AmphionASR: Personalized Context-Aware Speech Recognition

Amphion Team

Abstract

Practical speech recognition often requires more than unconditioned transcription: users may provide domain vocabulary or reference speech, while the input may contain degradations or whispered speech. We present **AmphionASR**, a 1.7 B SpeechLLM that supports all of these conditions within a single unified model driven by task-specific inputs. We evaluate four conditioned recognition settings on Mandarin and English benchmarks. For large-vocabulary hotword conditioning, the fine-grained retriever attains Recall@50 above 94 %, and the complete pipeline reduces average entity error by 37–55 % relative to no hotword context [1]. Online retrieval adds a paired median of 1.37–3.25 ms relative to precomputed Top-50 prompts in the measured single-request, steady-state setting. The unified model also demonstrates target-speaker selection when the requested speaker is present, records the lowest WER among the compared systems on eight of sixteen Voice-in-the-Wild degradation subsets [2], and obtains the lowest error on both the Mandarin wEar and English WTIMIT whispered-speech tests (0.58 % CER and 6.11 % WER, respectively) [3].

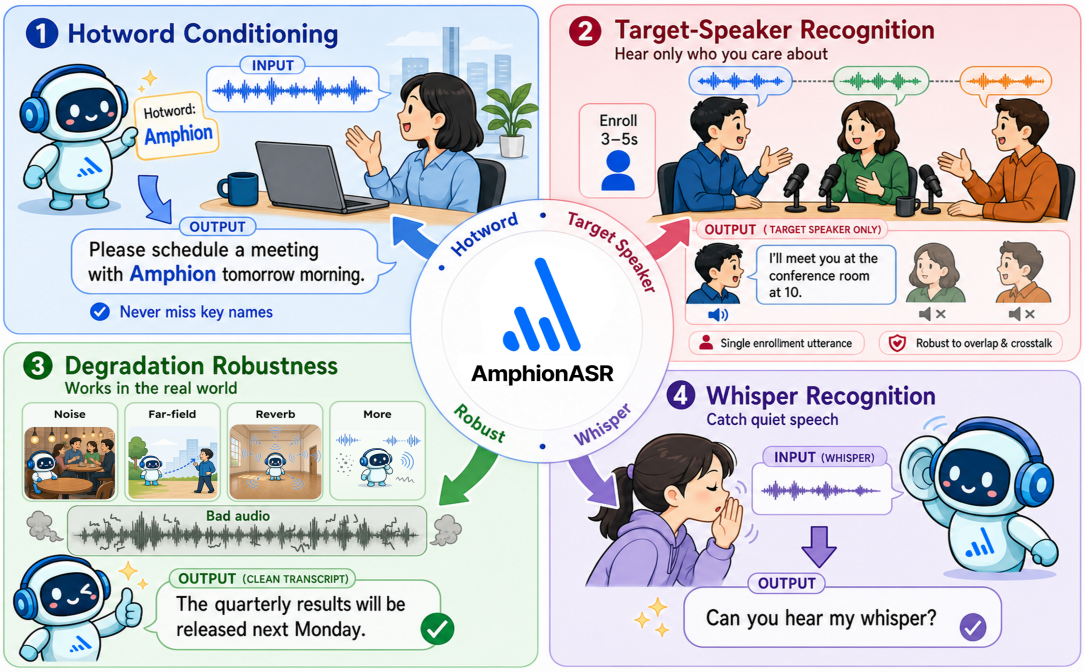

Figure 1: Four recognition capabilities of AmphionASR. *Top-left:* hotword conditioning with retrieval-based contextual biasing, prioritizing key names in the transcript. *Top-right:* target-speaker ASR that transcribes only the 3–5 s enrolled speaker from a multi-speaker mixture, robust to overlap and crosstalk. *Bottom-left:* degradation-robust ASR that remains accurate under various real-world audio degradations. *Bottom-right:* whispered-speech ASR that transcribes low-amplitude whispered input.

