MOSS-AUDIO: UNIFIED AUDIO UNDERSTANDING ACROSS SEMANTICS, ACOUSTICS, AND TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified audio understanding requires integrating semantic content with fine-grained acoustic cues and temporal structure. We present MOSS-Audio, a family of audio-language models for unified understanding and reasoning across speech, environmental sound, and music. An audio encoder trained from scratch and DeepStack cross-layer feature injection provide the language decoder with acoustic features at multiple abstraction levels. Explicit time markers anchor features to elapsed time for timestamped transcription and time-aware question answering through autoregressive generation. Our annotation pipeline segments recordings at event boundaries and merges specialist outputs into unified captions that preserve acoustic detail and event chronology, while retaining intermediate annotations for task-specific instruction tuning. Large-scale pretraining and staged post-training yield instruction-following (Instruct) and reasoning (Thinking) variants. Across evaluation, MOSS-Audio achieves the highest average accuracy among evaluated open-source systems, including 30B-scale models. MOSS-Audio also leads evaluated open and proprietary systems in average speech-captioning score and achieves more accurate timestamp alignment than evaluated omni-model baselines. We will release model weights, training data, code, and complete training recipes to support further research in unified audio understanding.

1 INTRODUCTION

Audio-language models have made substantial progress in bringing speech, environmental sound, and music into a shared language interface. Existing approaches include combining specialized speech and sound encoders, as in SALMONN (Tang et al., 2024), and adapting a shared audio front end through broad audio-language training, as in Qwen2-Audio and Audio Flamingo 3 (Chu et al., 2024; Ghosh et al., 2025). These advances extend audio modeling beyond transcription toward description, question answering, and reasoning, with potential applications in voice agents that must interpret both spoken content and surrounding acoustic context.

Despite this progress, two challenges remain central to unified audio understanding. **(1) Constructing supervision around coherent acoustic events.** Speech, environmental sounds, and music follow different temporal boundaries. Cuts based only on speech activity can fragment non-speech events and separate them from relevant context, while their descriptions require complementary annotations from different specialists. **(2) Making acoustic and temporal information available to the language decoder.** Transcription, vocal description, sound recognition, and music understanding depend on different acoustic cues. Analyses of speech encoders show that acoustic and linguistic information vary with depth (Pasad et al., 2021; Yang et al., 2021), motivating an examination of what intermediate representations contribute beyond the final encoder output. Audio is also inherently temporal: understanding its temporal structure requires relating acoustic content to when it occurs and how it unfolds. Together, these challenges concern both what evidence is retained in a training example and how its acoustic features reach the decoder.

These challenges motivate a central research question: **How can a unified audio-language model preserve event context in its training data and use acoustic representations from different encoder depths together with explicit temporal references during decoding?** Consider an alarm that begins before a spoken utterance and continues after it. Understanding the recording may require identifying the alarm, describing the speaker's delivery, and relating the two events in time. A

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.

useful model should encounter training examples that retain these relationships and have access to acoustic features that support the different aspects of the task.

To address these challenges, we present **MOSS-Audio**, a unified audio-language model family for speech, environmental sound, and music. To preserve event continuity in training examples, we use detected acoustic events to guide segmentation and annotation routing. Specialist annotations are then fused into coherent descriptions for pretraining and retained individually for task-oriented instruction tuning, allowing the same recording to provide both integrated and task-specific supervision. To support the diverse acoustic requirements of these tasks, we train a dedicated audio encoder from scratch with broad audio supervision. We further adapt DeepStack (Meng et al., 2024) to inject intermediate acoustic features into early language-model layers, giving the decoder access to representations from multiple depths alongside the final encoder output. To help the model learn temporal structure, we interleave explicit time tokens (time markers) with the audio features during training, alongside word- and sentence-level timestamp supervision. Building on these data and modeling designs, we train MOSS-Audio at two scales to support further research and development in the audio understanding community.

Experiments demonstrate strong audio understanding performance, while component comparisons evaluate the dedicated encoder and characterize the content-dependent effects of cross-layer conditioning. Our main contributions are:

- We introduce **MOSS-Audio**, a unified audio-language model family for speech, environmental sound, and music. It achieves state-of-the-art performance in general audio understanding.
- We propose an **event-oriented data pipeline** that integrates segmentation, specialist annotation, and supervision fusion to preserve acoustic event context. We will release the resulting training data to support further research in the audio understanding community.
- We develop a **dedicated audio encoder with DeepStack-based cross-layer conditioning** to support detailed audio captioning. The encoder improves Clotho and MECAT scores over the compared baselines, while controlled 0.6B-decoder experiments show gains in pure-sound and pure-music captioning with DeepStack. We will release both components to advance research in audio representation learning and understanding.
- We introduce **explicit time tokens** during training to model the inherent temporal structure of audio, helping the model associate acoustic content with elapsed time and generate temporally grounded outputs.

We will release model weights, training data, code, and complete training recipes to support further research in the audio understanding community.

2 Event-Oriented Audio Data Pipeline

Unified audio understanding requires supervision for spoken content, acoustic attributes, and temporal structure. Transcripts describe what is said, but leave vocal delivery, environmental sounds, and music largely unspecified. Global captions provide broader descriptions, yet can omit local events and their relationships. We therefore construct complementary supervision through an event-oriented pipeline: segment recordings using detected events, select annotation branches according to each clip's content, and assemble their outputs into transcription, captioning, and question-answering targets (Figure 1).

2.1 Event-Guided Segmentation and Routing

A training clip should retain enough context to describe an event and its relationship to surrounding activity. Fixed-duration cuts can interrupt events, while speech-only segmentation overlooks non-speech sounds extending across utterance boundaries. A BEATs-based PretrainedSED model supplies timestamped AudioSet events (Chen et al., 2023; Schmid et al., 2024; Gemmeke et al., 2017). We first merge speech intervals across short pauses, then combine them with other events, merge overlaps, add boundary padding, and cut at event gaps under a duration cap. Persistent non-speech events longer than 60 seconds are excluded from boundary computation, while their labels remain available for annotation.

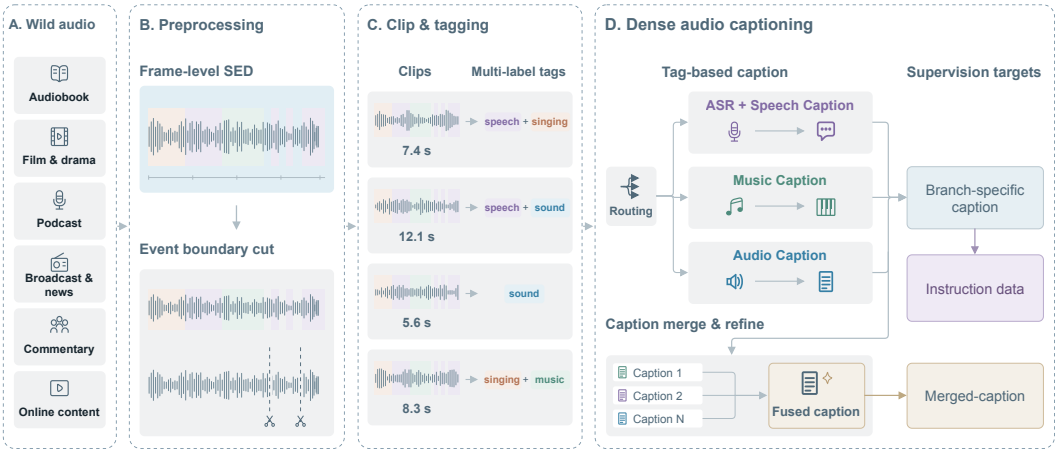

Figure 1: Event-oriented data construction. Event detection guides clip boundaries and annotation branches. Specialist outputs are filtered and fused into chronological captions for pretraining, and retained separately for task-specific instruction data. Illustrated durations are examples.

