
Qwen-Audio-3.1-ASR Technical Report

Alibaba Token Foundry, Alibaba Group

 [Project Page](#)

Abstract

We present **Qwen-Audio-3.1-ASR**, a family of audio-language models that builds upon its predecessor Qwen-Audio-3.0-ASR with improved speech recognition and expanded capabilities in speaker-diarized transcription and general audio understanding. The family retains support for 30 languages and 16 Chinese dialects, together with instruction-controlled capabilities including historical-context conditioning, hierarchical hotword customization, and native transcription polishing. Beyond conventional transcription, speaker-diarized ASR produces structured outputs containing speaker labels, timestamps, and recognized text, supporting speaker attribution across turns and the representation of overlapping speech. The general audio understanding capabilities extend to audio-, speech-, and music-captioning, temporal grounding and sound event detection, as well as audio question answering and multiple-choice reasoning. These tasks broaden the application scope of the models from recognizing spoken words to interpreting speakers, acoustic events, and their temporal relations. To accommodate different deployment requirements, the family includes variants for asynchronous file transcription, low-latency streaming, interactive voice input, and non-streaming recognition, with speaker-diarized transcription and general audio understanding available in certain variants. We conduct evaluations spanning conventional ASR, multilingual and dialectal speech, speaker-diarized transcription, and general audio understanding, demonstrating the capabilities of the Qwen-Audio-3.1-ASR family across transcription and broader audio analysis tasks.

1 Introduction

Automatic speech recognition (ASR) provides the linguistic foundation for voice interfaces, meeting transcription, and conversational systems. However, a transcript alone only captures partial information in an audio recording. Multi-party conversations require identifying who spoke and when, while interpreting speech, environmental sounds, and music requires information that may not be recoverable from words alone. Practical audio systems must therefore address three complementary questions: *what was said* [1, 2], *who said it and when* [3], and *what the audio conveys beyond its transcript* [4, 5]. These requirements motivate extending accurate speech recognition toward structured transcription and general audio understanding.

Recent audio-language models have made substantial progress toward this broader scope. Qwen3-ASR demonstrates the potential of building multilingual speech recognition upon an audio-capable language model [6]. In parallel, Qwen3-Omni integrates audio perception into a broader multimodal framework [7], while MOSS-Audio [8] explicitly supports captioning, timestamped transcription, and time-aware audio understanding. More recently, FireRedAudio [9] combines recognition and audio understanding with speech generation in a general-purpose audio-language model. Together, these developments establish audio-conditioned language modeling as a useful framework for tasks that range from lexical recognition to descriptions and answers grounded in acoustic input.

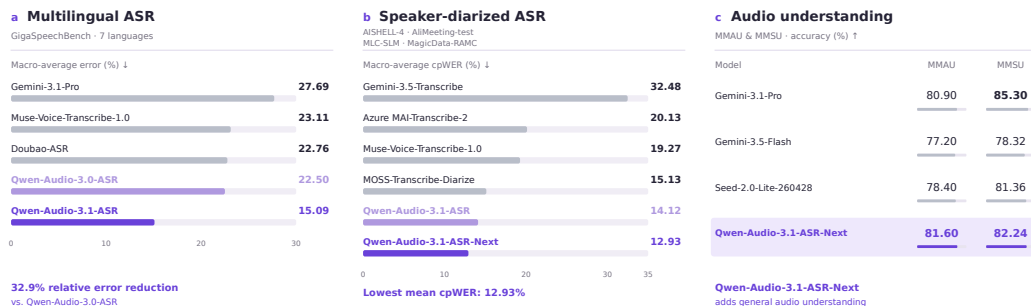


Figure 1: **Performance overview of Qwen-Audio-3.1-ASR on selected benchmarks.** (a) Multilingual ASR: unweighted mean of language-specific CER or WER across seven languages in GigaSpeechBench. Qwen-Audio-3.1-ASR reduces this mean error rate by 32.9% relative to Qwen-Audio-3.0-ASR. (b) Speaker-diarized ASR: unweighted mean cpWER across AISHELL-4, AliMeeting-test, MLC-SLM, and MagicData-RAMC. (c) General audio understanding: accuracy on MMAU and MMSU, evaluated using the **Qwen-Audio-3.1-ASR-Next** variant. All scores are percentages; lower is better in (a) and (b), and higher is better in (c). Purple highlights the Qwen-Audio models, whose names are shortened in the figure. Evaluation settings and full results are provided in the corresponding experimental sections.

Speaker-diarized transcription introduces a distinct structural requirement. Beyond recognizing individual utterances, a system must associate words with consistent speaker labels and temporal boundaries, including during rapid turn-taking and overlapping speech. Recent systems such as MOSS Transcribe Diarize [10] and VibeVoice-ASR [3] jointly model transcription, speaker diarization, and timestamp prediction, offering an alternative to assembling these outputs through separate processing stages. This direction is particularly relevant to meetings and interviews, where a lexically accurate transcript can still misrepresent a conversation if speaker assignments or temporal relations are incorrect.

General audio understanding further broadens the information that a model must preserve. Environmental acoustic events, vocal expressions, and musical characteristics can be central to audio comprehension even when they cannot be captured by a verbatim transcript. Moreover, identifying an event is not equivalent to locating it in time or reasoning about its relations to other sounds. Benchmarks such as MMAU [4] evaluate understanding and reasoning across speech, sound, and music, while MMSU [5] examines fine-grained spoken-language understanding involving semantic, paralinguistic, and phonological information. These complementary perspectives motivate evaluating audio models beyond transcription error rates alone.

The challenge for a production-oriented model family is to develop these capabilities while retaining reliable recognition and practical control over its outputs. Multilingual and dialectal speech, specialized terminology, and evolving conversational context continue to put demands on ASR accuracy. Speaker-aware outputs additionally require consistent attribution and timing, whereas audio understanding requires responses grounded in the recording rather than plausible text alone. Deployment also imposes different constraints: low-latency streaming, asynchronous file transcription, and richer non-streaming analysis need not expose identical task capabilities. We therefore treat **recognition quality**, **output structure**, **audio understanding**, and **deployment configuration** as related but distinct aspects of the system.

In this report, we present **Qwen-Audio-3.1-ASR**, a family of audio-language models that builds upon Qwen-Audio-3.0-ASR [11] and introduces two major capability extensions: *speaker-diarized ASR* and *general audio understanding*. The family retains support for 30 languages and 16 Chinese dialects, together with production-oriented controls such as contextual conditioning, hierarchical hotword customization, and native transcription polishing in the corresponding deployment profiles. Qwen-Audio-3.1-ASR supports speaker-diarized transcription, producing recognized text with recording-local speaker labels and timestamps. Its Next variant **Qwen-Audio-3.1-ASR-Next** additionally supports general audio understanding. Speaker-diarized transcription and general audio understanding are available through the corresponding non-streaming interfaces, but not through the Streaming or Message ASR profiles. Figure 1 summarizes representative results on multilingual ASR,

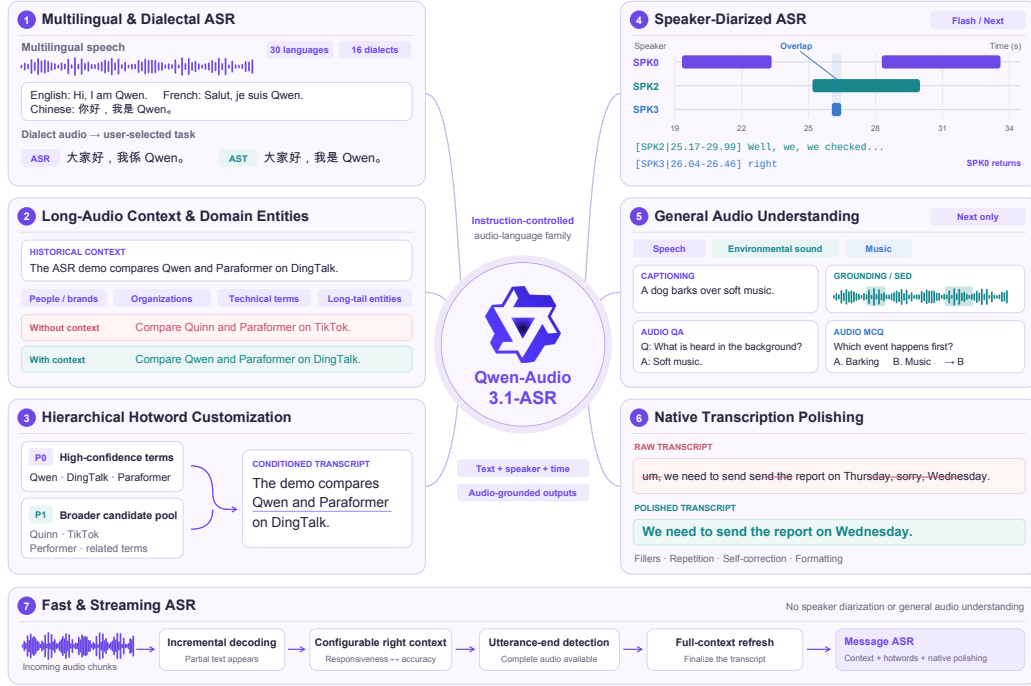


Figure 2: **Overview of Qwen-Audio-3.1-ASR and its capabilities.** The model family supports multilingual and dialectal speech recognition, user-selectable dialect transcription or translation into Chinese, long-audio context and domain entity recognition, hierarchical hotword customization, and native transcription polishing. Speaker-diarized ASR produces transcripts with speaker labels and timestamps, including overlapping speech, while general audio understanding supports captioning, temporal grounding, sound event detection, and audio question answering. Streaming recognition provides incremental transcripts for real-time display, followed by a non-streaming refresh at the utterance end. Capability availability depends on the model variant; streaming does not support speaker diarization or general audio understanding.

speaker-diarized transcription, and general audio understanding, highlighting improved multilingual recognition over Qwen-Audio-3.0-ASR and the capabilities introduced in the Qwen-Audio-3.1-ASR family.

The main advances of this generation are summarized as follows:

- **Enhanced speech recognition.** We strengthen the core ASR capability across multilingual, dialectal, and application-oriented recognition settings, building on the instruction-controlled recognition framework of the previous generation. Contextual conditioning, hotword customization, and transcription polishing remain complementary controls rather than substitutes for accurate recognition.
- **Speaker-diarized ASR.** We extend transcription to structured outputs containing recognized text, speaker labels, and timestamps. The capability targets multi-speaker recordings with speaker changes, short conversational responses, and overlapping speech, preserving conversational structure beyond a flat sequence of words.
- **General audio understanding.** In addition to transcription, the Next variant is trained on captioning tasks spanning speech, environmental sounds, and music, as well as temporal grounding and audio question answering in both open-ended and multiple-choice formats, achieving comprehensive audio understanding.

Our evaluations span speech recognition, speaker-diarized transcription, and audio understanding, using task-appropriate evaluation protocols to evaluate these different capabilities. The remainder of this report describes the model architecture in Section 2, training in Section 3, deployment variants in Section 4, and experiments in Section 5.

2 Model Architecture

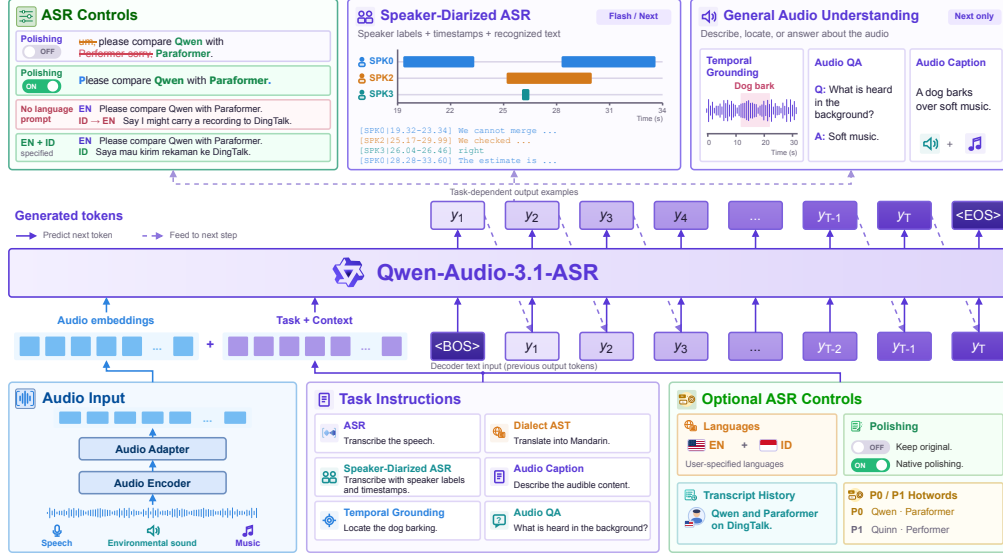


Figure 3: Overview of the instruction-controlled decoding workflow of Qwen-Audio-3.1-ASR.

Qwen-Audio-3.1-ASR builds upon the audio-language modeling framework of Qwen-Audio-3.0-ASR and extends its output space from speech transcription to speaker-diarized transcription and general audio understanding. As illustrated in Figure 3, the core framework consists of an audio encoder, an audio adapter, and an LLM-based decoder. Audio representations provide the acoustic evidence, while task instructions and optional textual context specify the desired output behavior. An auxiliary CTC branch, omitted from the figure for clarity, provides preliminary recognition hypotheses to support contextual processing. The available model capabilities depend on the model variant and deployment profile.

2.1 Audio Encoding and Instruction-Conditioned Decoding

The audio encoder transforms the input waveform into a sequence of acoustic representations. The audio adapter bridges these representations to the input space of the LLM decoder, allowing the decoder to generate outputs conditioned on both audio input and textual instructions. Unlike a transcript-only interface, this formulation retains direct access to the acoustic input when producing speaker-aware transcripts, audio descriptions, or answers to audio-related questions.

Let x denote the input audio, q the task instruction, and c the optional textual conditions. The conditional generation process can be represented as

$$H = A(E(x)), \quad p_{\theta}(y \mid x, q, c) = \prod_{t=1}^T p_{\theta}(y_t \mid y_{<t}, H, q, c), \quad (1)$$

where E is the audio encoder, A is the audio adapter, and H is the resulting sequence of audio embeddings. The model parameters are denoted by θ . The task-dependent output is represented as $y = (y_1, \dots, y_T)$, where T is the output sequence length, y_t is the token at position t , and $y_{<t} = (y_1, \dots, y_{t-1})$ is the preceding output prefix.

For ASR, the textual conditions can include target languages, historical transcripts, hierarchical hotword candidates, and a polishing instruction. These controls specify how the recognized content should be rendered or contextualized. For audio understanding, the instruction instead specifies a description, a target event, or a question to be answered from the recording.