1 Introduction

Modern speech recognition has evolved through several stages. Hybrid HMM–DNN systems [4] dominated for decades by combining separate acoustic, lexicon, and language models, but relied on hand-engineered components and a multi-stage decoding pipeline. End-to-end neural models replaced this pipeline with a single sequence-to-sequence mapping trained with connectionist temporal classification, attention encoder–decoder networks, and RNN-transducers [5, 6, 7], with the Conformer [8] becoming a strong hybrid of convolution and self-attention. Self-supervised pretraining then reduced reliance on labeled audio by learning speech representations from unlabeled data [9, 10], and weakly supervised learning at scale produced robust multilingual systems such as Whisper [11]. Most recently, SpeechLLMs pair audio encoders with autoregressive language backbones [12, 13, 14, 15, 16], inheriting the comprehension and instruction-following ability of large language models.

General transcription is only one part of practical ASR. A recognizer may also face four recognition settings beyond unconditioned transcription:

- **Hotword-conditioned ASR** addresses the recognition difficulty of rare entities and domain-specific terminology—proper nouns, product names, and professional terms that an unconditioned model frequently mis-transcribes—by accepting a user-supplied hotword list as contextual bias.
- **Target-speaker ASR** addresses multi-talker overlap: given a short enrollment utterance of a target speaker, the model must transcribe only that speaker's content from overlapping speech, rather than concatenating all talkers.
- **Degradation-robust ASR** handles real-world capture conditions such as far-field pickup, additive noise, reverberation, and transmission dropout that degrade captured speech.
- **Whispered-speech ASR** transcribes atypical, low-energy whispered phonation that deviates from normal speech.

These settings differ in their side information and acoustic conditions, but they share the same desired output: the transcript of the requested speech.

We present **AmphionASR**, a unified model for these four recognition settings. Hotword lists and target-speaker enrollment audio enter as task conditions, while degradation and whispered-speech examples adapt the model to atypical acoustic inputs. This formulation separates the recognition backbone from the information that specifies what and under which conditions to transcribe. The hotword path follows prior work that already combines GLCLAP top-K retrieval, prompt injection, and GRPO [17]; our scope is integration into the unified model and evaluation across the recognition settings studied here.

Our contributions are:

- **A unified interface for conditioned ASR.** We formulate hotword-conditioned, target-speaker, degradation-robust, and whispered-speech ASR within one model, using task instructions and optional reference inputs.
- **An integrated large-vocabulary hotword-conditioning pipeline.** We use a fine-grained dual encoder to retrieve a top-K subset from a larger vocabulary before decoding (Recall@50 above 94 % on CommonVoice) and inject the selected terms into the prompt as contextual bias. On GigaSpeechBench, the complete pipeline reduces average entity error by 55 % in Mandarin and 37 % in English relative to no hotword context [1]. In the measured single-request, steady-state setting, online retrieval adds median overheads of 1.37 ms in English and 3.25 ms in Mandarin relative to precomputed Top-50 prompts.
- **A multi-condition evaluation.** We evaluate the unified model on contextual ASR, target-speaker