Each clip receives a multi-label content profile. We map its events to nine coarse categories and compute their covered durations by merging within-category overlaps. These profiles select annotation branches; mixed clips can enter multiple branches so that speech, accompanying music, and background events contribute supervision for the same audio.

2.2 Complementary Annotation Branches

We use specialized models to annotate spoken content and timing, speaker characteristics, sound events, and musical attributes. Each branch produces domain-specific annotations that provide complementary evidence for the subsequent large-model merge stage.

Transcription and CTC alignment. For clips containing speech or singing, ASR models generate transcripts. We then apply CTC-based forced alignment to associate the recognized words with their start and end times. These word-level annotations provide lexical content and temporal boundaries; punctuation and timing heuristics group them into sentence-level segments for timestamped transcription supervision.

Speaker-specific speech captioning. We train a dedicated speech captioning model to describe the vocal characteristics of individual speakers. For clips containing multiple speakers, the model produces speaker-specific descriptions of attributes such as pitch, loudness, speaking rate, emotion, and vocal style. These annotations complement the transcripts with information about each speaker's voice and delivery.

Sound-event annotation. We use lightweight sound-event models to identify acoustic events and their temporal spans. Their predictions provide local annotations of sound sources and event occurrence, including events accompanying speech or music. The outputs are collected as evidence for the merge stage, which integrates them into a coherent description of the audio scene.

Music annotation. Specialized lightweight models annotate musical attributes, including instrumentation, rhythm, harmony, and structural cues where available. The resulting annotations describe musical content and its development over time. They are passed to the merge stage together with annotations from the other branches.

2.3 Supervision Assembly for Pretraining and Instruction Tuning

The specialist branches produce annotations with different formats and levels of detail. We normalize transcripts and word alignments, speaker-specific voice descriptions, sound-event annotations,

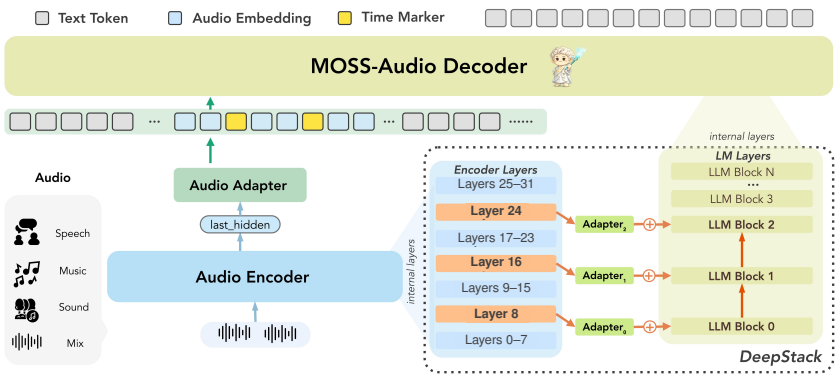

Figure 2: MOSS-Audio architecture. Audio features and time markers condition the decoder; DeepStack supplies intermediate encoder features. Separate adapters project encoder block outputs 8, 16, and 24 into decoder blocks 0, 1, and 2, respectively (all indices zero-based).

and musical attributes into a shared `tool_results` representation. Content priors guide evidence selection and ordering, while quality checks remove empty or repetitive transcripts and reconcile inconsistent speech or music claims. A large language model first plans the description from selected evidence, then merges it into a unified caption retaining supported content, speaker attributes, and temporal relationships. A deterministic fallback follows the same evidence order. Appendix D.6 details the annotation tools and merge procedure.

We organize the annotations into several training targets. Selected transcripts supply ordinary ASR targets, and aligned word and sentence intervals supply timestamped ASR targets. Merged captions describe the overall acoustic scene for pretraining; branch-specific descriptions retain focused speech, sound, and music supervision. Language models further derive question-answer pairs from these descriptions for instruction tuning.

Temporal information is preserved according to the target format. Timestamped transcription explicitly associates text with its boundaries; captions retain available event times and describe order or overlap where supported. This supervises content and temporal relationships without requiring explicit timestamps in every caption.

3 MOSS-Audio

MOSS-Audio uses an encoder–adapter–decoder architecture to generate text y from a recording a and an instruction u. A dedicated audio encoder supplies acoustic features to a Qwen3-4B or Qwen3-8B language backbone; the family names refer to decoder size. The design combines a shared encoder for speech, sound, and music, cross-layer acoustic conditioning, and explicit time tokens (Figure 2). Five training stages produce the Instruct and Thinking variants.

3.1 A Unified Audio Encoder

To capture vocal delivery, environmental events, and music alongside spoken content, we train one encoder from scratch with recognition and description supervision. The approximately 0.6B-parameter encoder takes 16 kHz audio represented as 128-channel log-mel features. Three stride-2 convolutional layers reduce the temporal dimension eightfold, followed by a 32-layer Transformer of width 1280. Its 12.5 Hz output supplies one feature every 80 ms, shortening the decoder's audio sequence. Its depth-ℓ states are $H^{(\ell)} \in \mathbb{R}^{T\times 1280}$.

The encoder processes local audio windows; the decoder integrates the resulting sequence across the recording.

3.2 Acoustic Conditioning Across Encoder Depths

The final encoder layer provides the primary representation of the recording. A GatedMLP adapter maps it to the language-model width:

$$Z = A_{\text{final}}\left(H^{(\text{final})}\right), \qquad Z \in \mathbb{R}^{T \times d_{\text{LM}}}. \tag{1}$$

This sequence forms the audio prefix. Because acoustic and linguistic information vary across encoder depth (Pasad et al., 2021), we also expose intermediate states to the decoder.

We adapt the DeepStack principle (Meng et al., 2024; Bai et al., 2025) to audio by projecting temporally aligned intermediate features into early decoder layers while retaining the final-layer prefix:

$$\mathcal{C} = (C_1, \ldots, C_K), \qquad C_k = A_k\left(H^{(\ell_k)}\right), \tag{2}$$

where $K = 3$, $\ell_k \in \{8, 16, 24\}$ uses zero-based encoder indices, and each A_k is an independent GatedMLP. The three projected sequences are added to the outputs of the first three decoder blocks, respectively, at audio-token positions. Marker and instruction positions receive no direct addition. The pathway adds adapter parameters while preserving temporal correspondence and prefix length; Section 4.4 compares it with final-layer conditioning.

3.3 Learning Time with Explicit Time Tokens

Audio is intrinsically temporal: events have an order, a duration, and positions on a continuous timeline. To help the model learn these relationships, we introduce explicit time tokens into the audio input sequence during training. These tokens associate acoustic features with elapsed time, providing anchors for temporal learning. At 12.5 Hz, we insert a time marker after every 25 features, or two seconds:

$$\mathcal{T}(Z) = \big(z_1, \ldots, z_{25}, M(2), z_{26}, \ldots, z_{50}, M(4), \ldots\big), \tag{3}$$

where $M(\tau)$ embeds tokens representing elapsed time τ in seconds. Word- and sentence-level timestamp targets constructed by the data pipeline (Section 2) provide output-side temporal supervision. Together with the input-side time markers, these targets train the decoder to associate acoustic content with positions and intervals on the audio timeline. Finer boundaries are learned from the timestamp targets; the two-second marker spacing does not set output precision. Appendix D.5 details tokenization, position mapping, and sequence length.