The auxiliary CTC branch operates on the audio encoder representations and provides an initial recognition hypothesis for hotword customization and a coarse alignment signal for long-audio

context injection, as described in Sections 5.4 and 5.5. Its role is auxiliary: CTC hypotheses are not the sole representation of the recording, and should not be confused with the speaker-attributed timestamped transcriptions or event boundaries produced for the corresponding tasks.

2.2 Speaker-Diarized Transcription

Speaker-diarized ASR extends the transcription target to include speaker attribution and temporal information. Instead of returning a single undifferentiated text sequence, the model produces a structured transcript that associates each speech segment with its recognized content, speaker label, and temporal boundaries.

At the task level, a speaker-diarized transcript can be represented as

$$\mathcal{S} = \{(b_i, e_i, k_i, w_i)\}_{i=1}^M, \quad (2)$$

where $\mathcal{S}$ denotes the structured transcript and M is the number of speech segments. For segment i , b_i and e_i are the segment start and end times, k_i is a speaker label, and w_i is the corresponding transcription. This notation specifies the semantic content of the output rather than an exact implementation format. Speaker labels identify distinct speakers within a recording; they do not imply recognition of a person’s real-world identity.

The output must preserve speaker consistency across turns while accommodating short responses and overlapping speech. In particular, temporal intervals associated with different speakers need not be disjoint. Jointly representing text, speaker-diarization labels and timing allows the transcript to retain conversational structure that would otherwise be lost in plain-text ASR output.

2.3 General Audio Understanding

The Next variant broadens instruction-conditioned generation beyond lexical transcription to comprehensive audio understanding across speech, environmental sounds, and music. Specifically, its training incorporates three complementary task families within a unified framework: audio captioning, temporal grounding, and audio question answering.

Audio captioning. Audio captioning provides natural-language descriptions of audio content and its associated attributes. Depending on the audio type, these attributes may include speaker and paralinguistic characteristics, sound sources and acoustic environments, or musical properties such as genre, instrumentation, rhythm, and mood.

Temporal grounding. Temporal grounding associates a semantic query, such as a fine-grained caption or a sound event category, with its corresponding temporal locations in audio. We consider two grounding tasks: *temporal boundary localization* predicts the start and the end boundaries of a single target interval, whereas *temporal retrieval* identifies all time intervals that match the query. This formulation provides a unified view of single-interval localization and multi-interval retrieval.

Audio question answering. Audio question answering requires the model to interpret acoustic evidence in response to a given query. Both open-ended and multiple-choice formats are included, supporting the integration of audio perception with semantic reasoning.

2.4 Deployment Variants

Table 1 summarizes the five deployment variants and their supported capabilities. Flash is reported as Qwen-Audio-3.1-ASR in the experimental results, while Next is reported as Qwen-Audio-3.1-ASR-Next. Deployment workflows are detailed in Section 4.

3 Training

The training of Qwen-Audio-3.1-ASR comprises six stages: audio encoder pretraining, audio-language model pretraining, LLM cooldown adaptation, supervised fine-tuning, reinforcement learning, and multi-teacher on-policy distillation (MOPD). These stages progressively align acoustic representations with linguistic knowledge, develop instruction-following capabilities, and optimize

Table 1: Deployment variants of Qwen-Audio-3.1-ASR. A checkmark indicates a supported capability; a dash indicates an unsupported capability. SD denotes speaker-diarized ASR with recording-local speaker labels and timestamps; AU denotes general audio understanding. Context, hotwords, and polishing refer to historical-context conditioning, hierarchical hotword customization, and native transcription polishing, respectively. SD and AU are available only in the corresponding non-streaming variants.

Variant	Target scenario	Supported capabilities					
		ASR	SD	AU	Context	Hotwords	Polishing
Qwen-Audio-3.1-ASR-Flash-Filetrans	Asynchronous audio-file transcription	✓	—	—	✓	✓	—
Qwen-Audio-3.1-ASR-Flash-Streaming	Real-time transcription	✓	—	—	✓	✓	—
Qwen-Audio-3.1-ASR-Flash-Message	Interactive voice input	✓	—	—	✓	✓	✓
Qwen-Audio-3.1-ASR-Flash	Non-streaming transcription	✓	✓	—	✓	✓	✓
Qwen-Audio-3.1-ASR-Next	Non-streaming transcription and audio analysis	✓	✓	✓	✓	✓	✓

task performance, followed by MOPD to consolidate capabilities of multiple teacher models into a single model.

3.1 Pretraining of Audio Encoder

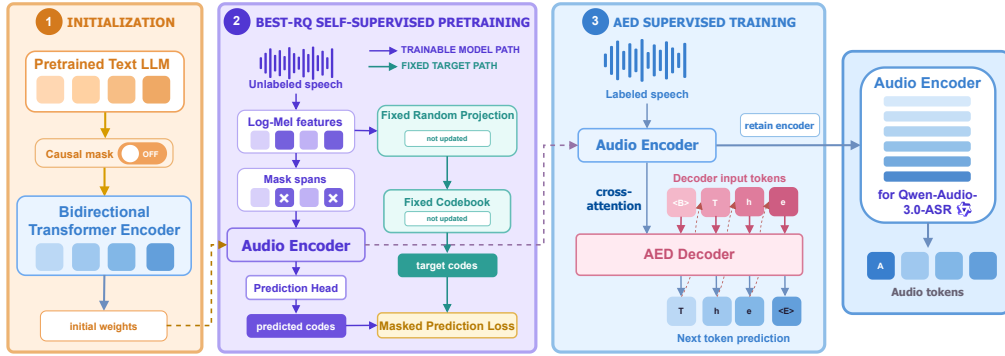


Figure 4: The pretraining pipeline for the audio encoder.

As illustrated in Figure 4, the audio encoder is trained in two consecutive stages: BEST-RQ self-supervised pretraining on large-scale unlabeled speech, followed by supervised training with an attention-based encoder–decoder (AED) objective on labeled audio–text data. Across the two stages, the pretraining corpus comprises tens of millions of hours of audio collected from diverse real-world scenarios.

BEST-RQ self-supervised pretraining. We first train the encoder using BEST-RQ [12] on large-scale unlabeled speech collected from diverse real-world scenarios, covering domains such as artificial intelligence, biotechnology, e-commerce, education, entertainment, finance, and mobility. Acoustic frames are randomly masked, and the encoder predicts discrete targets generated from the corresponding unmasked features by a fixed random projection and a frozen codebook. This objective enables the encoder to learn contextual speech representations from large-scale and diverse unlabeled audio. Following our previous study on cross-modal initialization [13], we initialize compatible Transformer layers of the encoder from a pretrained Qwen3 LLM [14]. For the non-streaming encoder, causal attention is replaced with bidirectional attention for speech encoding, while the acoustic front end is initialized separately. The resulting network is subsequently adapted to speech through BEST-RQ pretraining.

Supervised AED training. The pretrained encoder is then integrated into an AED model and optimized on large-scale labeled audio–text data, primarily covering Chinese and English. The labeled data are processed through a comprehensive pipeline that includes voice activity detection

(VAD), pseudo-label generation by multiple ASR systems, including Paraformer-V2 [15], Whisper-large-v3¹, and SenseVoice [16], followed by inverse text normalization (ITN).

Following the general training paradigm of SenseVoice-Large [16], an autoregressive decoder attends to the encoder representations and predicts the reference transcription token by token, providing direct acoustic–text supervision. After this stage, the AED decoder is discarded, and the trained encoder is used to initialize the audio encoder of Qwen-Audio-3.1-ASR for subsequent Audio LLM pretraining.

3.2 Pretraining of Audio LLM

Initialization. The Audio LLM decoder is initialized from the language-model component of Qwen, while its vision-specific components are not used. The backbone adopts a Mixture-of-Experts (MoE) architecture. This initialization provides substantial linguistic capacity and multilingual knowledge while maintaining relatively low per-token computation and decoding cost, making it suitable for Qwen-Audio-3.1-ASR, including its latency-sensitive streaming variant.

Training data and objective. During Audio LLM pretraining, we train the model on approximately 20 million hours of multilingual data constructed through our large-scale data pipeline, consisting primarily of ASR data and interleaved text–audio data. The pretrained audio encoder is kept frozen, while the audio adapter and the LLM decoder are jointly optimized. The model is trained with a cross-entropy loss over the target text tokens, enabling the audio adapter and LLM decoder to align acoustic representations with the LLM’s textual representation space. This stage establishes the fundamental audio–text alignment required for subsequent ASR adaptation.

The resulting Audio LLM combines the pretrained linguistic knowledge of Qwen with the audio–text alignment learned in this stage. It provides several useful priors for downstream ASR, including multilingual knowledge, world knowledge for resolving semantically ambiguous entities and homophones, and instruction-following capability for handling flexible textual conditions.

3.3 LLM Cooldown Adaptation

Following Audio LLM pretraining, we introduce a cooldown stage using approximately one million hours of higher-quality audio–text data. Compared with the preceding Audio LLM pretraining stage, this cooldown adaptation stage aims at refining audio–language alignment and emphasizes a more balanced distribution of languages and Chinese dialects, improved transcription quality, and a broader mixture of tasks and instructions before supervised fine-tuning. In addition to conventional ASR, the training mixture includes audio understanding, speaker diarization, and speech question-answering and dialogue data.

Data rebalancing and quality filtering. We rebalance the multilingual and dialectal ASR data to improve coverage across linguistic conditions. To further improve transcription quality, we re-transcribe the ASR data using VibeVoice and compare the resulting hypotheses with the existing reference transcripts. Samples with a word error rate greater than 0.3 are removed. This procedure uses agreement with an additional recognizer as a data-quality filter; it does not replace the reference transcripts with the re-recognition outputs.

Task diversification. The cooldown mixture broadens the training objectives beyond plain speech transcription. Audio understanding data introduce supervision for interpreting acoustic content, speaker diarization data expose the model to speaker-related structure, and speech question-answering and dialogue data enrich the range of audio-conditioned instructions and responses. Together with language and dialect rebalancing, this task mixture prepares the model for subsequent task-specific supervised fine-tuning.

The pretrained audio encoder remains frozen, while the audio adapter and LLM decoder are optimized using a cross-entropy objective over the target output tokens.

¹<https://huggingface.co/openai/whisper-large-v3>

3.4 Supervised Fine-tuning (SFT)

Supervised fine-tuning integrates the pretrained audio encoder, audio adapter, and cooldown-adapted LLM into an instruction-controlled audio-language model. Building on the recognition capabilities of the previous generation, the SFT mixture additionally incorporates speaker-diarized ASR and general audio understanding data. The task mixture is configured according to the capabilities of the corresponding model variant. During SFT, we jointly update the audio encoder and audio adapter, while keeping the base LLM parameters frozen and adapting the decoder through Low-Rank Adaptation (LoRA). Training examples follow a conversational format in which a user instruction and the associated audio are paired with an assistant response specifying the desired output. The model is optimized using an autoregressive cross-entropy objective over the target response. The ASR portion retains broad coverage of multilingual, dialectal, code-switching, noisy, far-field, and application-oriented speech. Recognition instructions may specify target languages, historical transcript context, hierarchical hotword candidates, or transcription polishing. The additional speaker-diarized ASR and audio understanding examples extend this instruction-following interface to structured transcription and audio-conditioned semantic tasks.

3.4.1 Recognition-oriented SFT

Specified-language transcription. To support flexible language control, the SFT mixture includes multilingual, dialectal, and code-switching speech covering the target languages and Chinese dialects. Each example may be accompanied by a list of target languages that specifies the expected transcription behavior. For monolingual transcription, the instruction requires the model to emit only speech belonging to the specified language and to avoid translating or rewriting speech from other languages into it. For multilingual speech, the model is instructed to preserve each supported language in its original written form. Training on diverse language combinations and code-switching examples enables the model to follow runtime language specifications and reduces unintended cross-language substitutions, particularly for lower-resource languages.

Domain-entity augmentation. To improve recognition of domain-specific entities that are individually rare in generic ASR data, we augment the SFT mixture with dedicated entity-focused examples. High-value entities are first mined from domains such as artificial intelligence, finance, healthcare, and IT, and an LLM generates diverse short utterances containing these entities based on retrieved factual context. The generated texts are converted into speech using CosyVoice3 [17], supplemented by human recordings for terms that are difficult to synthesize reliably. The resulting audio is then re-decoded by the current ASR model, and samples with unambiguous entity-level mismatches are filtered out. These curated examples are incorporated into SFT to increase the model’s exposure to rare and rapidly evolving domain terminology.

Native transcription polishing. A binary polishing instruction determines whether the target preserves the verbatim spoken form or produces a cleaned transcript. Polishing-enabled examples teach the model to remove non-semantic fillers, collapse stuttering and unnecessary repetitions, resolve explicit self-corrections, and normalize punctuation and textual formatting while preserving the speaker’s final intent. The polishing data are constructed from both real and synthetic examples. For real speech, we retain the original audio from existing ASR records and regenerate the textual supervision using a text LLM with production polishing instructions, followed by fidelity filtering to remove samples with semantic loss or unsupported insertions. For synthetic augmentation, an LLM injects fillers, repetitions, and self-corrections into canonical transcripts, and the resulting texts are converted into speech using TTS, followed by ASR-based pronunciation filtering and noise augmentation. Since the polished transcript itself serves as the ASR target, recognition and polishing are learned within the same autoregressive decoding process, without requiring a cascaded ASR–LLM rewriting pipeline.

Historical-context conditioning. For contextual training, transcripts or interaction records from preceding turns are prepended as reference information. The instruction explicitly indicates that the historical context may be incomplete, irrelevant, or incorrect, and that the current audio should remain the primary source of evidence. The context data are constructed from both real conversations and targeted synthetic examples. For real data, we match ASR records to their original sessions using audio identifiers, restore the chronological order of utterances, and use only preceding turns

as context without LLM rewriting. For synthetic augmentation, we construct examples involving homophone-ambiguous entities together with correct, conflicting, noisy, or empty historical contexts and hotword conditions. This mixture encourages the model to exploit recurring entities and topic cues when they are informative, while avoiding unconditional copying from the provided context.

Hierarchical hotword conditioning. To train hierarchical hotword conditioning, each utterance is paired with two tiers of candidate hotwords. **P0** contains high-confidence, high-priority terms, including key named entities and other contextually critical expressions, whereas **P1** contains a broader candidate pool. An LLM assigns candidates to the two tiers by semantically matching them against the reference transcription, promoting unambiguous contextual matches to P0 and retaining the remaining candidates in P1. During SFT, both candidate sets are included in the instruction as auxiliary conditioning inputs. The model learns to place greater preference on P0 candidates when supported by the acoustic and contextual evidence, while using P1 as a broader source of possible entities. This design improves long-tail hotword recall without forcing acoustically unsupported candidates into the transcript.