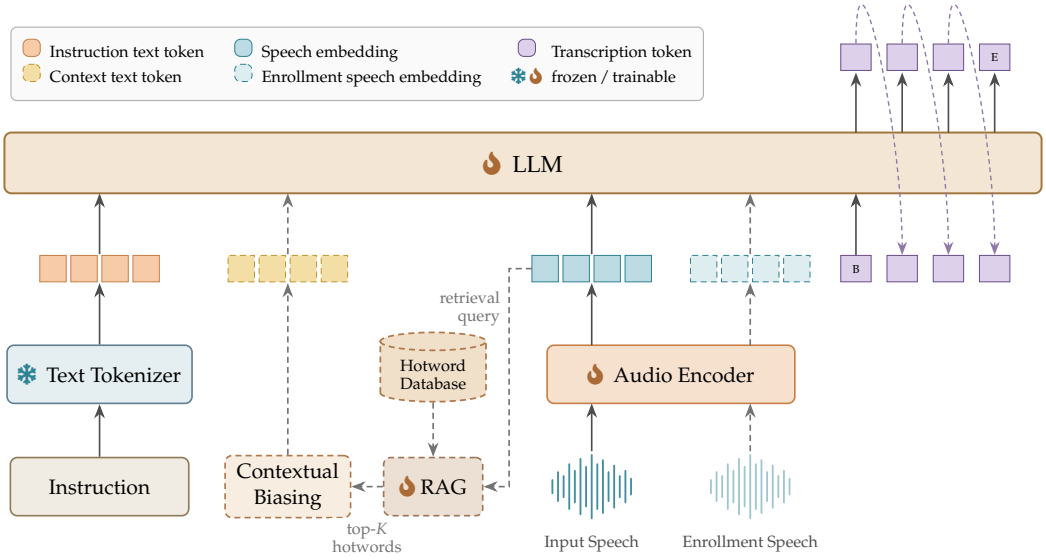

Figure 2: AmphionASR architecture. A 16 kHz waveform is encoded by the **audio encoder** into the LLM's embedding space, while the text branch tokenizes the instruction; an optional enrollment utterance passes through the same encoder for target-speaker recognition. The LLM consumes the concatenated prefix and decodes the transcription autoregressively, each emitted token being fed back as the input of the next step. A parallel **retrieval** branch queries a hotword database with the encoded speech, selects a top-K hotword subset from a large candidate vocabulary and inserts it into the sequence as contextual biasing tokens; the retrieval adapters and the LLM are updated at fine-tuning time, and the audio encoder is also updated in Stage 1 only and frozen thereafter. Dashed boxes and arrows mark the paths that are optional at inference time.

ASR, acoustic degradations, and whispered speech, reporting both improvements and remaining gaps across the evaluated conditions.

2 AmphionASR Architecture

2.1 Overview

AmphionASR couples three components, shown in Figure 2: an audio encoder that feeds frame embeddings directly into the LLM, a text branch that carries the instruction and the optional task conditions, and a retrieval branch that supplies hotword context. At inference time, the hotword retriever selects a top-K subset from a large candidate vocabulary and inserts it into the prompt before decoding. The audio encoder and the LLM are initialized from Qwen3-ASR-1.7B [16]; the retrieval adapters are trained from scratch.

Following the convention for LLM-based systems, we report the model scale by its LLM: AmphionASR is a 1.7 B model. Component-level parameter counts are summarized in Table 1.

2.2 Audio Encoder

The acoustic front-end is an audio encoder initialized with the Qwen3-ASR-1.7B AuT encoder [16]: a single stack that maps 16 kHz mono audio through a log-mel front-end (128 mel bins, 10 ms hop), a

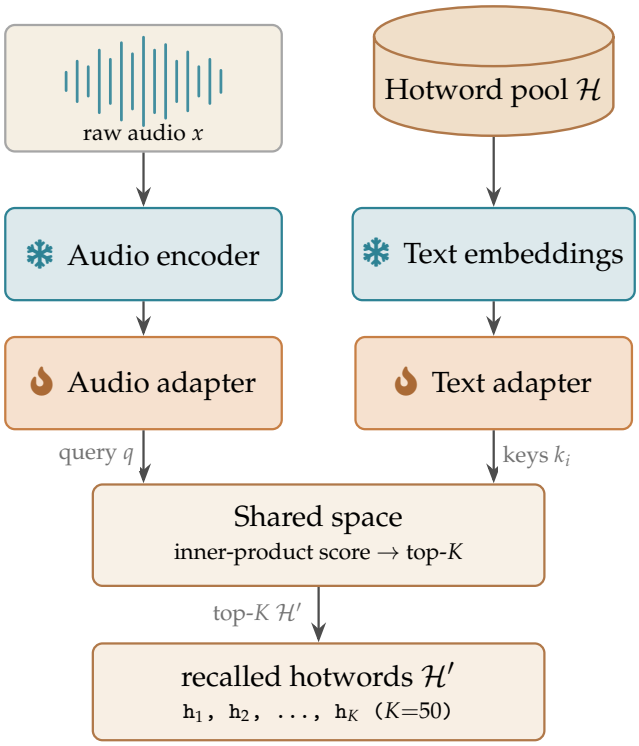

Figure 3: Retrieval-based contextual biasing paradigm (AmphionASR). Frozen acoustic and text features pass through two small trainable adapters into a shared embedding space; inner-product scores rank the global hotword pool $\mathcal{H}$ and the top-K subset $\mathcal{H}'$ is inserted into the hotword prompt line before a *single* ASR forward pass.

Table 1: Per-component parameter counts.