3.4 Multi-Stage Training

Training proceeds through audio–language alignment, joint pretraining, instruction tuning, reasoning cold start, and reinforcement learning. The first stage establishes the connection between acoustic features and the language backbone. Joint pretraining then adapts acoustic representations and text generation together, before instruction tuning teaches the model to respond to different task requests. The last two stages initialize multi-step responses and optimize them with task rewards. Stage 3 produces the *Instruct* models; Stages 4–5 extend them into the *Thinking* variants.

Supervised objective. For Stages 1–4, each audio example contains a recording a, a task instruction u, and a target text sequence y. We minimize

$$\mathcal{L}_{\text{sup}} = -\mathbb{E}_{(a,u,y)\sim\mathcal{D}} \sum_{t=1}^{|y|} \log p_\theta(y_t \mid y_{<t}, a, u), \tag{4}$$

where $\mathcal{D}$ is the training mixture for the current stage. Transcripts, captions, answers, and timestamped text share this autoregressive objective while supplying different supervision. Text-only examples omit audio conditioning. Section 2 describes target construction, and Appendix D lists the architecture and training hyperparameters.

Stage 1: Audio–language alignment. We first train the audio interfaces while holding the encoder and language decoder parameters fixed. The primary adapter learns to map final-layer encoder features into the language-model representation space, while the three cross-layer adapters learn projections from intermediate acoustic states into early decoder layers. Optimizing these connections first provides an audio interface for subsequent joint training.

This stage uses ASR and audio-captioning data. ASR establishes correspondence between acoustic content and words; captioning also requires information about vocal delivery, environmental sounds, and music. Both forms of supervision therefore enter the interface-learning stage. The ASR/caption/text group-weight multipliers are $0.3/0.4/0$, with text-only training disabled. Both model recipes allocate 10,000 updates at a configured learning rate of 10^{-3}. Encoder and decoder parameter groups use zero learning-rate multipliers; only the interface parameters are updated.

Stage 2: Joint pretraining. After alignment, we update the encoder, all adapters, and the decoder together. This allows acoustic feature extraction, cross-modal projection, and text prediction to adapt jointly to three complementary task groups. *ASR-related tasks* include ordinary transcription, word-level timestamped transcription, and sentence-level timestamped transcription, including long-audio examples. *Audio captioning* uses targets from the data pipeline to describe spoken content, vocal delivery, sound events, and music. *Text-only language modeling* uses mathematics, code, educational, literary, and general text to continue training language capabilities alongside audio understanding.

The ASR/caption/text group weights are $0.3/0.4/0.3$. Within each group, dataset weights are proportional to the square root of dataset size. This weighting gives larger datasets more exposure while reducing their dominance relative to sampling directly in proportion to size. The configured joint-training learning rates are 2×10^{-5} for 4B and 10^{-5} for 8B. The two pretraining stages account for approximately 1.2T tokens of reported exposure, including repeated examples; configured step budgets and continuation schedules appear in Appendix D.3.

Temporal learning couples the input representation to the target supervision. As described in Section 3.3, explicit time tokens are interleaved with the acoustic features every two seconds, marking elapsed times such as 2, 4, and 6 seconds. These tokens provide time references within the input sequence. Word- and sentence-level timestamp targets then require the decoder to associate recognized content with specific boundaries on that timeline. The input anchors and output supervision jointly train content–time correspondence; the marker interval does not quantize the predicted boundaries.

Stage 3: Instruction tuning. Supervised fine-tuning adapts the pretrained model to task instructions and response formats. The mixture contains audio question answering, overall and specialist captioning, ordinary and timestamped ASR, and self-identity examples. Question–answer pairs are constructed from speech, sound, and music descriptions, covering spoken content, speaker attributes, acoustic events, musical characteristics, and temporal relations.

Merged descriptions teach the model to integrate information across sources. Retained branch-specific descriptions supply focused targets, such as vocal delivery or musical content, so that the requested scope can vary with the instruction. Transcription and timestamp tasks continue to provide their corresponding output formats during this adaptation. This stage yields MOSS-Audio-Instruct for direct transcription, description, and question answering.

Stage 4: Reasoning cold start. Starting from the instruction-tuned model, we use audio-centered and text-only reasoning examples to initialize multi-step response patterns. Audio examples train the model to organize spoken content, prosodic and emotional cues, sound events, temporal relations, and musical structure into a reasoning process leading to an answer. Text-only examples provide general reasoning and answer-organization patterns.

Cold start retains the supervised objective in Equation 4. It extends task-format adaptation with examples of combining multiple cues before producing a final answer, providing the initial policy for online reward optimization.

Stage 5: Reinforcement learning. After cold start, we apply DAPO-based reinforcement learning (Yu et al., 2025). For each prompt, the current policy samples multiple candidate responses that are scored using task rewards. Differences among candidates for the same prompt provide the signal for updating the policy toward higher-reward responses.

We sample 16 responses per prompt with temperature 1.0, top-p 1.0, top-k 50, and a maximum generation length of 2,048 tokens. Dynamic filtering removes prompt groups with identical rewards because they provide no within-group discrimination signal. Policy updates use asymmetric clipping

Table 1: Audio understanding comparison among open models; higher is better. Bold marks the best score in each column.

Model	Size	MMAU	MMAU-Pro	MMAR	MMSU	Avg.
Qwen3-Omni-30B-A3B-Instruct	30B	75.00	61.22	66.40	69.00	67.91
Step-Audio-R1.1	33B	72.18	60.80	68.75	64.18	66.48
Step-Audio-R1	33B	**78.67**	59.68	**69.15**	75.18	70.67
Kimi-Audio	7B	72.41	56.58	60.82	54.74	61.14
Qwen2.5-Omni	7B	65.60	52.20	56.70	61.32	58.96
Audio Flamingo 3‡	7B	61.23	51.70	57.96	60.04	57.73
Audio Flamingo Next	8B	76.10	56.34	51.01	57.20	60.16
MiMo-Audio-7B	7B	74.90	53.35	61.70	61.94	62.97
MiniCPM-o-4.5	9B	70.97	39.65	55.75	60.96	56.83
MOSS-Audio-4B-Instruct	4B	75.79	58.16	62.54	59.68	64.04
MOSS-Audio-4B-Thinking	4B	77.64	60.75	63.91	71.20	68.37
MOSS-Audio-8B-Instruct	8B	77.03	57.48	64.42	66.36	66.32
MOSS-Audio-8B-Thinking	8B	77.33	**64.92**	66.53	**75.52**	**71.08**

coefficients of 0.2 and 0.28 and token-level importance-sampling correction; Appendix D.4 lists the remaining recorded rollout settings.

Through reasoning cold start and reinforcement learning, we obtain MOSS-Audio-Thinking for multi-step audio understanding.

4 Experiments

We test whether MOSS-Audio combines semantic understanding, acoustic detail, transcription, and temporal localization within one model. We then examine our encoder and cross-layer conditioning through controlled experiments.

Experimental setup. We use our 4B/8B Instruct models for all four tasks and the corresponding Thinking models for audio understanding. Understanding uses public benchmarks, captioning and ASR use internal suites, and timestamping uses public audio with human gold-standard labels. All models share task-specific scoring rules; scores average repeated evaluations (Appendix C). System comparisons report task performance; ablations control non-target settings.

4.1 General Audio Understanding

To test unified semantic understanding across speech, sound, and music, we use MMAU (Sakshi et al., 2025), MMAU-Pro (Kumar et al., 2025), MMAR (Ma et al., 2025), and MMSU (Wang et al., 2025). Table 1 gives individual scores and their equal-weight benchmark means.

Strong understanding across audio types. Our 8B-Thinking model achieves the highest four-benchmark mean among the compared open models, reaching 71.08 versus 70.67 for Step-Audio-R1. Each model leads on two benchmarks, demonstrating competitive unified understanding. Our mean advantage over Step-Audio-R1 depends on benchmark composition (Appendix **??**).