3.4.2 Speaker-diarized ASR

Training data. The training data combine carefully annotated human transcripts with model-generated supervision. For film and television data, we clean the recordings and use their original subtitles together with multiple specialized models, including language identification, sound event detection, and speaker diarization models, to cross-validate the annotations and curate high-quality speaker-attributed ASR examples. We additionally construct simulated meeting conversations with LLM-generated dialogue topics and varied reverberation and noise conditions to broaden coverage of conversational and acoustic scenarios.

Output serialization. Speaker-diarized ASR examples pair a multi-speaker recording with an instruction requesting both speaker attribution and transcription. The target response is serialized as newline-separated segments in the format `[SPK|start-end]text`. The example illustrates chronological serialization with overlapping time intervals and consistent speaker labels across turns.

Speaker-diarized transcription

[SPK0|19.36-23.36]

We can't merge until we know who owns the label definition for this release cycle.

[SPK2|25.20-30.00]

Well, we, we checked the documentation, yeah, and it is not clear at all.

[SPK3|26.08-26.48]

right.

[SPK0|28.32-33.60]

So the engineering estimate says it will take three days to fix the schema.

Segments are displayed in ascending order of start time; their time intervals may overlap. SPK3's brief response overlaps with SPK2, and SPK0 resumes before SPK2 finishes. Repeated speaker labels retain the same color. All timestamps are in seconds on an 80 ms grid.

Each segment contains a recording-local speaker label, a start–end time interval in seconds, and the corresponding recognized text. Repeated occurrences of the same speaker share the same label within the recording. The timestamps are expressed relative to the input audio, and intervals belonging to different speakers may overlap. The complete serialized response serves as the autoregressive training target, jointly supervising speaker labels, timestamps, and transcription text.

Speaker-token loss reweighting. To assess the balance of supervision across token categories, we examine token frequencies and loss contributions in the speaker-diarized ASR training data. Speaker-ID number tokens constitute 2.90% of all supervised tokens but contribute only 1.97% of the unweighted cross-entropy loss. Timestamp tokens account for 33.44% of tokens and 49.51% of loss, formatting tokens account for 14.48% and approximately 0.23%, and transcription tokens account for 46.50% and 47.74%, respectively. The remaining 2.68% of tokens and approximately 0.55% of loss are grouped as other supervised tokens, as shown in Figure 5.

These statistics reveal that speaker-ID prediction receives only a small share of the aggregate supervision despite its central role in speaker diarization. Timestamp prediction already accounts for a substantial fraction of the unweighted loss, while formatting is a relatively simple task that we observe the model already performs reliably. We therefore focus reweighting on speaker-ID prediction to better reflect its importance to diarization, rather than increasing the weight of every low-loss category. Specifically, we multiply the cross-entropy loss on speaker-ID number tokens by 10 while retaining unit weights for timestamp, formatting, transcription, and other supervised tokens. This adjusts the relative loss contributions without changing target sequences or token counts, and substantially improves both diarization error rate (DER) and concatenated minimum-permutation word error rate (cpWER) in our experiments.

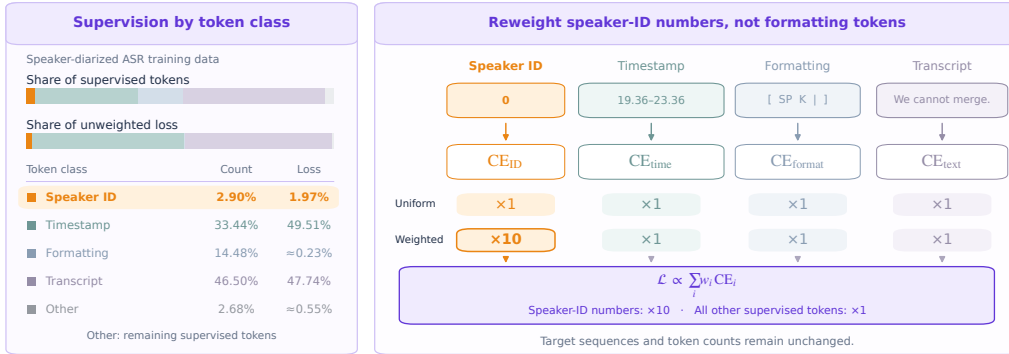


Figure 5: Speaker-token loss reweighting. Left: token-count and unweighted-loss shares for five categories in the speaker-diarized ASR training data. Other denotes supervised tokens outside the four named categories; its shares are calculated by subtraction. Formatting and other loss shares are approximate. Right: only speaker-ID number tokens receive a cross-entropy weight of 10; timestamp, formatting, transcription, and other supervised tokens retain a weight of 1. The example fields illustrate category membership rather than exact tokenizer segmentation. Target sequences and token counts remain unchanged. In the schematic loss expression, i indexes supervised output tokens, CE_i is the cross-entropy loss for token i , and w_i is its loss weight: 10 for speaker-ID number tokens and 1 for all other supervised tokens. The proportionality notation emphasizes relative token weights and omits loss normalization.

Frame-aligned timestamp quantization. We also find that the temporal resolution of the supervision materially affects optimization. Our initial targets retained the original timestamps at 1 ms precision. We instead quantize timestamps to an 80 ms grid, matching the 12.5 Hz frame rate of the audio representations produced by the audio encoder and adapter (Figure 6). Compared with retaining the original 1 ms precision, this quantization yields substantially faster convergence of the training loss and of both DER and cpWER, as well as better DER and cpWER under the same training-data mixture. Together, speaker-token loss reweighting and frame-aligned timestamp quantization improve learning of the speaker and temporal structure represented in the serialized output.

3.4.3 General Audio Understanding

For the Next variant, the SFT mixture is augmented with audio-understanding examples spanning speech, environmental sounds, and music. These examples follow the same conversational format as the transcription data, with each audio input paired with a task-specific instruction and a reference response.

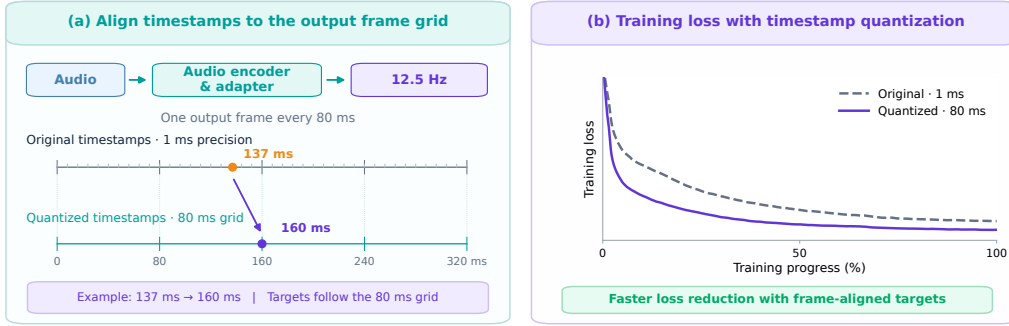


Figure 6: Frame-aligned timestamp quantization. (a) An illustrative timestamp is mapped from 1 ms precision to the 80 ms output frame grid of the audio encoder and adapter. (b) Training loss with original and quantized timestamps. Curves are digitized from training plots; absolute loss values and iteration counts are omitted. Both curves retain a shared vertical scale and use the same normalization of training progress.

The additional data cover three task families. Audio captioning examples supervise natural-language descriptions of audio content and its acoustic attributes. Temporal grounding examples pair fine-grained audio descriptions or sound-event queries with their annotated temporal intervals. Audio question-answering examples pair questions about the recording with reference answers in both open-ended and multiple-choice formats.

For all tasks, the model is trained to autoregressively predict the reference response conditioned on the audio and task instruction. This unified formulation extends supervision beyond lexical transcription to semantic descriptions, temporal annotations, and audio-grounded answers, while allowing the instruction to specify the required output form.

3.5 Reinforcement Learning (RL)

Following SFT, we apply reinforcement learning to optimize Qwen-Audio-3.1-ASR for transcription accuracy and speaker-diarized ASR. RL complements token-level maximum-likelihood training by directly optimizing sequence-level measures of recognition quality, speaker attribution, and temporal consistency.

RL data. We construct a compact, high-quality RL dataset focusing on challenging real-world cases. The ASR portion includes: (1) hard cases identified from disagreements between the base model and multiple external ASR systems, including Whisper, FireRed-ASR, and SenseVoice; (2) long-duration utterances exceeding 20 seconds; (3) hallucination-related samples, including both actual hallucinations and genuine long repetitions that may superficially resemble hallucinations; (4) keyword- and hotword-focused samples; and (5) regular ASR data included to mitigate catastrophic forgetting. Context-conditioned examples retain their associated historical context and instructions so that reward computation reflects the requested transcription behavior.

For speaker-diarized ASR, training examples pair multi-speaker recordings with reference transcripts, speaker labels, and segment timestamps. The structured references support joint evaluation of what is said, who says it, and when it occurs, using the output serialization described in Section 3.4.2. Each example retains its task identity and conditioning information, allowing the corresponding reward function to be selected for each generated response.

The FunVerl framework. We develop **FunVerl**, an audio-aware, fully asynchronous reinforcement learning framework built on verl [18]. FunVerl extends the training infrastructure with audio pre-processing, multimodal rollout generation, and task-specific reward computation. Its asynchronous execution decouples response generation from policy optimization, allowing these workloads to run concurrently on separate resources.

As illustrated in Figure 7, FunVerl consists of four modules orchestrated by Ray:

- **Audio processor.** Raw waveforms are converted into Mel-spectrogram features, while task instructions, context, and hotwords are tokenized. Audio features and text tokens are passed together to the rollout engine.
- **vLLM rollout engine.** The audio encoder and adapter transform Mel features into frame-level embeddings, which are combined with text embeddings. The MoE decoder then samples a group of G candidate responses for each audio–instruction pair.
- **Reward module.** Generated responses are scored using the reward associated with each sample’s task and data source. ASR rewards evaluate transcription and instruction-conditioned behavior, while speaker-diarized ASR rewards jointly assess speaker attribution, recognized words, and timing.
- **Megatron policy trainer.** Megatron-Core supports policy optimization with tensor, pipeline, and expert parallelism. Updated policy weights are synchronized to the rollout workers through an NCCL-based weight-transfer mechanism, avoiding full model reloading between updates.

The rollout engine pushes generated trajectories into a queue, while the trainer independently consumes scored batches and updates the policy. This overlaps generation, reward computation, and optimization. A policy-version staleness threshold excludes excessively outdated trajectories and limits the lag between the rollout and training policies.

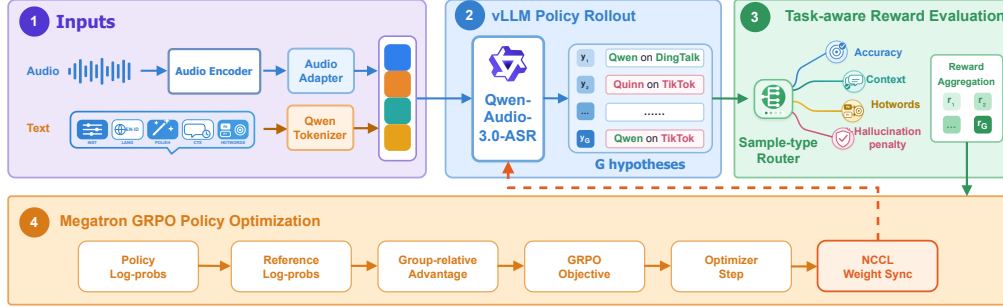


Figure 7: The FunVerl asynchronous RL framework. Audio preprocessing and vLLM rollout generation produce candidate responses, task-specific rewards score the responses, and the Megatron trainer updates the policy using GRPO. A trajectory queue decouples generation from optimization, and updated policy weights are synchronized to the rollout engine.

Policy optimization. We employ Group Relative Policy Optimization (GRPO) [19] for both tasks. Given an audio–instruction input q , a rollout policy $\pi_{\theta_{\text{old}}}$ generates responses $\{o_i\}_{i=1}^G$ with task-specific rewards $\{R_i\}_{i=1}^G$. The rewards are normalized within each group to obtain an advantage shared by the response tokens, without requiring a separately trained value network:

$$\hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G) + \delta}, \quad (3)$$

where $\delta > 0$ is a small constant for numerical stability. The policy maximizes the following group-level clipped surrogate objective with a KL regularizer:

$$J_{\text{GRPO}}(\theta) = \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left[\min(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_{i,t}) - \beta D_{\text{KL}}(\pi_{\theta}(\cdot | h_{i,t}) \| \pi_{\text{ref}}(\cdot | h_{i,t})) \right], \quad (4)$$

where $h_{i,t} = (q, o_{i,<t})$, $r_{i,t}(\theta) = \pi_{\theta}(o_{i,t} | h_{i,t}) / \pi_{\theta_{\text{old}}}(o_{i,t} | h_{i,t})$, π_{ref} is a fixed reference policy, ε is the clipping threshold, and β controls KL regularization.

3.5.1 ASR Reward

For ASR, the reward combines transcription accuracy with criteria for practical recognition scenarios. Optimizing a WER-derived reward shares the task-level motivation of minimum word error rate (MWER) training. We retain the following reward components, selected according to the sample type:

- **ASR Accuracy** (R_i^1). $1 - \text{WER}(y^*, y)$ as the primary reward, where y^* and y are the reference and hypothesized transcripts, respectively, and WER is expressed as a ratio.
- **Keyword Accuracy and Recall** (R_i^2). Keyword recall encourages correct entity recognition; keyword accuracy penalizes insertion errors to balance precision and recall. Keywords are either manually annotated or LLM-identified.
- **Hotword Reward** (R_i^3). For hotword-conditioned samples, a dedicated reward scores recognition of P0/P1 hotwords provided in the prompt, with asymmetric positive/negative gamma weighting.
- **Context Reward** (R_i^4). For context-conditioned samples, rewards hit/miss of context-derived entities with configurable penalties for false insertions.
- **Hallucination Suppression** (R_i^5). Hallucinated content is detected via pattern matching and penalized proportionally to the hallucinated span length.

FunVerl’s reward routing mechanism selects the appropriate reward configuration per sample based on its data source (general, dialect, hotword, or context), enabling a single training run to jointly optimize across these recognition scenarios.