Component	Parameters	Detail
Audio encoder	300 M	initialized from Qwen3-ASR-1.7B AuT encoder
LLM	1.7 B	initialized from Qwen3-ASR-1.7B language backbone
RAG adapters	2.6 M	two MLP adapters
Total	2.0 B	audio encoder + LLM + RAG adapters

convolutional sub-sampler and a 24-layer Whisper-style pre-LN Transformer into 2048-dimensional frame embeddings at 12.5 Hz. A chunked-inference window of 800 frames extends the effective input duration beyond the standard 30 s ceiling. The audio encoder is updated only in Stage 1 of fine-tuning and frozen thereafter; Stages 2–3 update only the LLM.

2.3 Language Model

The LLM is initialized from the Qwen3-ASR-1.7B text backbone [16], a 28-layer decoder-only Transformer with 2048-dimensional hidden states and a 64 K native context length. The LLM is updated during fine-tuning as described in Section 4.

Table 2: Task prompt templates used by AmphionASR. `{speech}` denotes the audio placeholder that the merge step fills with the encoder-output embedding span; `{enroll}` denotes the enrollment utterance.

Task	Prompt template
General ASR	`Transcribe the following audio.{speech}`
Hotword ASR	`Transcribe the following audio.[Hotwords:...].{speech}`
TS-ASR	`Given the speaker's voice:{enroll} Transcribe what this speaker says in the following audio.{speech}`
Whispered-speech ASR	`Transcribe the following whispered audio.{speech}`

2.4 Hotword Retrieval

LLM-based ASR can use a prompt-level hotword list to bias recognition towards rare or domain-specific entities. Directly inserting a large candidate vocabulary can exceed the context budget and exposes the decoder to many irrelevant terms. We therefore retrieve a compact, utterance-relevant subset before decoding.

CLAP [18] establishes the contrastive language-audio recipe: two encoders align audio and text in a shared embedding space, and an audio–text inner product scores relevance. Applied to hotword retrieval, pooling the whole utterance into one audio token yields a global query without explicit temporal locality. This representation can obscure hotwords that occupy only short spans of a longer utterance.

Following GLCLAP [19], AmphionASR retrieves hotwords with a fine-grained local scoring module (Figure 3) that reuses the frozen audio encoder and the frozen LLM text embedding table. Two small trainable MLP adapters map these frozen representations into a shared 512-dimensional space; the audio side is kept at the frame level (no utterance-level pooling), and each candidate hotword is scored against every frame,

$$\begin{aligned} \mathbf{a}_t &= g_{\text{audio}}\big(\text{AuT}(x)_t\big), \quad \mathbf{k}_h = g_{\text{text}}\big(\mathbf{E}_{\text{tok}}(h)\big), \\ s_h &= \max_t \mathbf{a}_t^\top \mathbf{k}_h, \quad \mathcal{H}' = \text{TopK}_{h\in\mathcal{H}}\, s_h. \end{aligned} \tag{1}$$

where g_{audio} and g_{text} are the two adapters, $\mathbf{a}_t$ is the per-frame audio embedding, $\mathbf{E}_{\text{tok}}$ is the LLM token embedding table, and $\mathcal{H}'$ is the retrieved top-K subset. The max-over-time reduction s_h retains the strongest frame-level match for each entity, while top-K selection bounds the number of prompt terms. The selected hotwords are inserted into the prompt line before a *single* ASR forward pass, with no second transcription [20].

This retrieval-and-prompt pattern follows recent contextual-ASR work. In particular, Kong et al. [17] already combine GLCLAP top-K retrieval, prompt injection, and GRPO. Our contribution boundary is their integration into the shared AmphionASR model and its broader set of evaluated recognition settings, rather than a new retrieval or reinforcement-learning principle.

2.5 Instruction Template

Each task is represented by a short instruction followed by the audio placeholder (Table 2). All tasks share the same model parameters and differ only in their instructions and optional reference inputs. Whispered-speech ASR changes the instruction, while hotword-conditioned and target-speaker ASR add task-specific fields.

Target-speaker conditioning. TS-ASR prepends an enrollment audio block before the main audio and switches the instruction to "*Transcribe what this speaker says in the following audio.*". Both audio

segments share the same audio encoder path and reach the LLM as two ordered audio-embedding spans (enrollment first, mixed audio second).

Hotword conditioning. Hotword conditioning is applied through the prompt: the retrieved hotwords are inserted as a single comma-separated hotword line after the instruction, biasing the decoder toward the listed entities without any architectural change. Hotword SFT exposes the model to the retrieved list as a contextual-bias signal for rare entities and domain-specific terms. The list carries no dedicated delimiter; the `Hotwords:` literal is therefore part of the prompt grammar presented during training.