Reasoning-oriented post-training improves understanding. Our Thinking variants outperform their Instruct counterparts on all four benchmarks at both model sizes. The four-benchmark average increases from 64.04 to 68.37 for 4B and from 66.32 to 71.08 for 8B, showing consistent improvements in audio understanding across model scales.

4.2 Describing Acoustic Detail Beyond Words

We use speech captioning to test whether our model captures accent, voice texture, and delivery beyond spoken words. We select 2,000 clips by balanced sampling over model-proposed speaker tags and obtain human-written, reviewed descriptions for 13 dimensions. All models produce the same 13-field output, which a text judge compares with the references under shared scoring rules.

Table 2: Speech-caption reference matching; higher is better. Macro-13 and Macro-10 are equal-weight means of 13 and 10 dimensions, with three exploratory personal-attribute fields excluded from Macro-10. Bold marks column maxima. All dimensions and original aggregates appear in Table 6.

Model	Accent	Texture	Clarity	Emotion	Speed	Macro-13	Macro-10
Audio-Flamingo-Next	3.160	1.941	2.839	2.056	2.818	2.682	2.485
Qwen3-Omni-Instruct	4.356	3.093	3.521	**3.328**	3.614	3.599	3.512
Qwen3-Omni-Thinking	4.327	3.179	3.403	3.232	3.610	3.567	3.473
Gemini-3-Pro	4.181	2.998	3.347	3.055	3.320	3.376	3.284
Gemini-3.1-Pro	4.310	3.134	3.559	3.231	**3.687**	3.577	3.497
MOSS-Audio-4B-Instruct	4.497	**3.407**	3.841	3.311	3.564	3.711	3.626
MOSS-Audio-8B-Instruct	**4.572**	3.403	**3.869**	3.314	3.638	**3.725**	**3.649**

Table 3: Timestamp accuracy against human gold-standard labels; lower AAS (ms) is better. "–" denotes an unreported result. Bold marks the lowest reported value per column; no cross-language average is computed.

Model	AISHELL-1 (zh)	LibriSpeech (en)
Audio-Flamingo-Next	–	211.15
Qwen3-Omni-Instruct	833.66	646.95
Gemini-3.1-Pro	708.24	871.19
MOSS-Audio-4B-Instruct	76.96	358.13
MOSS-Audio-8B-Instruct	**35.77**	**131.61**

Table 2 shows selected dimensions and equal-weight summaries. Macro-10 excludes Gender, Age, and Personality to test whether the advantage depends on these exploratory attributes.

MOSS-Audio captures fine-grained vocal characteristics. Both Instruct variants outperform the compared baselines on Accent, Texture, and Clarity and achieve higher Macro-13 scores. The advantage persists on Macro-10: our 4B and 8B models reach 3.626 and 3.649, respectively, versus the strongest baseline's 3.512. These results support detailed acoustic description under human-reference matching. Qwen3-Omni-Instruct still leads on Emotion and Gemini-3.1-Pro on Speed.

4.3 Timestamp Accuracy

To test localization of spoken content, we compare generated timestamps with human gold-standard labels on the official AISHELL-1 and LibriSpeech test sets. We use Accumulated Average Shift (AAS) (Shi et al., 2026), $\mathrm{AAS} = N^{-1} \sum_{i=1}^{N} |\hat{t}_i - t_i|$, the mean absolute timestamp-slot error in milliseconds, where N counts the scored slots. All models use the same matching and scoring rules; we report the two languages separately.

Accurate temporal localization in Chinese and English. Our 8B-Instruct model achieves the lowest AAS among the compared models on both datasets: 35.77 ms on AISHELL-1 and 131.61 ms on LibriSpeech, compared with 76.96 ms and 358.13 ms for our 4B model (Table 3). These results support temporal localization in both languages; isolating the contribution of two-second markers requires a separate ablation.

Across tasks, 8B improves the understanding and captioning summaries, reduces timestamp error, and lowers Average CER relative to 4B (Appendix **??**). The condition-level results show where these aggregate improvements are concentrated.

4.4 Ablation Study

We test whether our dedicated encoder improves transcription and whether intermediate features help the decoder describe audio.

Our audio encoder improves transcription. We first compare our encoder with Whisper-large-v3 and AuT Xu et al. (2025) using XARES-LLM (Dinkel et al., 2026) (Table 4). Our encoder leads

Table 4: Encoder comparison on XARES-LLM Task 2, which evaluates generative capabilities including ASR ($1-\mathrm{WER}/\mathrm{CER}$) and audio captioning (FENSE/DATE). Best results are highlighted in bold. AVG_2 averages the five task scores.

Encoder	Task 2: Speech & Generation Benchmark					
	AISHELL	Clotho	LibriSpeech	MECAT	SongDesc	AVG_2
Whisper-large-v3 (Radford et al., 2023)	0.467	0.402	0.458	0.652	0.480	0.492
AuT (Xu et al., 2025)	0.849	0.399	0.899	0.654	**0.505**	0.661
MOSS Audio Encoder	**0.888**	**0.411**	**0.918**	**0.657**	0.491	**0.673**

Table 5: DeepStack ablation with Qwen3-0.6B-base. DATE is higher-is-better on the original scale. Δ is DeepStack minus final layer.

Interface	Pure music	Pure sound	Pure speech	Mixed speech	Overall
Final layer	0.4705	0.4015	0.4861	0.4922	0.4823
DeepStack	0.4927	0.4255	0.4791	0.4838	0.4831
Δ	$+0.0222$	$+0.0240$	-0.0070	-0.0084	$+0.0008$

on both ASR benchmarks, Clotho, and MECAT, with the highest generation average (0.673 versus AuT's 0.661), primarily from ASR gains.

For the controlled comparison, we pair MOSS and AuT with the same fixed Qwen3-1.7B decoder and direct interface, without cross-layer injection. We match training data, the 100k-step budget, optimization, decoding, and scoring. Our encoder lowers error on 36 of 38 test conditions, reducing the equal-weight CER/WER mean from 17.61 to 16.31 (Table **??**). Grouping related conditions preserves the advantage (24/26 wins; 16.33 to 15.15), including when AISHELL labels receive one group weight (Appendix **??**). This comparison evaluates pretrained encoder choice across test conditions. The means weight the test-set CER/WER scores equally.

Cross-layer features improve sound and music captioning. To test whether intermediate acoustic features add useful information beyond the final-layer prefix, we compare final-layer conditioning with DeepStack. Both runs start from the same MOSS encoder and Qwen3-0.6B-base weights, with matched ASR alignment, MECAT-Caption fine-tuning, preprocessing, and scoring (Appendix **??**). Only the conditioning interface changes. We use DATE, which combines caption semantic similarity and cross-sample discriminability (Niu et al., 2025).

DeepStack raises Pure Sound DATE from 0.4015 to 0.4255 (about 6.0% relative) and Pure Music from 0.4705 to 0.4927 (about 4.7%) (Table 5). This supports intermediate features for non-speech description in the tested 0.6B setting. Both speech scores decrease, leaving Overall nearly unchanged ($+0.0008$). The aggregate thus obscures the benefit for pure sound and music.

5 Related Work

Detailed related works are provided in Appendix A.

6 Conclusion

We presented MOSS-Audio, a family of audio-language models for unified understanding of speech, environmental sound, and music. Our approach combines event-oriented data construction with a dedicated audio encoder, cross-layer acoustic conditioning, and explicit time tokens to capture semantic content, acoustic detail, and temporal structure. Multi-stage training produces Instruct and Thinking variants at 4B and 8B scales. Experiments demonstrate strong performance in audio understanding, fine-grained speech captioning, transcription, and timestamp prediction, with MOSS-Audio-8B-Thinking achieving the highest average score across four understanding benchmarks among the compared open models. We will release model weights, training data, code, and complete training recipes to support further research in unified audio understanding.