3.5.2 Speaker-diarized ASR Reward

For speaker-diarized ASR, we use three complementary reward components to evaluate transcription quality, speaker attribution, and temporal consistency:

- **Diarization Reward (DER)**. A reward based on diarization error rate penalizes missed speech, false-alarm speech, and speaker confusion, encouraging accurate speaker activity and consistent attribution across turns.
- **Transcription Reward (cpWER)**. A reward based on concatenated minimum-permutation word error rate evaluates the transcript concatenated for each speaker under the best speaker permutation. It encourages accurate recognition and speaker assignment without depending on arbitrary speaker-ID names.
- **Temporal Consistency Reward (tcpWER)**. A reward based on time-constrained cpWER additionally requires temporally compatible word alignments, encouraging recognized words to be associated with the correct time intervals.

Lower error rates correspond to better reward values. Together, these components encourage the model to produce accurate words, consistent speaker labels, and reliable timestamps within the same structured response.

3.6 On-policy Distillation

As illustrated in Figure 8, we introduce on-policy distillation (OPD) to consolidate complementary capabilities from task-specialized teachers into a single audio-language student. Whereas RL supplies sequence-level feedback through task metrics, OPD supplies token-level feedback from a teacher on responses generated by the student. This provides a common training mechanism for transferring transcription, speaker-diarization, and audio-understanding capabilities.

3.6.1 Learning from Student-generated Responses

In conventional supervised distillation, the student is often trained on fixed reference responses or teacher-generated sequences. Such sequences can differ from the student’s own predictions at inference time. OPD addresses this distribution mismatch by sampling responses from the current student and evaluating the teacher on the resulting student-generated prefixes. The teacher therefore provides guidance at the states that the student actually visits, including prefixes containing the student’s own mistakes.

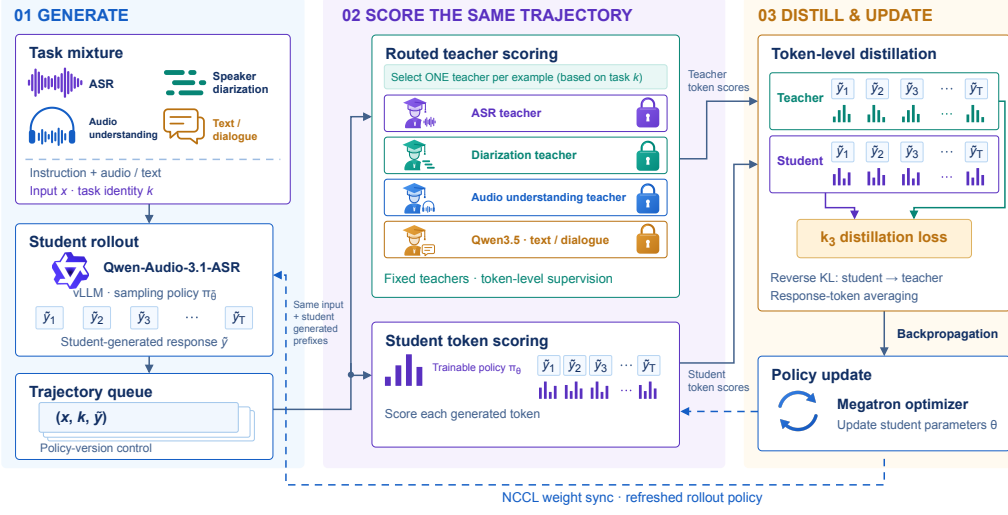


Figure 8: Overview of multi-teacher on-policy distillation in FunVerl. Student-generated responses are scored by task-routed teachers and the student using the same input and response prefixes. The k_3 objective provides token-level distillation supervision, while NCCL weight synchronization refreshes the rollout policy after student updates.

Let x denote the task input, including the instruction and any audio or conversational context, and let k identify the corresponding teacher. At each iteration, a student rollout policy $\pi_{\tilde{\theta}}$ samples a response $\tilde{y}$. The fixed teacher p_{ϕ_k} then scores the sampled tokens conditioned on the same input and student-generated prefix $\tilde{y}_{<t}$.

We use the k_3 **estimator** of the reverse KL divergence from the student to the teacher. For each generated token, we define the teacher-to-student probability ratio and its distillation penalty as

$$\rho_t(\theta) = \frac{p_{\phi_k}(\tilde{y}_t | x, \tilde{y}_{<t})}{\pi_{\theta}(\tilde{y}_t | x, \tilde{y}_{<t})}, \quad d_t^{(k_3)}(\theta) = \rho_t(\theta) - \log \rho_t(\theta) - 1. \quad (5)$$

The k_3 penalty is nonnegative and uses teacher and student log-probabilities for the sampled token. Under on-policy sampling, its expectation estimates $D_{\text{KL}}(\pi_{\theta} \| p_{\phi_k})$ at the corresponding prefix. The distillation objective averages these token-level penalties over student-generated responses:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{(x,k) \sim \mathcal{D}_{\text{distill}} \atop \tilde{y} \sim \pi_{\tilde{\theta}}(\cdot | x)} \left[\frac{1}{|\tilde{y}|} \sum_{t=1}^{|\tilde{y}|} d_t^{(k_3)}(\theta) \right], \quad (6)$$

where $\mathcal{D}_{\text{distill}}$ is the distribution of inputs paired with their assigned teachers. Teacher parameters and sampled trajectories are held fixed during each student update, and new trajectories are generated as the student evolves. FunVerl’s rollout, training, and weight-synchronization components provide the underlying execution pipeline; in asynchronous execution, policy-version control limits the lag of student-generated trajectories.

3.6.2 Teacher Models and Task Routing

We use three specialized audio teachers together with a Qwen3.5 teacher for text and dialogue. Examples are routed to the corresponding teacher according to the requested task:

- **ASR teacher.** The ASR teacher guides speech-to-text prediction, transferring its recognition capabilities to the student under the corresponding transcription instructions.
- **Speaker-diarization teacher.** The speaker-diarization teacher provides supervision for speaker-diarized transcription, including speaker labels, temporal boundaries, and the associated words. Its role is to guide the joint structured output, complementing the metric-based reward in Section 3.5.2.
- **Audio understanding teacher.** The audio understanding teacher guides responses that require interpreting speech, environmental sounds, or music beyond literal transcription, such as audio descriptions and answers to audio-conditioned questions.

- **Text and dialogue teacher (Qwen3.5).** Qwen3.5 provides token-level supervision for text and dialogue examples, helping preserve the student’s language modeling, instruction-following, and conversational capabilities during audio specialization.

Task-based routing lets each teacher supervise its corresponding data while contributing to the same student. The instruction determines whether the output is a transcript, a speaker-attributed transcript with timestamps, an audio-grounded answer, or a text response.

3.6.3 Training Data Mixture

The training mixture combines **ASR**, **speaker diarization (SD)**, and **audio understanding (AU)** data with **text and dialogue data**. ASR examples cover speech transcription; SD examples retain speaker and timing structure; and AU examples retain the task instructions needed to interpret and reason about audio. The three audio data groups are paired with their corresponding specialized teachers, preserving their distinct output requirements within a shared training pipeline.

Text and dialogue examples are distilled from **Qwen3.5** using the same student-generated rollout and k_3 objective. For text examples, the input contains the textual prompt or instruction; dialogue examples additionally preserve the preceding conversational context. These data help retain general language and conversational capabilities while the audio data develop recognition, speaker attribution, and audio understanding. The resulting mixture supports both audio-grounded tasks and context-appropriate text responses within the same student.

4 Deployment

Qwen-Audio-3.1-ASR provides deployment variants for file transcription, low-latency streaming, context-aware voice input, speaker-diarized transcription, and general audio understanding. The Streaming and Message ASR profiles retain the workflows of Qwen-Audio-3.0-ASR. The family additionally introduces explicit selection between dialect transcription and dialect-to-Mandarin translation, as well as speaker-diarized transcription and general audio understanding in the corresponding variants. These capabilities are variant-dependent rather than uniformly available across all deployment profiles.

4.1 Multilingual and Dialectal ASR

Qwen-Audio-3.1-ASR retains the instruction-controlled multilingual recognition interface of Qwen-Audio-3.0-ASR, supporting 30 languages and 16 Chinese dialectal varieties spanning eight major dialect regions. Applications can specify target languages at inference time to accommodate multilingual and code-switching speech without switching between language-specific checkpoints. For multilingual transcription, supported languages are preserved in their original writing systems.

A new capability in this generation is user-selectable output behavior for Chinese dialectal speech. Through an explicit task instruction, users can choose between ASR and automatic speech translation (AST):

- **Dialect ASR.** The model transcribes the dialectal speech while preserving its dialect-specific wording and expressions, rather than translating them into Mandarin.
- **Dialect-to-Mandarin AST.** The model translates the dialectal speech into written Mandarin, conveying the meaning of the utterance rather than preserving its original dialectal expressions.

This explicit choice separates faithful dialect transcription from dialect-to-Mandarin translation. It allows applications to select the appropriate output for scenarios such as dialect documentation or cross-dialect communication. The ASR/AST selection specifies the linguistic task and is distinct from transcription polishing, which controls the removal of disfluencies and the presentation of recognized text.

4.2 Low-Latency Streaming Recognition

The streaming deployment profile retains the hybrid incremental and full-context recognition workflow of Qwen-Audio-3.0-ASR. Incoming audio is processed incrementally to provide provisional

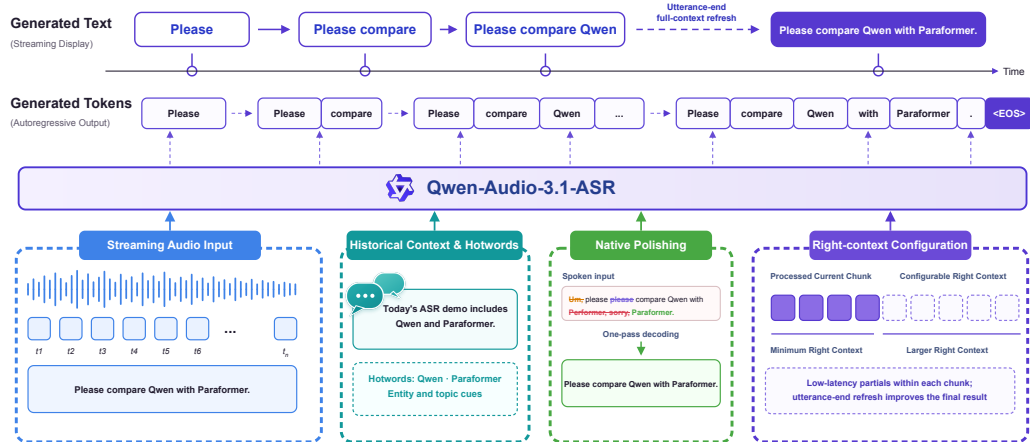


Figure 9: Overview of the Message ASR workflow used in Qwen-Audio-3.1-ASR, following the design of Qwen-Audio-3.0-ASR. Streaming audio is decoded incrementally to provide responsive partial transcripts. Historical context supplies entity and topic cues, native polishing improves readability within the same decoding pass, and configurable right context balances streaming latency and recognition accuracy. At utterance end, the model refreshes the displayed hypothesis using the complete available context.

transcripts before an utterance is complete. After endpoint detection, the complete utterance is used to refresh and finalize the transcript. This workflow combines immediate textual feedback with recognition informed by the full utterance context.

Low-latency recognition is achieved by processing audio in short chunks with limited right context. Chunk size and right-context length provide configurable operating points for the latency–accuracy trade-off: shorter contexts prioritize responsiveness, while longer contexts provide additional acoustic evidence before the transcript is updated. Partial hypotheses may be revised as more audio becomes available.

The streaming variant supports incremental speech recognition but does not provide speaker-diarized transcription or general audio understanding. These capabilities are available through the corresponding non-streaming variants.

4.3 Message ASR for Context-Aware Transcription

The Message ASR deployment profile follows Qwen-Audio-3.0-ASR, retaining its contextual conditioning, hierarchical hotword customization, and native transcription polishing workflow. It is designed for applications such as voice input and conversation documentation, where spontaneous speech must be converted into readable text and terminology may recur across successive utterances.

Historical conversation turns or preceding transcript segments provide topic and entity cues. Hierarchical hotword candidates distinguish a small set of high-confidence terms from a broader candidate pool. These conditions guide recognition together with the acoustic input, supporting the transcription of names, products, and specialized terminology.

A polishing instruction controls whether the output preserves the spoken form or produces a cleaned transcript. When enabled, native polishing removes non-semantic fillers and redundant repetitions, resolves explicit self-corrections, and normalizes punctuation and formatting while preserving the speaker’s intended meaning. This processing takes place within recognition rather than through a separate post-recognition rewriting service.

In the interactive workflow, incremental hypotheses provide immediate feedback, and an utterance-end full-context refresh produces the finalized transcript. Language specifications, historical context, hotwords, and polishing can be configured according to the application. These controls remain unchanged from the previous generation and do not imply support for speaker-diarized output or general audio understanding in the Message ASR profile.

4.4 Speaker-Diarized Transcription

The non-streaming Flash and Next variants support speaker-diarized transcription for multi-speaker recordings. Given an audio input and a corresponding task instruction, the model returns recognized text associated with recording-local speaker labels and temporal boundaries.

The output uses the segment format [SPK|start-end] text, where the time interval is expressed in seconds relative to the input recording. Repeated occurrences of a speaker retain the same label within the recording, and intervals from different speakers may overlap. This representation preserves speaker turns, brief responses, and simultaneous speech rather than reducing the conversation to a single unstructured transcript.

The resulting segments support speaker-aware transcript display, navigation to the corresponding audio intervals, and downstream analysis of conversational content. Speaker labels distinguish participants within a recording and do not imply real-world identity recognition.

4.5 General Audio Understanding

The Next variant supports instruction-conditioned understanding of speech, environmental sounds, and music. Given an audio recording together with a task instruction or question, the model produces the corresponding textual response or temporal annotations.

Audio captioning. The model generates natural-language descriptions of audible content, including acoustic events, spoken delivery, and musical characteristics. This mode provides information beyond verbatim transcription and is applicable to recordings with or without speech.

Temporal grounding. An input query specifies the acoustic content to be localized within the recording. For single-interval localization, the query identifies the start and end boundaries of one target interval, whereas for multi-interval retrieval, it identifies all matching intervals. The resulting annotations provide a unified representation for both temporal grounding tasks.

Audio question answering. The model supports both open-ended and multiple-choice questions about the audio. These questions may concern audible content, acoustic characteristics, or temporal relations among events. This interface enables audio-conditioned analysis without first converting the recording into a textual transcript.

These modes complement rather than replace speech recognition. ASR recovers spoken content, speaker-diarized transcription introduces speaker attribution and temporal structure, and general audio understanding addresses tasks that require broader acoustic evidence.