3 Data

AmphionASR is trained through a three-stage LoRA supervised fine-tuning (SFT) recipe over eight curated training sets (Table 3), totalling 3.74 million utterances built from eleven public corpora (Table 4). The sets map directly onto the recognition settings introduced in Section 1: whispered-speech ASR, degradation-robust ASR, hotword-conditioned ASR, target-speaker ASR (TS-ASR), and general transcription data included to reduce the risk of catastrophic forgetting. The eight sets are distributed across the three SFT stages: degradation and whisper data in Stage 1, hotword and target-speaker data in Stage 2, and a replay of the Stage 1–2 sets together with general-transcription corpora in Stage 3. Section 4 documents the per-stage optimizer settings.

3.1 Whispered-Speech ASR

The WhispEar training set covers the full bilingual training pool (222,440 utterances): 199,805 English whisper clips from the Emilia-based subset, 18,588 English whisper clips from the WTIMIT subset, and 4,047 Mandarin whisper clips from the wEar subset [3]. Section 5.6 reports evaluation on the wEar and WTIMIT whisper test sets.

3.2 Degradation-Robust ASR

The Voice-in-the-Wild-2M training set provides 556,421 single-utterance examples covering far-field capture, additive noise, reverberation, clipping, dropout and their superpositions [2].

3.3 Hotword-Conditioned ASR

Hotword training data are materialized offline from five public ASR corpora whose transcripts carry mined entity lists (Table 5). At conversion time each utterance receives a prompt-level hotword line with probability 0.8; the line combines real hotwords with hard-negative and random distractors, with the total list size controlled in $[10, 50]$ (Section 3.6). The resulting training set contains only single-audio examples.

Two CommonVoice hotword *test* splits (ZH and EN) with the same annotation protocol are reserved for evaluation (Table 6).

3.4 General Transcription Retention

To limit catastrophic forgetting of general transcription behavior, the training mixture retains exposure to two large single-speaker corpora: the full AISHELL-2 training split (1,009,223 utterances) and LibriSpeech train-clean-100/360 plus train-other-500 (281,241 utterances).

Table 3: Training sets used across the AmphionASR three-stage SFT recipe. "Utterances" counts training examples after offline preparation; TS-ASR rows contain two audio inputs (enrollment + mixture) per example.

Training set	Task	Utterances	Notes
WhispEar [3]	Whispered-speech ASR	222 440	bilingual whisper corpus
Voice-in-the-Wild-2M [2]	Degradation	556 421	real-world degradations
Five-corpus hotword mixture	Hotword ASR	1 043 317	single-audio
AISHELL-2 [21]	General ASR	1 009 223	Mandarin retention
LibriSpeech [22]	General ASR	281 241	English retention
CommonVoice EN target-speaker	TS-ASR	184 747	synthesized
MagicData target-speaker	TS-ASR	106 116	synthesized
Multi-corpus target-speaker ASR	TS-ASR	333 640	synthesized
Total		**3 737 145**	

Table 4: Public corpora that appear in the training mixture. A corpus may contribute to multiple training sets (e.g. CommonVoice EN feeds both hotword-conditioned and target-speaker training); utterance counts are *not* additive across rows.

Corpus	Role in SFT
WhispEar [3]	Whispered-speech ASR (222 k utterances)
Voice-in-the-Wild-2M [2]	Degradation-robust ASR (556 k)
CommonVoice EN (cleaned) [23]	Hotword-conditioned ASR (500 k cap) + target-speaker syntheses
MagicData Mandarin [24]	Hotword-conditioned ASR (350 k cap) + target-speaker syntheses
AISHELL-1 [25]	Hotword-conditioned ASR
AISHELL-2 [21]	General Mandarin ASR retention (1.01 M)
AISHELL-3 [26]	Hotword-conditioned ASR + target-speaker synthesis
LibriSpeech [22]	General English ASR retention (train-clean-100/360 + train-other-500)
MLS English [27]	Target-speaker synthesis
THCHS-30 [28]	Hotword-conditioned ASR + target-speaker synthesis
aidatatang_200zh	Target-speaker synthesis

3.5 TS-ASR Synthesis

Target-speaker training examples are produced by an offline synthesizer. From single-speaker source manifests, each output sample contains a target waveform mixed with K interferers, an enrollment utterance from the same speaker but *a different recording*, and the target's transcript; a fraction of *anti-hallucination negatives*—empty-transcript samples whose target speaker is removed from the mixture—are intended to expose the model to cases where the requested speaker is absent. The mixing equation, RIR/noise pools, and negative sub-types are detailed in Appendix C.