Ethics Statement

MOSS-Audio is developed for research on general-purpose audio understanding. Before releasing data, we will screen audio recordings, metadata, and derived annotations for personally identifiable and sensitive information, including names, contact details, and precise addresses. Identified information will be redacted, and samples that cannot be adequately de-identified will be excluded from the release. These measures aim to reduce privacy risks, including unintended disclosure of personal information. Data redistribution will follow source-specific permissions and usage restrictions. Speaker descriptions represent perceived vocal characteristics rather than verified personal attributes and should not be used for high-stakes judgments about individuals.

Reproducibility Statement

We will release model weights, training data, inference and training code, and the full recipe for data processing, pretraining, and post-training. Appendices C–D describe the architecture, training, evaluation procedures, and component comparisons. Reported scores average repeated evaluations under the same protocol, and each ablation holds non-target settings fixed. Caption judging uses dimension-wise matching against reviewed human descriptions (Appendix C). Timestamp accuracy is evaluated against human gold-standard labels.

AI Use Statement

In preparing this work, we used generative AI tools primarily to summarize and analyze existing literature and to edit the manuscript for clarity and readability. The authors are responsible for verifying AI-assisted summaries against the original sources and ensuring that edits preserve the intended scientific meaning. Model-generated training annotations, captions, and question–answer pairs are described in Section~**??**sec:data. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

References

Keyu An, Yanni Chen, Zhigao Chen, Chong Deng, Zhihao Du, Changfeng Gao, et al. Fun-asr technical report, 2025. URL https://arxiv.org/abs/2509.12508.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José Zapata, and Xavier Serra. Essentia: an open-source library for sound and music analysis. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pp. 855–858, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324045. doi: 10.1145/2502081.2502229. URL https://doi.org/10.1145/2502081.2502229.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, et al. BEATs: Audio pre-training with acoustic tokenizers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5178–5193. PMLR, 2023.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Heinrich Dinkel, Jiahao Zhou, Guanbo Wang, Yadong Niu, Junbo Zhang, Yufeng Hao, et al. The interspeech 2026 audio encoder capability challenge for large audio language models, 2026. URL https://arxiv.org/abs/2603.22728.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, et al. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Sreyan Ghosh, Sonal Kumar, Ashish Seth, Chandra Kiran Reddy Evuru, Utkarsh Tyagi, S. Sakshi, et al. GAMA: A large audio-language model with advanced audio understanding and complex reasoning abilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024.

Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Mojtaba Heydari, Frank Cwitkowitz, and Zhiyao Duan. Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431. Association for Computational Linguistics, April 2017.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 119–132. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1011.

Apoorva Kulkarni, Kaousheik Jayakumar, Sreyan Ghosh, Utathya Aich, Ramani Duraiswami, and Dinesh Manocha. TEMPO: Temporally-grounded multi-task post-training for large audio-language models, 2026. URL https://arxiv.org/abs/2608.29999v1.

Sonal Kumar, Šimon Sedláček, Vaibhavi Lokegaonkar, Fernando López, Wenyi Yu, Nishit Anand, Hyeonggon Ryu, Lichang Chen, Maxim Plička, Miroslav Hlaváček, et al. MMAU-Pro: A challenging and comprehensive benchmark for holistic evaluation of audio general intelligence. *arXiv preprint arXiv:2508.13992*, 2025.

Sonal Kumar, Prem Seetharaman, Ke Chen, Oriol Nieto, Jiaqi Su, Zhepei Wang, et al. Tac: Timestamped audio captioning, 2026. URL https://arxiv.org/abs/2602.15766.

Ziyang Ma, Yinghao Ma, Yanqiao Zhu, Chen Yang, Yi-Wen Chao, Ruiyang Xu, et al. MMAR: A challenging benchmark for deep reasoning in speech, audio, music, and their mix. *arXiv preprint arXiv:2505.13032*, 2025.

Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 135–140, Utrecht, The Netherlands, 2010.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, et al. WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024. doi: 10.1109/TASLP.2024.3419446.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. DeepStack: Deeply stacking visual tokens is surprisingly simple and effective for LMMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/29cd7f8331d13ede6dc6d6ef3dfacb70-Abstract-Conference.html.

Yadong Niu, Tianzi Wang, Heinrich Dinkel, Xingwei Sun, Jiahao Zhou, Gang Li, et al. Mecat: A multi-experts constructed benchmark for fine-grained audio understanding tasks. *arXiv preprint arXiv:2507.23511*, 2025.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pp. 914–921. IEEE, 2021. doi: 10.1109/ASRU51503.2021.9688093. URL https://doi.org/10.1109/ASRU51503.2021.9688093.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, et al. Scaling speech technology to 1,000+ languages. *arXiv*, 2023.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, et al. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97): 1–52, 2024. URL http://jmlr.org/papers/v25/23-1318.html.

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

S. Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, et al. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025.

Florian Schmid, Tobias Morocutti, Francesco Foscarin, Jan Schlüter, Paul Primus, and Gerhard Widmer. Effective pre-training of audio transformers for sound event detection, 2024. URL https://arxiv.org/abs/2409.09546.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, et al. SALMONN: Towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, et al. MMSU: A massive multi-task spoken language understanding and reasoning benchmark. *arXiv preprint arXiv:2506.04779*, 2025.

Hualei Wang, Yiming Li, Shuo Ma, Hong Liu, and Xiangdong Wang. Listening between the frames: Bridging temporal gaps in large audio-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26233–26241, 2026. doi: 10.1609/aaai.v40i31.39827. URL https://ojs.aaai.org/index.php/AAAI/article/view/39827.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, et al. Qwen3-omni technical report, 2025. URL https://arxiv.org/abs/2509.17765.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*, 2021.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

A Related Work

Audio-Language Models and Acoustic Representations. Large audio-language models, including SALMONN, Qwen2-Audio, and the Audio Flamingo series, connect audio encoders to language models to support transcription, captioning, and question answering (Tang et al., 2024; Chu et al., 2024; Ghosh et al., 2025). Beyond final-layer representations, GAMA aggregates features from multiple encoder layers to enrich the acoustic information supplied to the language model (Ghosh et al., 2024). However, this aggregation occurs at the input: intermediate encoder states are combined before decoding, without separate pathways that expose them directly to internal decoder layers. This leaves the placement of acoustic information within the decoder as an additional design consideration. Inspired by the visual feature injection strategies of DeepStack and Qwen3-VL (Meng et al., 2024; Bai et al., 2025), we introduce cross-layer acoustic conditioning in MOSS-Audio. Temporally aligned intermediate audio representations are projected into early decoder layers alongside the final-layer audio prefix, providing direct access to multiple encoder depths without increasing the prefix length.

Audio Supervision and Event-Oriented Data Construction. Audio captioning datasets extend supervision beyond transcription by describing audible events and scene context. AudioCaps provides human-written descriptions (Kim et al., 2019), while WavCaps scales caption supervision through filtering and language-model-assisted refinement (Mei et al., 2024). Audio Flamingo 3's Speech-in-Sound-Caps further combines spoken content, vocal delivery, and surrounding sounds (Ghosh et al., 2025). These efforts enrich the descriptions associated with audio clips, but richer annotations alone cannot recover event context removed during segmentation. In particular, fixed-duration or speech-based cuts may divide a sound that begins before an utterance and continues after it. We therefore organize data construction around detected acoustic events, using their temporal spans to guide clip boundaries and their categories to select complementary speech, sound, and music annotators. The resulting annotations are fused into chronological captions for pretraining and retained separately for instruction tuning, linking the formation of training clips to both holistic scene descriptions and detailed acoustic supervision.