5 Evaluations

We evaluate the Qwen-Audio-3.1-ASR family on public benchmarks and internal application-oriented test sets covering conventional ASR, multilingual and dialectal recognition, speaker-diarized transcription, and general audio understanding. We include Qwen-Audio-3.0-ASR results on shared Chinese, English, and multilingual ASR benchmarks to characterize changes across model generations.

For conventional and multilingual ASR, we use the corresponding test sets of AISHELL-1 [20], AISHELL-2 [21], FLEURS [22], LibriSpeech [23], WeNetSpeech [24], GigaSpeechBench [25], and Common Voice 15 [26]. Public dialect evaluations additionally include KeSpeech and WSYue. Speaker-diarized transcription is evaluated on AISHELL-4, AliMeeting-test, MLC-SLM, and MagicData-RAMC. General audio understanding is evaluated on MMAU [4], MMSU [5], CochIScene, Nonspeech7k, MELD, NSynth, and DiDiSpeech, with temporal grounding evaluated on Clotho-Moment and DESED. Task-specific subsets, metrics, and evaluation settings are described in the corresponding subsections.

In addition to these public benchmarks, we evaluate the system on internal test sets covering dialect ASR and AST, industry-specific entities and long-tail terminology, hotword customization, transcription polishing, and streaming recognition. These evaluations span diverse acoustic conditions, speaking styles, and application domains. We also present qualitative examples of long-audio

contextual recognition to illustrate how preceding transcription history can assist the recognition of recurring entities and technical terms.

Evaluated models and deployment profiles. In the result tables and figures, Qwen-Audio-3.1-ASR denotes the base model evaluated through its non-streaming interface for ASR and speaker-diarized transcription. Qwen-Audio-3.1-ASR-Next denotes the Next variant evaluated for speaker-diarized transcription and general audio understanding; where the name Qwen-Audio-3.1-ASR-Flash-Next is used, Flash specifies the model-size designation. Message ASR latency and industry-term recall are evaluated using Qwen-Audio-3.1-ASR-Flash-Message, with streaming outputs used for real-time display and non-streaming outputs used for the utterance-end refresh. Deployment profiles and their capability scope are summarized in Table 1. Published baseline results and results obtained through our own evaluation are distinguished in the corresponding captions.

5.1 Public Benchmark Results

5.1.1 Chinese and English ASR Benchmarks

We evaluate Qwen-Audio-3.1-ASR on seven widely used Chinese and English ASR test sets, with results summarized in Table 2. Panel A presents baseline results collected from public model repositories and technical reports alongside our results, including Qwen-Audio-3.0-ASR as a reference for the previous generation. Panel B presents results obtained using our unified API-based evaluation pipeline. Given the differences in evaluation settings and result provenance, we discuss the two panels separately and base direct quantitative comparisons primarily on Panel B.

Comparison with published benchmark results. The results in Panel A indicate that Qwen-Audio-3.1-ASR maintains broadly comparable Chinese and English recognition performance to Qwen-Audio-3.0-ASR while extending the model family to speaker-diarized transcription and general audio understanding. On the Chinese test sets AISHELL-1, AISHELL-2, and FLEURS-zh, it achieves CERs of 1.08%, 2.05%, and 2.25%, respectively, compared with 1.03%, 2.02%, and 2.08% for the previous generation. On LibriSpeech-clean, the two generations achieve WERs of 1.23% and 1.19%, respectively. On LibriSpeech-other, Qwen-Audio-3.1-ASR reduces WER from 2.24% to 2.19%, achieving the lowest value in the table. On FLEURS-en, the corresponding WERs are 4.72% and 4.17%, representing the largest absolute difference between the two generations among these test sets. Overall, the two generations exhibit similar recognition performance, with differences varying across test sets.

Comparison under the unified evaluation pipeline. In Panel B, Qwen-Audio-3.1-ASR achieves the lowest error rate on four of the seven test sets. Compared with the strongest baseline on each of these four test sets, it reduces the error rate by 0.13 percentage points (5.5% relative) on FLEURS-zh, 0.23 percentage points (4.6% relative) on FLEURS-en, 0.12 percentage points (8.9% relative) on LibriSpeech-clean, and 0.32 percentage points (12.7% relative) on LibriSpeech-other. Fun-ASR-Flash achieves the lowest error rate on the remaining three test sets: AISHELL-1, AISHELL-2, and WeNetSpeech-net. All systems in Panel B are evaluated using the same corrected reference transcripts for WeNetSpeech-net. These results demonstrate competitive recognition performance across Chinese and English benchmarks, with relative performance varying across test sets.

5.1.2 Multilingual ASR Benchmarks

We evaluate the multilingual recognition performance of Qwen-Audio-3.1-ASR on selected language subsets of three public benchmarks: GigaSpeechBench, Common Voice 15, and FLEURS. Table 3 reports WER or CER according to the language-specific evaluation protocol and includes Qwen-Audio-3.0-ASR as a reference for the previous generation. Macro-average error rates are computed as the unweighted mean across the evaluated languages and are reported only for systems with results on every language in the corresponding benchmark subset.

On the seven East and Southeast Asian languages evaluated in GigaSpeechBench, Qwen-Audio-3.1-ASR achieves the lowest error rate on every language and the lowest macro-average error rate of 15.09%, compared with 22.50% for Qwen-Audio-3.0-ASR. Improvements over the previous generation are observed across all seven languages, with particularly large absolute reductions on Japanese, Malay, and Thai.

Table 2: Performance on **public ASR benchmark test sets**. We report CER (%) for Chinese test sets and WER (%) for English test sets. For baselines, Panel A presents results taken from the official public repositories or the original technical reports, whereas Panel B presents results obtained using our unified API-based evaluation pipeline on the same test sets. A dash indicates that a result was not reported in the original source. **Direct quantitative comparisons are primarily made within Panel B** since the results in Panel A may involve source-specific preprocessing and scoring conventions. All WeNetSpeech-net results in Panel B use corrected reference transcripts following [WenetSpeech issue #63](#). All AISHELL-2 results use the test-ios subset. Lower values indicate better performance, and the best result within each panel on each test set is bold-faced.

Panel A: Published Benchmark Results								
Test set	Metric	Kimi -Audio	FireRedASR2 -LLM	FireRed Audio	LongCat -Next	Qwen3-ASR -1.7B	Qwen-Audio -3.0-ASR	Qwen-Audio -3.1-ASR
AISHELL-1	CER	0.60	0.64	0.71	1.47	–	1.03	1.08
AISHELL-2	CER	2.56	2.15	2.63	2.82	2.71	2.02	2.05
FLEURS-zh	CER	2.69	–	3.14	3.24	2.41	2.08	2.25
FLEURS-en	WER	4.44	–	2.53	5.24	3.35	4.17	4.72
LibriSpeech-clean	WER	1.28	–	0.67	1.63	1.63	1.19	1.23
LibriSpeech-other	WER	2.42	–	2.91	3.42	3.38	2.24	2.19
WeNetSpeech-net	CER	5.37	4.44	5.18	5.98	4.97	4.31	4.37

Panel B: Results from Our API Evaluation								
Test set	Metric	GPT-4o Transcribe	Azure MAI- Transcribe-2	Doubao-ASR	Tencent Hy-ASR-3.0- preview	Muse-Voice Transcribe-1.0	Fun-ASR -Flash	Qwen-Audio -3.1-ASR
AISHELL-1	CER	3.34	1.49	1.54	0.78	6.28	0.76	1.08
AISHELL-2	CER	4.47	2.82	2.66	2.15	6.80	1.93	2.05
FLEURS-zh	CER	6.14	5.38	8.18	2.38	8.34	5.52	2.25
FLEURS-en	WER	5.79	5.38	6.50	5.45	7.48	4.95	4.72
LibriSpeech-clean	WER	1.58	1.68	3.36	1.60	2.17	1.35	1.23
LibriSpeech-other	WER	3.87	3.13	4.46	2.68	5.10	2.51	2.19
WeNetSpeech-net	CER	12.27	4.38	4.81	4.16	10.84	3.83	4.37

On the Common Voice 15 subset, which comprises five Asian languages as well as Spanish, Portuguese, and French, Qwen-Audio-3.1-ASR achieves the lowest macro-average error rate of 3.74% among systems with complete results and the lowest error rate on five of the eight languages: Indonesian, Japanese, Thai, Korean, and Portuguese. Tencent Cloud ASR achieves the lowest WER on Vietnamese, while Qwen-Audio-3.0-ASR retains the lowest WER on Spanish and French. Compared with the previous generation, Qwen-Audio-3.1-ASR shows particularly pronounced improvements on Japanese and Korean, reducing CER from 5.64% to 1.40% and from 3.09% to 0.73%, respectively. A full-subset macro average is not reported for Qwen-Audio-3.0-ASR because its Portuguese result is unavailable.

On the FLEURS subset, comprising seven Asian languages together with Spanish and Portuguese, Qwen-Audio-3.1-ASR achieves the lowest error rate on Indonesian, Japanese, and Korean. Its macro-average error rate is 3.85%, compared with 4.94% for Qwen-Audio-3.0-ASR, with lower error rates on all seven Asian languages. Azure MAI-Transcribe-2 achieves the lowest macro-average error rate of 3.25% and the lowest error rates on Thai, Filipino, Vietnamese, and Malay, while Qwen-Audio-3.0-ASR retains the lowest WER on Spanish and Portuguese. Overall, these results demonstrate improved multilingual recognition over the previous generation across most shared language subsets, particularly on the evaluated Asian languages, while relative performance varies across languages and benchmarks.

5.2 Chinese Dialect ASR and AST

We evaluate Qwen-Audio-3.1-ASR against Doubao-ASR and Tencent Hy-ASR-3.0-preview on public dialect benchmarks and internal test sets. ASR requires transcription in the corresponding dialect, whereas AST requires translation of dialectal speech into Mandarin. We report character error rate (CER) for the public benchmarks and internal ASR evaluation, and semantic sentence accuracy for

Table 3: Performance of **multilingual ASR** on GigaSpeechBench, Common Voice 15, and FLEURS.

Results are reported as CER or WER (%), as specified in the second column; lower is better. Qwen-Audio-3.0-ASR results are taken from its technical report [11]. Japanese CER for all systems is computed using the same normalization and scoring protocol specified in [Fun-ASR PR #165](#). Macro averages are unweighted means across the listed languages within each benchmark and are reported only when all corresponding results are available. A dash indicates an unavailable result, not necessarily an unsupported language. Bold values denote the lowest error rate in each row.

Language	Metric	Doubao-ASR	Tencent Cloud ASR	Gemini 3.1 Pro	GPT-4o Transcribe	Azure MAI-Transcribe-2	Muse Voice Transcribe-1.0	Qwen-Audio -3.0-ASR	Qwen-Audio -3.1-ASR
GigaSpeechBench									
Indonesian	WER	27.73	32.55	24.47	39.52	29.18	22.90	23.68	16.69
Japanese	CER	29.47	34.56	39.12	45.46	34.17	32.05	28.48	18.47
Thai	CER	17.65	31.40	17.37	53.55	22.42	15.27	16.22	7.75
Filipino	WER	25.36	33.38	37.41	37.73	32.95	25.48	29.45	21.69
Vietnamese	WER	9.53	9.61	15.55	32.89	12.77	10.76	11.54	8.04
Korean	CER	14.52	15.00	19.70	32.52	15.96	16.52	13.50	7.98
Malay	WER	35.05	41.06	40.19	51.59	38.18	38.81	34.65	25.04
Macro average	–	22.76	28.22	27.69	41.89	26.52	23.11	22.50	15.09
Common Voice 15									
Indonesian	WER	4.25	7.79	12.92	8.74	4.33	5.98	3.66	3.32
Japanese	CER	6.44	12.02	11.52	11.22	7.21	12.24	5.64	1.40
Thai	CER	2.32	4.17	12.33	6.64	1.90	2.94	1.56	1.43
Vietnamese	WER	6.31	5.67	10.64	15.50	5.93	7.77	8.29	8.44
Korean	CER	3.89	8.16	4.86	5.80	2.89	7.49	3.09	0.73
Spanish	WER	5.42	4.95	3.97	4.18	3.64	5.09	3.09	3.23
Portuguese	WER	8.06	6.24	8.65	8.78	4.89	6.49	–	4.47
French	WER	8.89	22.06	10.51	9.26	7.38	10.72	6.69	6.87
Macro average	–	5.70	8.88	9.43	8.77	4.77	7.34	–	3.74
FLEURS									
Indonesian	WER	5.24	12.77	3.72	3.73	2.62	8.16	2.89	2.01
Japanese	CER	3.16	15.39	3.95	2.98	1.85	6.55	1.61	0.36
Thai	CER	6.18	11.38	6.05	4.99	3.85	8.27	6.73	5.37
Filipino	WER	10.34	18.05	8.33	7.42	5.98	9.95	11.48	10.80
Vietnamese	WER	6.32	15.19	4.70	3.23	2.42	3.97	4.95	4.46
Korean	CER	4.86	8.93	4.24	4.06	3.92	6.87	4.94	2.38
Malay	WER	8.14	17.13	4.22	4.21	4.08	9.54	8.79	5.12
Spanish	WER	5.31	8.09	2.62	2.13	2.04	5.61	1.23	1.95
Portuguese	WER	7.19	13.71	3.46	2.80	2.53	7.12	1.87	2.21
Macro average	–	6.30	13.40	4.59	3.95	3.25	7.34	4.94	3.85

the internal AST evaluation. Macro averages are computed with equal weight for each included subset, separately for each evaluation configuration.

Public dialect benchmarks. We evaluate CER on 11 public benchmark subsets: nine KeSpeech subsets under the AST setting and the long- and short-form WSYue Cantonese subsets under the ASR setting. The four WSC Sichuan subsets are excluded from this comparison and its macro averages.

As shown in Figure 10, Qwen-Audio-3.1-ASR achieves a macro-average CER of 4.55%, compared with 7.15% for Doubao-ASR and 4.61% for Tencent Hy-ASR-3.0-preview. It achieves the lowest CER on six subsets: KeSpeech-Beijing, KeSpeech-Ji-Lu, KeSpeech-Jiang-Huai, KeSpeech-Jiao-Liao, KeSpeech-Mandarin, and WSYue-long. These results indicate a substantial improvement over Doubao-ASR and a small macro-average advantage over Tencent Hy-ASR-3.0-preview on the selected public benchmarks.

Internal dialect ASR. The internal evaluation includes an ASR configuration covering 16 Chinese dialects and an AST configuration covering 11 dialects. These configurations are evaluated separately rather than assigning each dialect exclusively to one task. For the internal ASR evaluation, all subsets require transcription in the corresponding dialect and are evaluated using CER.

