 2025), HoliTok (Li et al., 2026a), MingTok-Audio (Yan et al., 2025a), X-Codec (Ye et al., 2025), WavTokenizer (Ji et al., 2025), Encodec (Défossez et al., 2022), and binaural audio tokenizers, including DualSpec-VAE (Zhao et al., 2026) and BANC (Ratnarajah et al., 2025).

Second, in terms of speech understanding and generation, we use non-spatial unified understanding and generation models such as Audex (Kong et al., 2026), Uniaudio 2.0 (Yang et al., 2026), Mimo-Audio (Zhang et al., 2025), Ming-UniAudio (Yan et al., 2025a), FireRedAudio (Li et al., 2026b), and Qwen2.5-Omni (Xu et al., 2025) for basic capability comparison. In addition, for spatial audio grounding capability, we also include existing works on spatial sound grounding like BAT (Zheng et al., 2024), OWL (Biswas et al., 2026), and DSpAST (Wilkinghoff & Tan, 2026) for comparison.

Third, regarding spatial speech editing capability, we primarily compare against current SOTA open-source non-spatial models like Unison (Li et al., 2026c), AuK (Ma et al., 2026), Step-Audio-EditX (Yan et al., 2025b) and Ming-UniAudio on regular editing tasks.

F LIMITATIONS

HRTF adaptation. This work does not explicitly address adaptation to listener-specific HRTFs. Personalized spatial rendering and generalization across listeners with different HRTFs remain to be investigated. Of course, many tech products today, such as Apple earphones, are capable of actively learning and adapting to an individual user's HRTF. In the future, we will continue to explore HRTF parameter adaptation.

Multi-speaker scenarios. Our current task design focuses on single-speaker far-field scenario and does not include multi-speaker tasks. Extending the framework to overlapping speech while maintaining distinct spatial cues for each speaker remains future work.