Temporal Grounding in Audio Understanding. Clip-level captions and question–answer pairs supervise what a recording contains, but do not necessarily associate that content with explicit time intervals. Consequently, learning to describe events or their order does not by itself specify how audio-feature positions map to elapsed time. TimeAudio addresses temporal localization through temporal tokens and absolute time encoding (Wang et al., 2026), while TEMPO combines timestamp tokens, a time-aware projector, and multitask temporal supervision (Kulkarni et al., 2026). These approaches motivate explicit temporal information at the audio–language interface. In MOSS-Audio, we interleave numerical time markers with the audio features and provide word- and sentence-level timestamp supervision. The markers establish a reference timeline, while the supervision teaches the decoder to associate textual units with their temporal boundaries, supporting ordinary text generation and timestamped transcription within the same autoregressive interface.

B Detailed Experimental Results

These tables retain every original captioning and ASR score and aggregate; Tables 1 and 3 give the understanding and timestamp results. "Recomputed" denotes an equal-weight average of displayed scores using decimal round-half-up. Missing entries are marked with a dash.

B.1 Speech captioning: all dimensions

In Table 6, AFN denotes Audio-Flamingo-Next; Q3-I and Q3-T denote Qwen3-Omni-Instruct and Qwen3-Omni-Thinking; G3 and G3.1 denote Gemini-3-Pro and Gemini-3.1-Pro; M4-I and M8-I denote MOSS-Audio-4B-Instruct and MOSS-Audio-8B-Instruct. Bold identifies row-wise maxima except for the three flagged exploratory dimensions. Values use the original judge-score scale.

Gender, Age, and Personality measure agreement with annotators' reference descriptions. Macro-13 weights all thirteen dimensions equally; Macro-10 averages the ten dimensions after excluding these three fields.

Table 6: Complete internal speech-caption reference-matching scores; higher means closer agreement. ⋆Exploratory speaker-description dimensions excluded from Macro-10; bold highlights maxima in the other ten dimensions.

Dimension	AFN	Q3-I	Q3-T	G3	G3.1	M4-I	M8-I
Gender⋆	4.617	4.436	4.419	4.191	4.347	4.697	4.683
Age⋆	3.461	3.936	4.026	3.835	4.030	3.980	3.979
Accent	3.160	4.356	4.327	4.181	4.310	4.497	**4.572**
Pitch	2.679	3.590	3.610	3.392	3.474	3.628	**3.682**
Volume	2.391	3.682	3.577	3.254	3.580	**3.722**	3.709
Speed	2.818	3.614	3.610	3.320	**3.687**	3.564	3.638
Texture	1.941	3.093	3.179	2.998	3.134	**3.407**	3.403
Clarity	2.839	3.521	3.403	3.347	3.559	3.841	**3.869**
Fluency	2.788	3.531	3.526	3.524	3.720	3.744	**3.747**
Emotion	2.056	**3.328**	3.232	3.055	3.231	3.311	3.314
Tone	2.025	3.224	3.154	2.997	3.245	**3.282**	3.253
Personality⋆	1.940	3.292	3.197	3.023	3.158	3.305	3.272
Summary	2.157	3.179	3.107	2.775	3.028	3.259	**3.307**
Reported average	2.683	3.599	3.567	3.376	3.577	3.711	**3.725**
Recomputed mean	2.682	3.599	3.567	3.376	3.577	3.711	**3.725**
Recomputed Macro-10	2.485	3.512	3.473	3.284	3.497	3.626	**3.649**

The benchmark uses 2,000 clips selected by balanced sampling over model-proposed speaker tags (Section 4.2). Human references are reviewed for consistency with the audio, and all models receive the same output schema and judging rules. The final score averages matching scores over the evaluated samples and dimensions. Reported values average repeated evaluations with the same judging procedure. Table 9 summarizes the evaluation targets and aggregation.

Macro-10 is a sensitivity calculation on the displayed scores: remove Gender, Age, and Personality and average the other ten dimensions equally, using the same decimal rounding. The ordering is unchanged for the two MOSS-Audio variants, and both remain above the five listed baselines under this exclusion.

B.2 ASR: ALL CATEGORIES

In Table 7, PF-L denotes Paraformer-Large; GLM denotes GLM-ASR-Nano; FN denotes Fun-ASR-Nano; SV-S denotes SenseVoice-Small; Kimi denotes Kimi-Audio-7B-Instruct; AFN denotes Audio-Flamingo-Next; Q2-3 and Q2-7 denote Qwen2.5-Omni-3B and Qwen2.5-Omni-7B; Q3 denotes Qwen3-Omni-Instruct; M4-I and M8-I denote the two MOSS-Audio Instruct variants. Bold identifies a separate minimum for each row.

We retain the original Average for the main ASR comparison and compute Macro-12 as a supplementary aggregation. Macro-12 averages the twelve displayed category CERs equally: it gives 11.02 for 8B-Instruct, 10.84 for 4B-Instruct, and 10.62 for Qwen3-Omni-Instruct. These values differ from their original Average scores of 11.30, 11.58, and 11.39, respectively. Average and Macro-12 are labeled separately throughout.

Observed condition trade-offs. Relative to 4B-Instruct, 8B-Instruct has lower reported CER in seven of the twelve categories and higher CER in five. The largest increases are $+5.56$ on far-/near-field and $+4.03$ on multi-speaker speech; the largest decreases are -3.08 on dialect and -1.93 on health condition. These contrasts characterize the two models across acoustic conditions.

Macro-12 gives each category equal weight. Average is the original summary used in the main comparison. Both retain the original numerical scale.

Table 7: Internal ASR results with a supplementary aggregation analysis; lower CER is better. Average matches the main table. Recomputed Macro-12 weights the twelve categories equally and is shown separately.

Category	PF-L	GLM	FN	SV-S	Kimi	AFN	Q2-3	Q2-7	Q3	M4-I	M8-I
Health condition	22.18	24.49	21.99	24.04	21.11	36.13	24.65	23.85	20.73	21.11	**19.18**
Dialect	43.45	22.39	**7.80**	8.89	29.34	25.85	33.87	31.91	15.63	11.84	8.76
Singing	32.34	51.95	19.35	23.79	21.76	25.80	24.24	22.69	16.01	10.79	**9.81**
Non-speech vocalizations	4.95	4.65	4.76	4.92	4.68	8.25	5.54	4.56	4.73	**4.01**	4.31
Code-switching	12.65	11.88	11.23	13.90	16.38	34.53	11.66	12.97	11.30	**10.11**	10.18
Acoustic environment (clean)	3.11	3.68	2.98	4.13	**2.20**	8.64	2.76	2.52	2.23	3.11	2.70
Acoustic environment (noisy)	4.67	5.02	3.46	4.93	**2.15**	38.85	3.56	3.16	2.47	3.72	3.20
Whisper	5.02	4.94	3.78	5.57	2.66	12.84	4.32	3.64	**1.90**	3.29	2.75
Far-field / near-field	17.46	27.51	18.38	26.66	21.02	42.90	22.15	25.38	**17.08**	18.48	24.04
Multi-speaker	20.33	28.02	19.82	24.06	20.61	62.13	22.91	21.01	**18.15**	20.33	24.36
Age	14.96	17.19	14.95	17.63	16.74	38.30	15.17	16.13	**11.46**	15.09	15.26
Semantic content	7.14	7.32	6.08	7.55	6.12	30.18	7.24	6.78	**5.74**	8.15	7.69
Average	15.77	17.29	12.04	14.50	14.12	30.19	15.26	15.05	11.39	11.58	**11.30**
Recomputed Macro-12	15.69	17.42	11.22	13.84	13.73	30.37	14.84	14.55	**10.62**	10.84	11.02

B.3 Timestamp references and scoring

Table 3 reports AAS on official AISHELL-1 and LibriSpeech test audio against human gold-standard timestamp labels. All models use the same matching and scoring rules; AAS averages the absolute differences at evaluated timestamp slots and is reported in milliseconds. Language-specific results are reported separately, with a dash for Audio-Flamingo-Next's missing AISHELL-1 result.