As shown in Figure 11, Qwen-Audio-3.1-ASR achieves the lowest CER on 11 of the 16 internal ASR subsets. Its macro-average CER is 10.38%, compared with 20.23% for Doubao-ASR and 17.13% for Tencent Hy-ASR-3.0-preview, corresponding to absolute reductions of 9.85 and 6.75 percentage

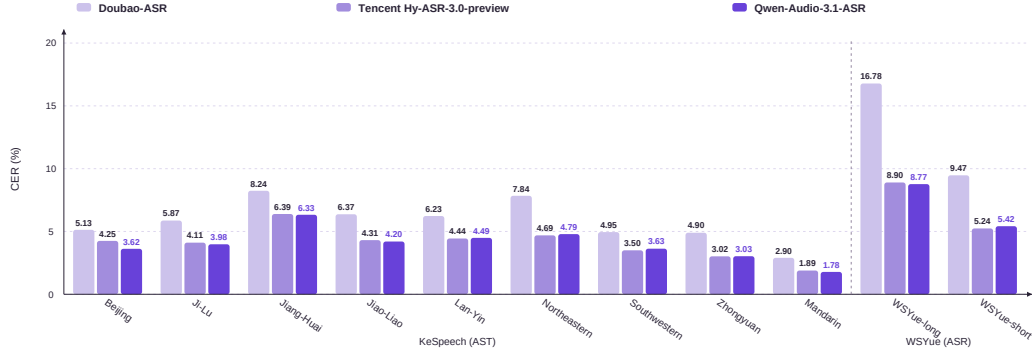


Figure 10: CER (%) on public dialect ASR and AST benchmarks, comprising nine KeSpeech AST subsets and two WSYue Cantonese ASR subsets. Lower is better.

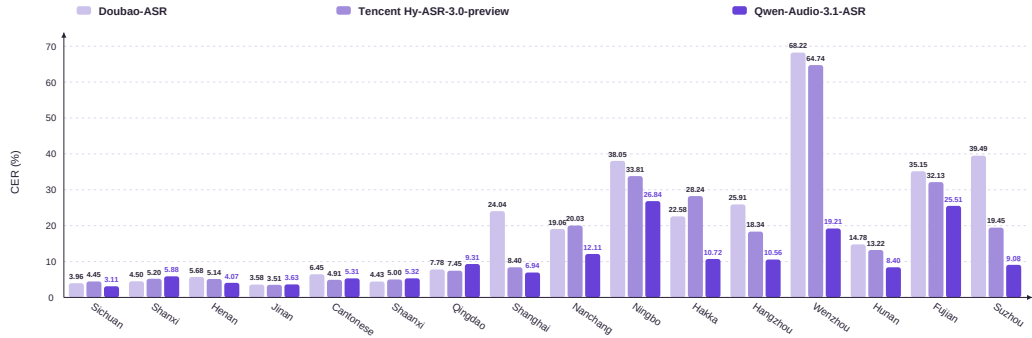


Figure 11: CER (%) on the internal ASR test set covering 16 Chinese dialects. All subsets use the ASR setting. Doubao-ASR does not explicitly list Henan, Qingdao, Nanchang, Ningbo, Hangzhou, Wenzhou, Hunan, or Suzhou in its publicly documented dialect coverage. Tencent Hy-ASR-3.0-preview does not explicitly list Shanxi, Shanghai, Ningbo, Hakka, Hangzhou, Wenzhou, Fujian, or Suzhou. Absence from these lists does not necessarily imply lack of recognition capability. Lower is better.

points, respectively. The largest improvement occurs on Wenzhou, where CER decreases to 19.21%, compared with 68.22% and 64.74% for the two baselines. Qwen-Audio-3.1-ASR also achieves the lowest CER on Shanghai, Nanchang, Sichuan, Ningbo, Hakka, Hangzhou, Henan, Hunan, Fujian, and Suzhou.

Internal dialect AST. For the internal AST evaluation, we report *semantic sentence accuracy*, defined as the percentage of outputs that preserve the reference meaning without errors in named entities or other key information. Qwen3.7-Max evaluates each model output independently against the reference text. To validate this automated evaluation, we manually annotate a subset of the evaluation data using the same criterion, achieving 95.35% agreement between human and Qwen3.7-Max judgments. Based on this high agreement, we use Qwen3.7-Max to score the full evaluation set.

An output is considered semantically incorrect if it contains an incorrect named entity, changes or omits key information such as a participant, action, object, time, location, quantity, negation, causal relation, or fact, or introduces punctuation or text-normalization errors that alter the meaning. Minor wording, grammatical, punctuation, or formatting differences are accepted when the core meaning and key information remain unchanged. Semantically equivalent dialectal and Mandarin expressions are also accepted: this metric assesses semantic fidelity, independently of whether the output conforms to the requested language form.

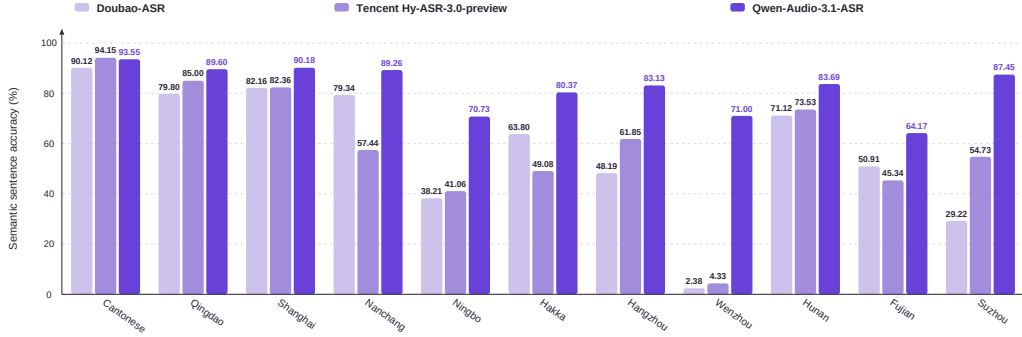


Figure 12: Semantic sentence accuracy (%) on the internal AST test set covering 11 Chinese dialects, with Mandarin translation as the target task. Qwen3.7-Max assesses whether each output preserves the reference meaning, including named entities and key information. Semantically equivalent expressions and minor differences that do not affect meaning are accepted. Higher is better.

Semantic sentence accuracy is computed as:

$$\text{Semantic Sentence Accuracy}(\%) = \frac{100}{N} \sum_{i=1}^N \mathbf{1}[z_i = 0]. \quad (7)$$

Here, N is the number of evaluated utterances, $z_i = 1$ indicates that the i -th output contains a semantic error under the criteria above, and $z_i = 0$ otherwise. The indicator function $\mathbf{1}[\cdot]$ equals 1 when its condition holds and 0 otherwise.

On the internal AST test set (Figure 12), we compare Qwen-Audio-3.1-ASR with each baseline on subsets corresponding to that baseline’s explicitly documented dialect coverage. Against Doubao-ASR, the comparison includes Shanghai, Hakka, and Cantonese. Qwen-Audio-3.1-ASR achieves higher semantic sentence accuracy on all three subsets, with a macro-average accuracy of 88.03%, compared with 78.69% for Doubao-ASR, an improvement of 9.34 percentage points. Against Tencent Hy-ASR-3.0-preview, the comparison includes Nanchang, Hunan, Cantonese, and Qingdao. Qwen-Audio-3.1-ASR achieves a macro-average accuracy of 89.03%, compared with 77.53% for Tencent Hy-ASR-3.0-preview, and performs better on three of the four subsets. The largest gain is observed on Nanchang, where accuracy increases from 57.44% to 89.26%. On Cantonese, Qwen-Audio-3.1-ASR achieves 93.55%, slightly below Tencent Hy-ASR-3.0-preview at 94.15%. These pairwise comparisons indicate competitive semantic preservation under our dialect-to-Mandarin AST evaluation setting.

5.3 Industry-Domain Entity Recognition

Reliable recognition of domain-specific terminology is essential for practical ASR applications. Financial and medical terms, software libraries, API names, organization names, and other specialized entities may be underrepresented in general-purpose training data, making them susceptible to substitution and omission errors. To improve coverage of these long-tail entities, we continually mine domain-specific vocabularies and construct corresponding ASR training examples using the entity-mining and data-synthesis pipeline described in Section 3.4.

Figure 13 compares Doubao-ASR, Tencent Hy-ASR-3.0-preview, and Qwen-Audio-3.1-ASR on an internal entity-recognition evaluation set spanning 15 industry domains. Qwen-Audio-3.1-ASR achieves the highest entity recall in nine domains and ties for the highest recall in Construction & Home and Manufacturing. Its recall ranges from 94.9% to 99.5% across the evaluated domains. The largest improvements over the stronger competing system are observed in Artificial Intelligence, Agriculture, and Commerce & Logistics, with gains of 2.3, 2.2, and 2.1 percentage points, respectively. Tencent Hy-ASR-3.0-preview retains higher recall in Software & IT, Energy & Chemicals, Finance & Business, and Lifestyle Services. Overall, these results indicate strong recognition of industry-specific entities across a diverse range of domains. The observed performance is consistent with the intended

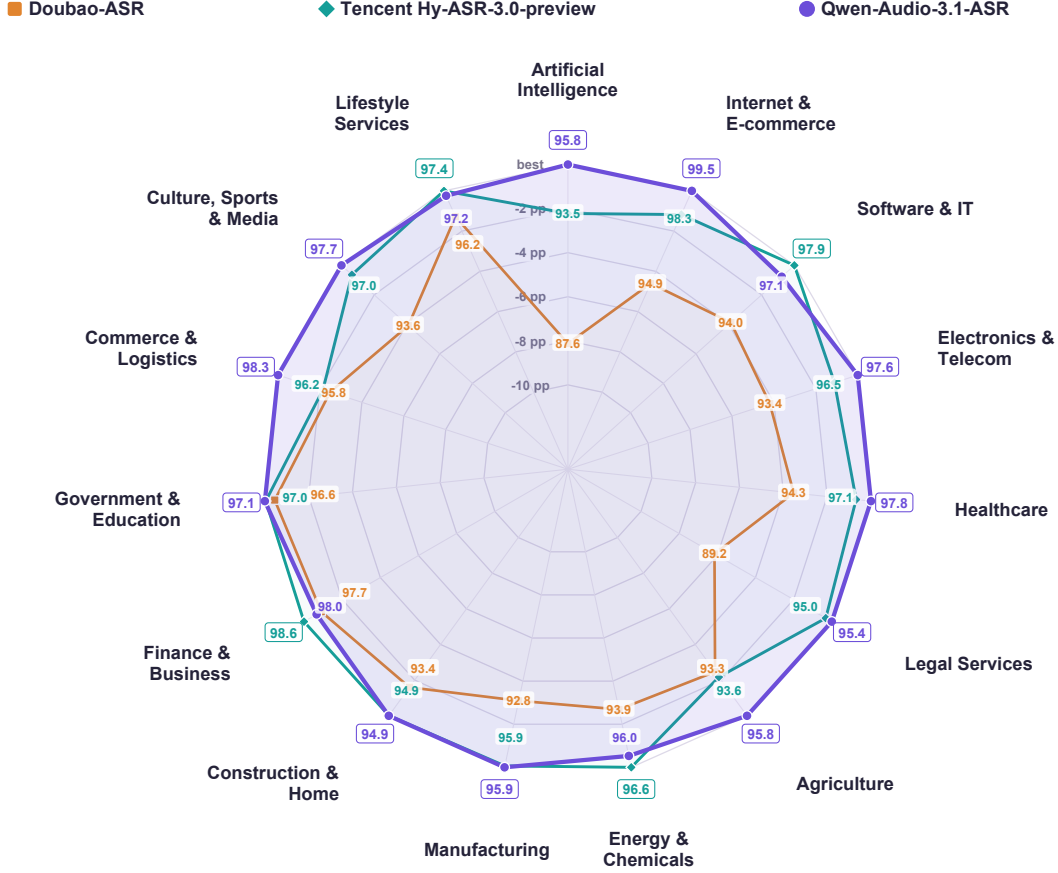


Figure 13: Entity recall (%) across 15 industry domains. Radial positions represent percentage-point gaps from the highest recall in each domain, with the outer ring indicating the domain-specific best result. Numerical labels show the original recall values. Higher recall indicates better performance.

benefits of the entity-mining and data-synthesis pipeline; however, this comparison does not isolate its contribution from other improvements incorporated into Qwen-Audio-3.1-ASR.

5.4 Hotword Customization

Qwen-Audio-3.1-ASR incorporates a hierarchical hotword customization mechanism that distinguishes between two priority tiers: a high-priority set (**P0**) and a broader candidate set (**P1**). During decoding, P0 candidates receive stronger conditioning than P1 candidates. This priority-aware design emphasizes high-confidence terms while retaining coverage of a broader set of potentially relevant entities. Table 4 compares hotword recall with and without hotword conditioning across six categories of long-tail entities. With conditioning enabled, Qwen-Audio-3.1-ASR achieves the highest recall among the evaluated systems in every P0 category, with recall ranging from 97.95% to 99.46%. These results demonstrate the effectiveness of the proposed mechanism in recognizing high-priority long-tail entities. For subject terms and trending buzzwords, hotword conditioning also improves P1 recall, although the gains are smaller than those for the corresponding P0 categories. This pattern is consistent with the weaker conditioning assigned to P1 candidates. However, Qwen-Audio-3.1-ASR does not achieve the highest conditioned recall in these two P1 categories, highlighting the distinction between prioritizing high-confidence terms and maximizing recall over a broader candidate set.

5.5 Long-Audio Context Modeling

Accurate transcription may require contextual information beyond the current audio segment, particularly when resolving homophones, technical terminology, and entity names. For example, a technical

Table 4: Hotword recall (%) by entity category and priority tier. Results are reported without (w/o) and with (w/) hotword conditioning. P0 denotes high-priority hotwords, and P1 denotes the broader candidate set. Higher values indicate better performance. Bold values denote the highest recall among systems with hotword conditioning in each row.

Category	Tier	Doubao-ASR		Fun-ASR-Flash		Qwen-Audio-3.1-ASR	
		w/o	w/	w/o	w/	w/o	w/
Person names	P0	32.27	82.64	37.33	95.26	62.12	99.43
Subject terms	P0	75.14	93.06	71.84	92.53	69.36	99.42
	P1	84.36	93.04	86.02	96.73	88.46	95.31
Trending buzzwords	P0	48.24	91.95	43.26	82.53	63.08	99.46
	P1	51.24	89.48	53.92	87.00	74.00	88.34
AI entities	P0	44.21	65.24	79.88	90.55	88.22	99.39
Brand names	P0	44.18	57.88	80.14	95.55	83.83	97.95
Product entities	P0	67.59	78.70	73.64	89.92	72.87	99.07

Table 5: Representative corrections enabled by long-audio context. The context cues occurred earlier in the same session. Colored spans highlight recognition errors and their context-conditioned corrections.