Target-speaker training sets. Stage 2 of the SFT recipe includes three synthesized target-speaker partitions totalling 624,503 examples (Table 7). The first two are built from single-corpus manifests (CommonVoice EN and MagicData respectively); the third draws from six public corpora. Positive counts by underlying corpus in the multi-corpus partition are: CommonVoice EN (225,638), MLS

Table 5: Public corpora in the hotword-conditioned training set. "Cap" is the utterance cap applied during offline preparation.

Source	Cap	Utterances
CommonVoice EN (cleaned) [23]	500 k	500 000
MagicData Mandarin [24]	350 k	349 958
AISHELL-1 [25]	–	120 098
AISHELL-3 [26]	–	63 261
THCHS-30 [28]	–	10 000
Total		1 043 317

Table 6: Hotword evaluation splits (not used in SFT).

Split	Underlying audio
CommonVoice ZH hotwords	CommonVoice ZH test
CommonVoice EN hotwords	CommonVoice EN test

Table 7: Target-speaker training sets in the training mixture. "Pos." counts positive (target-speaker-present) examples; "Neg." counts anti-hallucination negatives with empty transcripts.

Set	Primary source	Pos.	Neg.
CommonVoice EN	CommonVoice EN	175 510	9 237
MagicData	MagicData	100 810	5 306
Multi-corpus	6 corpora	281 989	51 651
Total		558 309	66 194

English (40,184), MagicData (10,119), aidatatang_200zh (4,140), AISHELL-3 (1,425), and THCHS-30 (483); negatives account for the remaining 51,651 examples ($\approx$15 %).

Section 5.4 reports evaluation on an in-house test set generated by the same pipeline with a held-out random seed.

3.6 Hotword Pool Construction at Training Time

For hotword-bearing training sets, training-time hotword lists are assembled per sample so that the total number of real hotwords and distractors is controlled in $[10, 50]$. Each list is built in three stages:

1. **Real terms.** Each ground-truth hotword is independently dropped with probability 0.05 to simulate imperfect retrieval recall. Keeping most real hotwords lets the model use them as contextual bias to correct rare entities and domain-specific terms that an unconditioned decoder would mis-transcribe.
2. **Hard negatives.** An online retriever fetches up to 10 confusable terms per sample using the weighted score $0.6 \cdot \text{char-overlap} + 0.4 \cdot \text{bigram-Jaccard}$ over a per-corpus hotword index, augmented for Mandarin by a syllable-level nearest-neighbor search. These phonetically or orthographically similar distractors are intended to expose the model to distinctions among look-alike hotwords rather than favoring any listed term.
3. **Random distractors.** The remaining slots are filled from the global hotword pool, with the hard-to-random ratio set so that hard negatives saturate first and the list total stays within $[10, 50]$. Random fillers expose the model to irrelevant hotwords and control the prompt list length; this

is intended to discourage reliance on non-matching candidates.

The whole hotword prompt line is then kept with probability 0.8 and dropped otherwise. This exposes the model to inputs both with and without a hotword side channel during training.

4 Training

4.1 Overview

AmphionASR is trained from Qwen3-ASR-1.7B [16] in three successive stages that share a single set of model parameters but differ in which modules are trainable and which data are presented. The main body of training is a three-stage supervised fine-tuning (SFT) recipe (Section 4.2), which first jointly adapts the audio encoder and the LLM to degraded and whispered speech, then introduces hotword- and target-speaker-conditioned transcription with the encoder frozen, and finally replays general-transcription data to mitigate catastrophic forgetting. On top of the resulting checkpoint, two further training steps are applied without revisiting the SFT schedule: Group Relative Policy Optimization (GRPO) refines the LLM with a sequence-level reward that combines transcription accuracy with hotword-agreement (Section 4.3), and the hotword retriever is subsequently trained in isolation by updating only two small MLP adapters while the audio encoder and LLM token embeddings stay frozen (Section 4.4). Throughout, the audio encoder is updated only in the first SFT stage and remains frozen thereafter.