Reported AAS values average repeated evaluations with the same human references and matching procedure.

C Experimental Protocols

All models within a task use the same scoring rules. The reported scores are means over repeated evaluations. The component comparisons hold the non-target settings fixed, and the model-family comparisons assess the complete Instruct and Thinking models.

The captioning and timestamp results concern the Instruct models. Instruct and Thinking are separate post-training variants; their comparison includes both reasoning cold start and RL.

C.1 Task instructions and output formats

Table 8 summarizes the task instructions and required output formats used in our evaluation. Each model receives the corresponding instruction and audio input.

Table 8: Task-specific output formats used in the experiments.

Task	Requested response
ASR	Transcription only, without headings or explanatory text.
Speech captioning	JSON descriptions of gender, age, pitch, speed, volume, clarity, fluency, accent, texture, emotion, tone, personality, and summary.
Timestamped ASR	Consecutive `[start]text[end]` segments at sentence or word level, without extra commentary.
Open-ended QA	An answer grounded in the audio and the supplied question.
Multiple-choice QA	One option letter from A–D, without an explanation.

C.2 Scoring and aggregation

Speech-caption evaluation compares each generated field with its human reference and averages matching scores across samples and dimensions. Macro-10 excludes Gender, Age, and Personality.

Table 9: Experimental tasks and aggregation. Scores are averaged over repeated evaluations under the same task protocol.

Task	Evaluation target	Metric and aggregation
Understanding	Questions about speech, sound, and music	Benchmark scores; equal-weight mean over four benchmarks
Speech captioning	Agreement with reviewed human descriptions of 2,000 clips	Text-judge matching across 13 dimensions; Macro-13 and the complementary Macro-10
ASR	Transcription under diverse acoustic conditions	CER by category and original Average; supplementary Macro-12 in Appendix B.2
Timestamped ASR	Human gold-standard timestamp labels on AISHELL-1 and LibriSpeech	Mean absolute timestamp shift in milliseconds; languages reported separately

These fields describe agreement with annotators' reference descriptions. Automatic judging and human reference annotation have different roles: the former compares text, while the latter establishes the audio-grounded descriptions.

Timestamp evaluation uses human gold-standard reference times for the official AISHELL-1 and LibriSpeech test audio. Predicted and reference slots are matched using the same rules across models. AAS is their average absolute time difference in milliseconds. The human evaluation labels are distinct from the automatic alignment annotations used for training.

C.3 Component comparisons

The encoder experiment shares the Qwen3-1.7B decoder, direct audio interface, training data, 100k-step budget, optimization, decoding, and scoring. The DeepStack experiment shares the initial MOSS encoder and Qwen3-0.6B-base decoder, two-stage training, preprocessing, test inputs, prompts, and scoring; only cross-layer conditioning differs. Appendix **??** gives the complete results and their content-level analysis. All reported values average repeated evaluations.

D Architecture and Training Hyperparameters

This appendix specifies the model interface and training configuration. Tables 10–13 describe the 512-GPU 4B and 8B pretraining configurations. Schedule endpoints specify the configured training budgets. Post-training settings are listed separately in Appendix D.4.

D.1 Model configuration

Each GatedMLP maps 1,280-dimensional encoder states to the corresponding decoder hidden width. The decoder hidden width and the total attention projection width are separate configuration fields: the 4B configuration uses hidden width 2,560 and 32×128 attention dimensions.

D.2 Training stages and parameter updates

Stages 1 and 2 correspond to the *Adapter* and *Full* phases in the training configuration. In Stage 1, encoder and decoder parameter groups use `lr_mult=0`; the remaining interface parameters are updated. Stage 2 updates the full model.

D.3 Pretraining hyperparameters

Both recipes have a nominal capacity of 8,388,608 sequence positions per update: $8{,}192 \times 1{,}024$ for 4B and $16{,}384 \times 512$ for 8B. This capacity includes both conditioning and text positions. The approximately 1.2T-token pretraining exposure reported in the paper includes repeated examples.

Table 10: Model and front-end parameters in the 4B and 8B training recipes. Encoder and decoder layer indices are zero-based in the configuration; the cross-layer pathway injects into the first three decoder blocks.

Parameter	4B	8B
Language backbone	Qwen3-4B	Qwen3-8B
Decoder layers	36	36
Decoder hidden width	2,560	4,096
Decoder feed-forward width	9,728	12,288
Attention heads / KV heads	32 / 8	32 / 8
Attention head dimension	128	128
Vocabulary size	151,936	151,936
RoPE base	1,000,000	1,000,000
Audio encoder parameters	≈0.6B	≈0.6B
Encoder layers / hidden width	32 / 1,280	32 / 1,280
Encoder heads / feed-forward width	20 / 5,120	20 / 5,120
Audio sampling rate	16 kHz	16 kHz
Mel bins / FFT size / hop (samples)	128 / 400 / 160	128 / 400 / 160
Convolutional temporal strides	2, 2, 2	2, 2, 2
Encoded feature rate	12.5 Hz	12.5 Hz
Primary / cross-layer adapters	1 / 3	1 / 3
Adapter intermediate width	8,192	8,192
Cross-layer encoder indices	8, 16, 24	8, 16, 24
Time-marker interval	25 features / 2 s	25 features / 2 s

Table 11: Training stages and resulting model variants.

Stage	Objective and data	Output
1: Alignment	ASR and captioning; update the primary and cross-layer interfaces with encoder/decoder learning-rate multipliers set to zero	Aligned audio–language model
2: Joint pretraining	Joint optimization of encoder, adapters, and decoder on ASR, captioning, and text	Pretrained model
3: Instruction tuning	Audio QA, overall and specialist captioning, ordinary and timestamped ASR, and self-identity examples	Instruct model
4: Reasoning cold start	Audio-centered and text-only reasoning examples	Reasoning initialization
5: Reinforcement learning	DAPO-based task reward optimization	Thinking model

Table 12: Shared optimization and runtime settings for Stages 1–2 of the 512-GPU recipes. Batch sizes count sequences per optimizer update.

Hyperparameter	4B	8B
Optimizer	Adam	Adam
Adam β_1, β_2, ϵ	$0.9, 0.95, 10^{-8}$	$0.9, 0.95, 10^{-8}$
Weight decay	0.01	0.01
Gradient clipping threshold	1.0	1.0
Training precision	BF16	BF16
Sequence length (tokens)	8,192	16,384
Global batch size (sequences)	1,024	512
Micro-batch size per GPU	1	1
Gradient accumulation steps	2	1
Nodes × GPUs per node	64×8	64×8
Data / tensor / pipeline / context parallelism	512 / 1 / 1 / 1	512 / 1 / 1 / 1
Learning-rate schedule	WSD, cosine decay	WSD, cosine decay
Activation recomputation	Full, block method	Full, block method
Recomputed layers	18	36
DataLoader workers per rank	32	4
Random seed	3407	3407

Table 13: Configured pretraining schedules. Iteration ranges use the global training counter; the 8B continuation is a branch from iteration 80,000. All warmup and decay entries are scheduler arguments in iterations, and endpoints are planned training targets.