Earlier context cue	Reference	Without long context	With long context
xsltproc style.xml input.xml occurred earlier	style.xml data.xml > out.html	xsl data.xml > out.html	style.xml data.xml > out.html
Earlier explanation: “这叫法条竞合”	法条竞合	划掉即可	法条竞合
The term unigram occurred twice earlier	我们称之为unigram	我们称之为unique	我们称之为unigram
TCC was repeatedly established as the metric name	our TCC will be...	our GGC will be...	our TCC will be...
The name “洛军是阿占的儿子” occurred earlier	洛军是阿占的儿子	洛君是阿占的儿子	洛军是阿占的儿子
The technical term softmax occurred earlier	softmax 形式	softmax 形式	softmax 形式
The mode name debug occurred earlier	只能debug 模式	只能第八个 模式	只能debug 模式
The name “张贵生” occurred earlier	张贵生	张桂生	张贵生

filename may be confused with a phonetically similar common expression, while a person’s name may be difficult to identify from an isolated segment. Independently decoding each segment limits the use of preceding conversational information that could help disambiguate such cases.

Building on the contextual supervised fine-tuning described in Section 3.4, Qwen-Audio-3.1-ASR conditions the transcription of each audio segment on recently recognized text from the same session. This transcription history provides a dynamically updated source of context, enabling the model to incorporate information from preceding segments without requiring a manually maintained hotword list.

Table 5 presents representative examples of context-dependent corrections. Previously mentioned entities and technical terms, including codebase names, abbreviations, and personal names, provide contextual cues for recognizing subsequent occurrences. In these examples, conditioning on the transcription history improves recognition consistency across segments, whereas decoding the same audio without prior context produces phonetically plausible but incorrect alternatives.

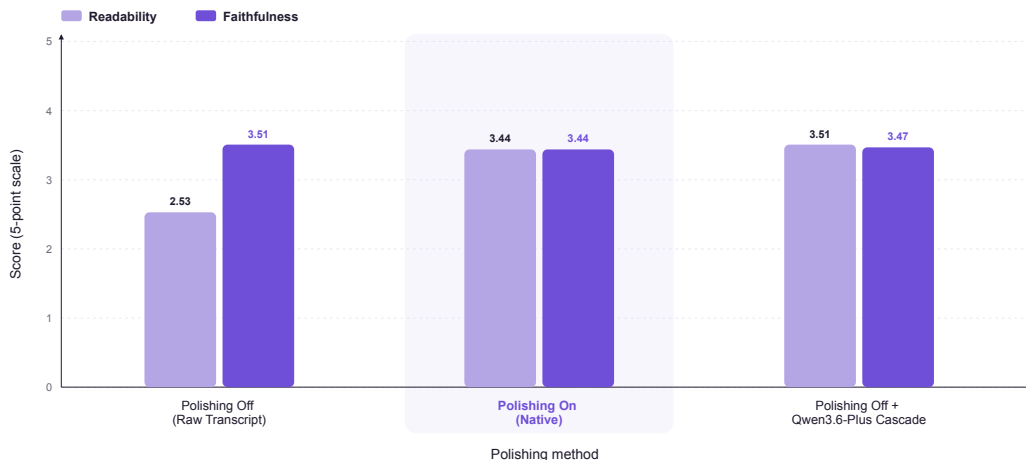


Figure 14: Readability and faithfulness of native polishing vs. the raw transcript and vs. an external ASR+LLM cascade.

5.6 Native Transcription Polishing

Conventional transcription pipelines typically handle fillers, disfluent repetitions, and self-corrections through a separate text-refinement stage, in which an LLM rewrites the ASR output. Although this approach separates recognition from transcript refinement, it requires an additional model invocation and introduces a sequential processing dependency before the polished transcript becomes available.

Qwen-Audio-3.1-ASR integrates transcription polishing directly into the recognition process. An instruction flag selects between raw transcription and polished output within a single decoding pass. When polishing is enabled, the model removes fillers and disfluent repetitions, incorporates the speaker’s final stated corrections, and standardizes punctuation and formatting without an additional model call. Figure 14 compares native polishing with raw transcription and a two-stage cascade in which the same raw ASR output is refined by Qwen3.6-Plus. On an internal five-point rating scale, native polishing improves readability from 2.53 to 3.44, compared with 3.51 for the cascade. Its faithfulness score is 3.44, close to the cascade’s 3.47 and slightly below the raw transcript’s 3.51. These results indicate that native polishing provides a substantial improvement in readability with a modest reduction in rated faithfulness, while eliminating the separate text-refinement invocation.

Figure 15 presents representative examples in Chinese and English. In the English examples, the model removes hesitation fillers such as *um* and *uh*, consolidates repeated words such as *the the the*, and reformulates false starts into more coherent expressions. For instance, the fragmented phrase *I just to put it / some comparison* is rendered as *Just to put it in some comparative context*. The model also standardizes capitalization, punctuation, and acronym formatting, converting the spoken letter sequence *d r r* into *DRR*. These examples illustrate how native polishing improves transcript fluency and readability while largely retaining the meaning conveyed in the original speech.

5.7 Streaming Display and Non-streaming Refresh in Message ASR

Message ASR combines streaming recognition for real-time transcript display with non-streaming recognition for refreshing the transcript at utterance end. We compare qwen-audio-3.1-asr-flash-message with Doubao Streaming Speech Recognition Model 2.0, abbreviated as Qwen Message and Doubao, respectively. We evaluate streaming latency and industry-term recall separately for the streaming and non-streaming outputs to distinguish the responsiveness and quality of the intermediate displayed text from the quality of the final refreshed transcript.

We measure display latency as the elapsed time between the end of the corresponding spoken content and its appearance in the displayed transcript. Doubao Streaming ASR 2.0 is evaluated through its API, while Qwen-Audio-3.1-ASR-Flash-Message is evaluated using a self-hosted service running on NVIDIA H100 GPUs. Both systems are tested with 10 concurrent audio streams. Network latency is assumed to be comparable between the two systems. As shown in Table 6, Qwen-Audio-3.1-ASR-

	Chinese Example	English Example
1 Filler Removal	<p>Before</p> <p>那个就是说 有没有 AI Logo 的设计工具，就是说那个就是说 AI 生成 Logo 的</p> <p>Polished</p> <p>有没有 <u>AI 生成 Logo</u> 的设计工具？</p>	<p>Before</p> <p>Um and this means that, um once, uh It it comes together with a voice recognition.</p> <p>Polished</p> <p><u>And this means that once</u> it comes together with a voice recognition.</p>
2 Repetition Reduction	<p>Before</p> <p>天天 天天 发 发发 抖音。</p> <p>Polished</p> <p><u>天天发</u> 抖音。</p>	<p>Before</p> <p>and uh not all the the the the stuff you can do with it, but the the the essential stuff is there</p> <p>Polished</p> <p>And not all <u>the stuff</u> you can do with it, but <u>the essential stuff</u> is there.</p>
3 Self-Correction	<p>Before</p> <p>搜一下 遛果香味的红富士苹果，啊不是，是 烟台栖霞红富士.....</p> <p>Polished</p> <p>搜一下 <u>烟台栖霞红富士苹果</u></p>	<p>Before</p> <p>Just to put it just to put it in some comparison, some comparative context.</p> <p>Polished</p> <p><u>Just to put it in some comparative context.</u></p>
4 Formatting	<p>Before</p> <p>要 三 斤装的，还带“脆”字 的 那个。</p> <p>Polished</p> <p>要 <u>3</u> 斤装的，还带“脆”字 <u>的</u> 那个。</p>	<p>Before</p> <p>with chasing ch-r share your daily plan as per your ch-r with your team also</p> <p>Polished</p> <p><u>With</u> chasing <u>DRR</u>, share your daily plan as per your <u>DRR</u> with your team also .</p>

Figure 15: Representative Chinese and English examples of native transcription polishing. Qwen-Audio-3.1-ASR removes fillers and disfluent repetitions, incorporates self-corrections, and standardizes sentence structure, punctuation, capitalization, and acronyms. The examples illustrate improvements in readability while largely retaining the original meaning.

Table 6: Streaming latency of Doubao Streaming ASR 2.0 and Qwen-Audio-3.1-ASR-Flash-Message. All values are in milliseconds; lower is better.

Metric	Doubao Streaming ASR 2.0	Qwen-Audio-3.1-ASR-Flash-Message
Text-emission latency	530	220
First-character latency	570	100
Last-character latency	307	103

Flash-Message achieves lower latency on all three reported metrics. Its first-character latency is 100 ms, compared with 570 ms for Doubao Streaming ASR 2.0, while text-emission latency decreases from 530 ms to 220 ms and last-character latency from 307 ms to 103 ms. These results indicate faster transcript display under the evaluated conditions, before the non-streaming refresh at utterance end.

For streaming recognition, Figure 16(a) shows that Qwen-Audio-3.1-ASR-Flash-Message achieves higher industry-term recall than Doubao Streaming ASR 2.0 on all nine domain-specific test sets. Its unweighted mean recall is 63.93%, compared with 46.28% for Doubao Streaming ASR 2.0, a difference of 17.64 percentage points. The largest gains occur in stocks (+35.74 points), animal husbandry (+27.16 points), and IT (+26.55 points), indicating improved recognition of domain-specific terms in the intermediate displayed transcripts.

For the non-streaming refresh, Figure 16(b) shows that Qwen-Audio-3.1-ASR-Flash-Message again achieves higher industry-term recall than Doubao Streaming ASR 2.0 on all nine sets, with an unweighted mean of 92.82%, compared with 78.27%, a difference of 14.54 percentage points. The largest gains occur in IT (+31.72 points), encyclopedia hotwords (+26.66 points), and AI terminology (+23.18 points). Both systems achieve higher recall after the non-streaming refresh than in their corresponding streaming outputs on every evaluated domain. Together, these results illustrate the

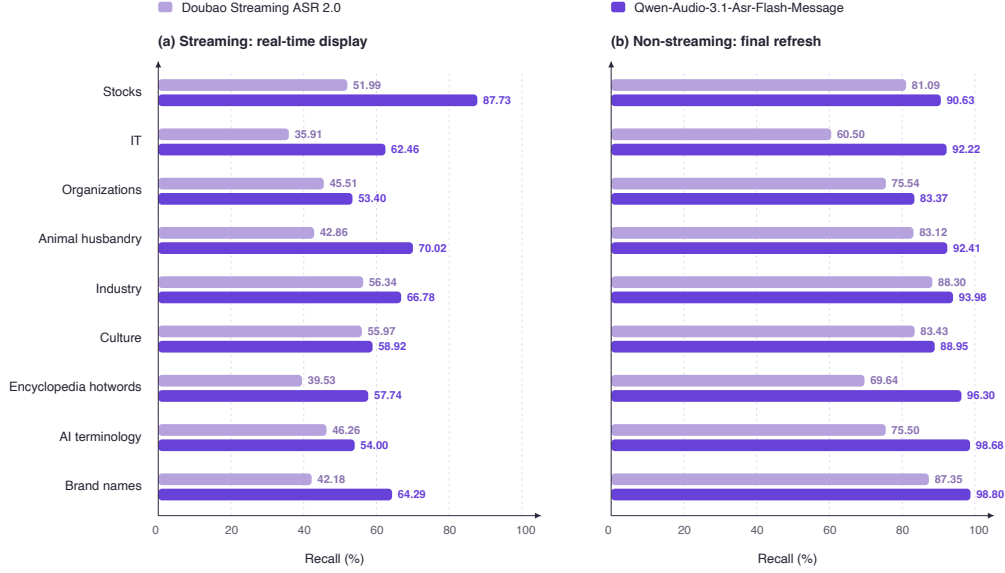


Figure 16: Industry-term recall on nine domain-specific test sets for (a) streaming outputs used for real-time display and (b) non-streaming outputs used for the final transcript refresh. Doubao Streaming ASR 2.0 abbreviates Doubao Streaming Speech Recognition Model 2.0. Each panel compares the two systems within the same output stage. Recall is reported as a percentage; higher is better.

Table 7: Speaker-diarized ASR results on four evaluation sets. To ensure an equitable comparison and mitigate inference instability common in LALMs during long-form audio processing, all test sets are standardized into approximately 4-minute audio chunks. The chunking methodology and evaluation scripts are provided in the [SDR Evaluation Toolkit](#). Fun-ASR is evaluated as a cascaded system. DER, WER, and cpWER are reported as percentages; lower is better. The best result in each column is shown in bold. Systems marked with an asterisk (*) use results taken directly from their respective technical reports.

System	AISHELL-4			AliMeeting-test			MLC-SLM			MagicData-RAMC		
	DER	WER	cpWER	DER	WER	cpWER	DER	WER	cpWER	DER	WER	cpWER
MOSS-Transcribe-Diarize (MOSI)	4.20	14.33	15.23	3.38	19.47	17.23	3.20	11.71	14.06	4.79	10.88	14.00
Muse-Voice-Transcribe-1.0 (Meta)	3.04	17.78	19.04	6.16	26.93	27.97	3.48	12.64	14.60	4.55	12.55	15.47
Gemini-3.5-Transcribe (Google)	13.99	24.38	36.39	21.84	33.18	45.16	12.71	14.47	23.05	11.44	15.29	25.32
Mai-Transcribe-2 (Microsoft)	5.96	14.92	22.25	8.52	23.83	27.65	7.01	12.26	15.63	5.42	11.99	14.98
Fun-ASR-API (Cascaded)	8.45	17.08	21.22	9.03	25.37	34.68	6.17	29.26	33.83	5.37	11.52	20.42
VibeVoice-ASR* (Microsoft)	-	-	24.99	-	-	29.33	-	-	14.81	-	-	-
SoulX-Transcriber* (Soul)	-	14.49	17.82	-	16.22	16.99	-	-	-	-	-	-
Qwen-Audio-3.1-ASR	3.26	13.02	15.02	2.11	17.23	15.78	1.80	9.89	12.35	3.08	10.21	13.32
Qwen-Audio-3.1-ASR-Next	2.70	12.03	13.51	2.15	13.89	12.04	1.78	10.32	12.60	3.18	10.50	13.58

complementary roles of the two stages: streaming recognition provides responsive intermediate text, while non-streaming recognition improves industry-term recall in the final refreshed transcript.

5.8 Speaker-diarized ASR

We evaluate joint speaker attribution and transcription on four test sets using DER and cpWER. Table 7 compares the two Qwen-Audio-3.1-ASR variants with a cascaded Fun-ASR baseline and three other systems.

The two Qwen-Audio-3.1-ASR variants collectively achieve the lowest error rates across all 12 dataset-metric combinations, covering DER, WER, and cpWER on four test sets. Qwen-Audio-3.1-ASR-Next achieves the lowest DER on AISHELL-4 and MLC-SLM, as well as the lowest WER

Table 8: General audio understanding and temporal grounding results for Qwen-Audio-3.1-ASR-Next and three baselines. The first seven benchmarks report accuracy (%): MMAU and MMSU assess broad understanding, while CochScene, Nonspeech7k, MELD, NSynth, and DiDiSpeech evaluate acoustic scenes, non-speech vocal sounds, emotion, musical instruments, and speaker gender, respectively. Clotho-Moment and DESED report temporal mean intersection over union (mIoU, %) for single-interval localization and multi-interval retrieval, respectively. Baseline results are obtained through our API-based evaluation. Higher is better, and the highest value in each row is boldfaced.

Benchmark	Gemini-3.1-Pro	Gemini-3.5-Flash	Seed-2.0-Lite	Qwen-Audio-3.1-ASR-Flash-Next
Comprehensive Audio Understanding (Accuracy)				
MMAU	80.90	77.20	78.40	81.60
MMSU	85.30	78.32	81.36	82.24
Specialized Audio Understanding (Accuracy)				
CochScene	63.90	55.07	49.82	76.30
Nonspeech7k	90.90	76.69	88.83	93.20
MELD	52.40	45.94	57.93	58.12
NSynth	39.20	16.31	29.79	66.90
DiDiSpeech	93.00	64.00	98.33	99.67
Temporal Grounding (mIoU)				
Clotho-Moment	52.56	44.24	71.79	82.20
DESED	49.37	37.86	66.64	66.60

and cpWER on AISHELL-4 and AliMeeting-test. Qwen-Audio-3.1-ASR achieves the lowest DER on AliMeeting-test, the lowest WER and cpWER on MLC-SLM, and the lowest values for all three metrics on MagicData-RAMC. The largest cpWER difference between the two variants occurs on AliMeeting-test, where Next achieves 12.04%, compared with 15.78% for Qwen-Audio-3.1-ASR, an absolute reduction of 3.74 percentage points. These results highlight complementary strengths across speaker diarization and transcription.

Both variants outperform the cascaded Fun-ASR baseline on all four test sets in both DER and cpWER. Next also achieves lower error rates than all non-Qwen systems across all eight metrics. These results demonstrate strong speaker attribution and transcription performance across the evaluated test sets, with the relative strengths of the two Qwen variants depending on the dataset and metric.

5.9 General Audio Understanding

We evaluate Qwen-Audio-3.1-ASR-Flash-Next on seven comprehensive and specialized audio understanding benchmarks measured by accuracy, and two temporal grounding benchmarks measured by mean Intersection over Union (mIoU). Table 8 compares its results with three competing systems.

Qwen-Audio-3.1-ASR-Flash-Next achieves the best performance on seven of the nine benchmarks. On the comprehensive audio understanding benchmarks, it obtains an accuracy of 81.60% on MMAU, which evaluates audio understanding and reasoning across speech, non-speech sounds, and music, outperforming Gemini-3.1-Pro by 0.70 percentage points. On MMSU, which assesses fine-grained spoken-language perception and reasoning, it achieves an accuracy of 82.24%, trailing the best result by 3.06 percentage points.

On the specialized audio understanding benchmarks, Qwen-Audio-3.1-ASR-Flash-Next achieves the best results across all five tasks: acoustic scene recognition (CochScene), non-speech vocal sound recognition (Nonspeech7k), emotion recognition (MELD), musical instrument recognition (NSynth), and speaker gender classification (DiDiSpeech). The improvements are particularly pronounced on CochScene and NSynth, with absolute margins of 12.40 and 27.70 percentage points over the respective next-best results.

For temporal grounding, Qwen-Audio-3.1-ASR-Flash-Next achieves 82.20% mIoU on Clotho-Moment, a benchmark for precise single-interval boundary localization, surpassing the next-best result by 10.41 percentage points. On DESED, which evaluates multi-interval retrieval of recurring

acoustic events, it achieves 66.60% mIoU, only 0.04 percentage points below the best result. Together, these results demonstrate robust fine-grained temporal grounding across complementary localization and retrieval settings.

6 Conclusion

In this report, we present Qwen-Audio-3.1-ASR, a family of instruction-controlled audio-language models that combines improved speech recognition with speaker-diarized transcription and general audio understanding. Building on a Qwen Mixture-of-Experts backbone and large-scale speech pretraining, the family supports recognition across 30 languages and 16 Chinese dialectal varieties, with user-selectable dialect transcription or translation into Mandarin. It retains production-oriented capabilities including industry-domain entity recognition, historical-context conditioning, hierarchical hotword customization, and native single-pass transcription polishing. The non-streaming Flash and Next variants extend transcription to structured outputs containing speaker labels, timestamps, and recognized text, including overlapping speech. Next additionally supports captioning, temporal grounding, sound event detection, and audio question answering across speech, environmental sounds, and music. Evaluations demonstrate competitive recognition performance across Chinese, English, multilingual, and dialectal benchmarks, together with strong results on speaker-diarized transcription and audio understanding tasks. Deployment variants accommodate asynchronous file transcription, low-latency streaming, and context-aware voice input, with capabilities available according to the selected variant. In particular, Message ASR combines streaming transcript display with an utterance-end full-context refresh, while integrating contextual conditioning, hierarchical hotwords, and native polishing without a separate text-rewriting service. Together, these advances extend the scope of ASR from lexical recognition to speaker-aware transcription and audio-grounded understanding while retaining configurable workflows for practical deployment.

7 Authors (in Alphabetical Order of Last Name)

- | | | |
|------------------|-------------------|------------------|
| • Chuanmeng Bian | • Zengxi Li | • Tianzi Wang |
| • Daren Chen | • Zheng Li | • Wen Wang |
| • Peixin Chen | • Chengdong Liang | • Wupeng Wang |
| • Zhigao Chen | • Baiji Liu | • Cheng Wen |
| • Chong Deng | • Yaming Liu | • Yuzhong Wu |
| • Zhiyun Fan | • Ying Liu | • Zijian Xia |
| • Zhifu Gao | • Bin Ma | • Yunchong Xiao |
| • Bo Gong | • Yiping Peng | • Nan Yang |
| • Qing Gu | • Yue Zhang Peng | • Jianwei Yu |
| • Jiajun He | • Zhendong Peng | • Jixing Yu |
| • Yawei Hu | • Yu Pu | • Binbin Zhang |
| • Yunjie Ji | • Yang Shi | • Lei Zhang |
| • Wei Ju | • Xin Shu | • Sitong Zhao |
| • Jingbei Li | • Jian Tang | • Guangdong Zhou |
| • Xiangang Li | • Biao Tian | • Yuan Zhou |
| • Xu Li | • Peiyao Wang | • Jianheng Zhuo |

8 Acknowledgments

We are immensely grateful for the invaluable discussions, support, and assistance provided by many colleagues throughout the development of Qwen-Audio-3.1-ASR. Special thanks go to Qian Chen, Han Zhao, Chaohong Tan, Cheng Wen, Qinglin Zhang, Xiang Lv, Wei Ju, Luyao Cheng, Haoxu Wang, Yongchao He, Chen Ding, Bangduo Chen, Siyu Wang, Zelan Yang, Changwei Li, and Shengkui Zhao.

References

- [1] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proc. ICML*,

- volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.
- [2] Jiajun He, Naoki Sawada, Koichi Miyazaki, and Tomoki Toda. CMT-LLM: Contextual multi-talker ASR utilizing large language models. In *Proc. Interspeech*, 2025. URL https://www.isca-archive.org/interspeech_2025/he25_interspeech.html.
 - [3] Zhiliang Peng, Jianwei Yu, Yaoyao Chang, et al. VIBEVOICE-ASR technical report, 2026. URL <https://arxiv.org/abs/2601.18184>.
 - [4] S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. MMAU: A massive multi-task audio understanding and reasoning benchmark, 2024. URL <https://arxiv.org/abs/2410.19168>.
 - [5] Dingdong Wang, Junan Li, Jincenzi Wu, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. MMSU: A massive multi-task spoken language understanding and reasoning benchmark, 2025. URL <https://arxiv.org/abs/2506.04779>.
 - [6] Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-ASR technical report, 2026. URL <https://arxiv.org/abs/2601.21337>.
 - [7] Jin Xu, Zhifang Guo, Hangrui Hu, et al. Qwen3-Omni technical report, 2025. URL <https://arxiv.org/abs/2509.17765>.
 - [8] Chen Yang, Chufan Yu, Hanfu Chen, et al. MOSS-Audio technical report, 2026. URL <https://arxiv.org/abs/2606.01802>.
 - [9] Feiyu Shen, Fenglong Xie, Junjie Li, Kun Xie, Lei Xie, Xu Tang, Xuelong Geng, Yan Jia, Yao Hu, Yichen Han, Yichen Wu, Ziqi Dai, Junjie Chen, Kai Huang, Manzhen Wei, and Yixuan Li. FireRedAudio: A general-purpose audio language model with decoupled continuous representations for understanding and generation, 2026. URL <https://arxiv.org/abs/2608.24168>.
 - [10] MOSI.AI. MOSS Transcribe Diarize technical report, 2026. URL <https://arxiv.org/abs/2601.01554>.
 - [11] Chuanmeng Bian, Daren Chen, Peixin Chen, Zhigao Chen, Zhiyun Fan, Zhifu Gao, Bo Gong, Qing Gu, Jiajun He, Yawei Hu, Yunjie Ji, Jingbei Li, Xiangang Li, Xu Li, Zengxi Li, Zheng Li, Chengdong Liang, Baiji Liu, Ying Liu, Bin Ma, Yiping Peng, Yuezhong Peng, Zhendong Peng, Yu Pu, Yang Shi, Xin Shu, Jian Tang, Biao Tian, Peiyao Wang, Tianzi Wang, Wen Wang, Wupeng Wang, Cheng Wen, Yuzhong Wu, Zijian Xia, Yunchong Xiao, Nan Yang, Jianwei Yu, Jixing Yu, Binbin Zhang, Lei Zhang, Sitong Zhao, Guangdong Zhou, Yuan Zhou, and Jianheng Zhuo. Qwen-Audio-3.0-ASR Technical Report, 2026. URL <https://arxiv.org/abs/2609.07549>.
 - [12] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *Proc. ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 3915–3924. PMLR, 2022. URL <https://proceedings.mlr.press/v162/chiu22a.html>.
 - [13] Keyu An, Shiliang Zhang, and Zhijie Yan. Are Transformers in Pre-trained LM A Good ASR Encoder? An Empirical Study. *arXiv preprint*, abs/2409.17750, 2024. doi: 10.48550/arXiv.2409.17750. URL <https://doi.org/10.48550/arXiv.2409.17750>.
 - [14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang

- Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint*, abs/2505.09388, 2025. doi: 10.48550/arXiv.2505.09388. URL <https://doi.org/10.48550/arXiv.2505.09388>.
- [15] Keyu An, Zerui Li, Zhifu Gao, and Shiliang Zhang. Paraformer-v2: An improved non-autoregressive transformer for noise-robust speech recognition. *arXiv preprint*, abs/2409.17746, 2024. doi: 10.48550/arXiv.2409.17746. URL <https://doi.org/10.48550/arXiv.2409.17746>.
 - [16] Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, Shengpeng Ji, Yabin Li, Zerui Li, Heng Lu, Haoneng Luo, Xiang Lv, Bin Ma, Ziyang Ma, Chongjia Ni, Changhe Song, Jiaqi Shi, Xian Shi, Hao Wang, Wen Wang, Yuxuan Wang, Zhangyu Xiao, Zhijie Yan, Yexin Yang, Bin Zhang, Qinglin Zhang, Shiliang Zhang, Nan Zhao, and Siqi Zheng. FunAudioLLM: Voice Understanding and Generation Foundation Models for Natural Interaction Between Humans and LLMs. *arXiv preprint*, abs/2407.04051, 2024. doi: 10.48550/arXiv.2407.04051. URL <https://doi.org/10.48550/arXiv.2407.04051>.
 - [17] Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, Guanrou Yang, Yabin Li, Yanni Chen, Zhifu Gao, Qian Chen, Yue Gu, Mengzhe Chen, Yafeng Chen, Shiliang Zhang, Wen Wang, and Jieping Ye. CosyVoice 3: Towards In-the-wild Speech Generation via Scaling-up and Post-training. *arXiv preprint*, abs/2505.17589, 2025. doi: 10.48550/ARXIV.2505.17589. URL <https://doi.org/10.48550/arXiv.2505.17589>.
 - [18] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proc. EuroSys*, pages 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL <https://doi.org/10.1145/3689031.3696075>.
 - [19] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
 - [20] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. AISHELL-1: An Open-Source Mandarin Speech Corpus and a Speech Recognition Baseline. In *Proc. O-COCOSDA*, pages 1–5, 2017. doi: 10.1109/ICSDA.2017.8384449. URL <https://doi.org/10.1109/ICSDA.2017.8384449>.
 - [21] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. AISHELL-2: Transforming Mandarin ASR Research Into Industrial Scale, 2018. URL <https://arxiv.org/abs/1808.10583>.
 - [22] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. FLEURS: Few-Shot Learning Evaluation of Universal Representations of Speech. In *Proc. SLT*, pages 798–805, 2023. doi: 10.1109/SLT54892.2023.10023141. URL <https://doi.org/10.1109/SLT54892.2023.10023141>.
 - [23] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR Corpus Based on Public Domain Audio Books. In *Proc. ICASSP*, pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.
 - [24] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. WeNetSpeech: A 10000+ Hours Multi-Domain Mandarin Corpus for Speech Recognition. In *Proc. ICASSP*, pages 6182–6186, 2022. doi: 10.1109/ICASSP43922.2022.9746682. URL <https://doi.org/10.1109/ICASSP43922.2022.9746682>.
 - [25] Yujie Tu, Yifan Yang, Tianrui Wang, Yanqiao Zhu, Guodong Lin, Mingchen Shao, Haoran Wang, Junzhe Liu, Yuxiang Fu, Yizhou Peng, et al. GigaSpeechBench: A Real-World Multilingual Speech-to-Text Benchmark. *arXiv preprint*, 2026. doi: 10.48550/arXiv.2606.28884.

- [26] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common Voice: A Massively-Multilingual Speech Corpus. In *Proc. LREC*, pages 4218–4222, Marseille, France, 2020. European Language Resources Association. URL <https://aclanthology.org/2020.lrec-1.520/>.