Recipe / stage	Global range	LR	Min. LR	Warmup	Decay	WSD decay
4B / 1	0–10k	10^{-3}	10^{-4}	300	10k	500
4B / 2	10k–990k	2×10^{-5}	2×10^{-6}	3k	980k	98k
8B initial / 1	0–10k	10^{-3}	10^{-4}	200	10k	200
8B initial / 2	10k–100k	10^{-5}	10^{-6}	3k	90k	10k
8B resumed / 2	80k–1,000k	10^{-5}	10^{-6}	3k	1,000k	100k

Here LR and Min. LR are the configured learning rate and minimum rate. WSD denotes warmup–stable–decay with a cosine decay segment. The *Decay* and *WSD decay* columns record `lr_decay_iters` and `lr_wsd_decay_iters`, respectively. The 4B schedule allocates 10,000 alignment updates and 980,000 joint updates. The initial 8B schedule allocates 10,000 and 90,000 updates. The 8B continuation inherits 10,000 alignment and 70,000 joint updates from its source checkpoint, with 920,000 additional joint updates configured. It retains the global iteration counter and optimizer/scheduler state with scheduler override enabled. The 3,000-step warmup argument belongs to the global schedule. The resumed branch includes the first 80,000 updates inherited from its source checkpoint.

Data mixture and run control. Stage 1 sets the ASR/caption/text group-weight multipliers to $0.3/0.4/0$; Stage 2 uses $0.3/0.4/0.3$. Within-group weights are proportional to the square root of dataset size and are read from cached shard weights after exclusion filtering. The Stage-1 multipliers are unnormalized group weights. ASR includes ordinary transcription, word- and sentence-level timestamp targets, and long-audio examples. Captioning uses the merged and captioner-generated targets; text-only data enters at Stage 2. Both recipes configure a train/validation/test split ratio of $998:1:1$, validation every 1,000 iterations for 10 iterations, checkpoints every 10,000 iterations, snapshots every 1,000 iterations, and logs every five iterations.

D.4 Instruction and reasoning post-training

Stage 3 produces Instruct models from audio QA, captioning, ordinary and timestamped ASR, and self-identity examples. Stage 4 initializes multi-step reasoning with audio-centered and text-only examples, and Stage 5 applies DAPO to obtain Thinking models. The comparison between Thinking and Instruct includes both reasoning stages.

Table 14: Stage-5 rollout and DAPO settings. These parameters describe reasoning post-training, independently of the pretraining recipe.

Hyperparameter	Setting
Reasoning rollout sampling	Temperature 1.0; top-p 1.0; top-k 50
Responses per prompt	16
Rollout batch size	128
Rollouts per round	160
Maximum rollout length	2048 tokens
Clipping coefficients	Lower 0.2; upper 0.28
Importance-sampling correction	Token-level; thresholds 2.0/0.0
Dynamic filtering	Remove zero-reward-variance prompt groups

Table 14 reports rollout sampling and DAPO parameters. Tables 12 and 13 specify Stages 1–2 of the 4B/8B models; Appendix **??** describes the separate component studies with 1.7B and 0.6B decoders.

D.5 AUDIO CONDITIONING

The acoustic representations and elapsed-time references condition autoregressive text generation:

$$p_\theta(y \mid a, u) = \prod_{t=1}^{|y|} p_\theta(y_t \mid y_{<t}, u, \mathcal{T}(Z), \mathcal{C}) \,. \tag{5}$$

Here $\mathcal{C}$ supplies features inside the decoder. The same factorization applies to ordinary and timestamped text.

Front end and adapters. The 10 ms spectrogram hop and three stride-2 convolutions yield an 80 ms feature cadence. Each adapter uses bias-free projections: $A(h) = W_{\text{down}}[\text{SiLU}(W_{\text{gate}}h) \odot W_{\text{up}}h]$. The primary adapter and three intermediate-feature adapters have separate parameters. Encoder indices 8, 16, and 24 select block outputs using zero-based indexing, corresponding to the ninth, seventeenth, and twenty-fifth encoder blocks.

Audio-position mapping. Let P be the ordered audio-token positions after time markers are inserted. Primary adapter outputs replace embeddings at these positions. Let D_k be the output of the kth decoder block (one-based here). Cross-layer injection gives

$$D_k[P] \leftarrow D_k[P] + C_k, \qquad k = 1, 2, 3. \tag{6}$$

The jth acoustic feature maps to the jth audio position even when marker digits occupy intervening positions. The direct addition affects only audio positions; other tokens access audio through decoder attention.

Time-token encoding and sequence length. Time markers use decimal-digit tokens from the tokenizer vocabulary. A marker is inserted after every 25 acoustic features and represents $2k$ seconds after the $25k$th feature, using one token per decimal digit. Any remaining features are retained without a trailing partial marker. For T acoustic features and marker lengths m_k, the primary sequence contains

$$N_{\text{primary}} = T + \sum_{k=1}^{\lfloor T/25 \rfloor} m_k. \tag{7}$$

One minute thus contributes 750 acoustic features and 30 marker occurrences, before boundary, instruction, and output tokens.

D.6 SUPERVISION CONSTRUCTION

Table 15: Event-guided supervision construction and training use.

Step	Operation
Segmentation	Merge speech intervals, combine other detected events, and apply the duration cap. Non-speech events up to 60 s participate in boundary selection.
Routing	Event tags select specialist branches; speech-caption annotation uses a speech or singing score of at least 0.5.
Annotation	Speech, sound, and music specialists provide transcripts, temporal information, and acoustic descriptions.
Fusion	Structured event and timing information guides a chronological caption combining the specialist outputs.
Target assembly	Transcripts supply ordinary and timestamped ASR targets. Merged and branch-specific captions supply holistic and focused supervision; captions also support QA generation.

Segmentation and annotation selection. The nine event categories are speech, non-speech human voice, singing, music, natural sounds, source-ambiguous sounds, sounds of things, channel/environment/background, and animal sounds. Category durations count the union of detected intervals within a clip. Overlapping spans within a category are merged before counting, while

simultaneous events from different categories are counted separately. Segmentation uses event-aligned boundaries with a hard-cut fallback at the duration cap. Speech-caption annotation requires a predicted speech or singing score of at least 0.5.

Specialist annotations. The ASR ensemble includes Qwen3-Omni-Instruct, FunASR Nano, and Qwen3-ASR (Xu et al., 2025; An et al., 2025; Shi et al., 2026). Language identification cross-checks fastText on recognized text with MMS-LID on the waveform (Joulin et al., 2017; Pratap et al., 2023). TorchAudio MMS_FA provides training-time forced alignment (Pratap et al., 2024); timestamp evaluation instead uses human gold-standard labels. Dense sound captions are generated by Qwen3-Omni-30B-Thinking and checked against transcripts, TimeAudio, and event metadata (Wang et al., 2026). The synthetic sound-caption path follows Timestamped Audio Captioning, composing sound sources and backgrounds with explicit event layouts to cover rare combinations, overlaps, and precise temporal boundaries (Kumar et al., 2026). Music analysis uses available outputs from tools including Chordino, BeatNet, and Essentia for harmonic, rhythmic, and tonal attributes (Mauch & Dixon, 2010; Heydari et al., 2021; Bogdanov et al., 2013).

Evidence selection and synthesis. Normalized class scores yield speech, music, and event priors for fusion. The routing policy uses these priors, top-ranked classes, and class uncertainty to select and order non-empty annotation slots. Its checks remove repetitive ASR, distinguish non-linguistic vocalizations from lexical speech, and reconcile speech/music claims with branch evidence. The planning prompt specifies the theme, evidence sources, and order; the generation prompt requests one English description preserving available temporal and speaker information. A deterministic fallback uses the same evidence order.