4.2 SFT Recipe Overview

AmphionASR is initialized from Qwen3-ASR-1.7B [16] and produced by a three-stage supervised fine-tuning (SFT) recipe with all-linear LoRA over the eight training sets of Table 3. LoRA uses rank 64 and $\alpha = 128$, applied to every linear projection of each trainable module. All stages use ms-swift [29] with DeepSpeed ZeRO-1; each stage runs for one epoch at learning rate 1×10^{-6} with cosine decay and a 5 % warmup ratio, effective batch size 384 on six GPUs, and 1 % of examples held out for validation.

Stage 1 – degradation and whisper; encoder and LLM jointly tuned. The audio encoder and the LLM are both trainable. Training uses the Voice-in-the-Wild-2M degradation set [2] and the WhispEar whispered-speech set [3]. Both domains contain acoustic conditions that differ from clean speech. We jointly update the audio encoder and the LLM to allow both modules to adapt before prompt-conditioned training is introduced.

Stage 2 – hotword and target-speaker; encoder frozen. The audio encoder is frozen and only the LLM is updated. Training uses the hotword-conditioned mixture from five public corpora (Table 5) together with three synthesized target-speaker partitions (Table 7). This stage trains the LLM on hotword and target-speaker conditions while retaining the Stage 1 encoder.

Stage 3 – replay against catastrophic forgetting; encoder frozen. The audio encoder remains frozen and only the LLM is updated. The training pool combines the Stage 1 and Stage 2 training sets with two general-transcription corpora replayed to mitigate catastrophic forgetting: AISHELL-2 [21] and LibriSpeech [22]. The replay mixture includes both conditioned and unconditioned examples to reduce the risk of forgetting general transcription while continuing training on the Stage 2 tasks.

4.3 Group Relative Policy Optimization

After the three-stage SFT recipe, we apply Group Relative Policy Optimization (GRPO) to the LLM while keeping the audio encoder frozen. The sequence-level reward combines transcription accuracy

with agreement on whether each candidate hotword appears in the output.

Training data. GRPO uses a subset of the hotword-conditioned and general-transcription SFT data, mixed at a 4:1 ratio. Sampling preserves the task and language proportions of the corresponding SFT data.

Data preprocessing and prompt construction otherwise follow the SFT recipe; the GRPO-specific optimization settings are reported below.

Rewards. The accuracy reward is the negative CER clipped to $[0, 1]$,

$$R_{\text{asr}}(\hat{y}, y) = \max(0,\, 1 - \text{CER}(\hat{y}, y)), \tag{2}$$

with empty-reference boundary cases scored as full credit. The hotword match-accuracy reward averages per-candidate binary agreement between predicted and reference hotword presence,

$$R_{\text{hw}}(\hat{y}, y; C) = \frac{1}{|C|} \sum_{c \in C} \mathbf{1}\{\mathbf{1}[c \in \text{Pred}] = \mathbf{1}[c \in \text{Ref}]\}\,. \tag{3}$$

The total reward is the weighted sum

$$R = w_{\text{asr}}\, R_{\text{asr}} + w_{\text{hw}}\, R_{\text{hw}}, \qquad (w_{\text{asr}}, w_{\text{hw}}) = (1.0, 0.3). \tag{4}$$

The GRPO run uses group size $G = 16$, sampling temperature 0.6, PPO clip $\varepsilon = 0.2$, KL coefficient $\beta = 1 \times 10^{-3}$, peak learning rate 1×10^{-6} for one epoch, and a per-device prompt batch of 8 with gradient accumulation of 2.

The evaluation reports the resulting model as a complete training recipe. Without an SFT-only control, it does not isolate the contribution of GRPO from the preceding supervised stages.

4.4 Training of Hotword Retriever

The hotword retriever is trained separately from the three-stage SFT recipe, on top of the merged Stage 2 checkpoint, optimizing only the two small MLP adapters of Section 2.4; the audio encoder and the LLM token-embedding table remain frozen. Training is bilingual with per-language hotword pools: we collect the hotwords annotated in the CommonVoice Mandarin and English hotword splits, deduplicate them, and build a Mandarin pool and an English pool that serve as the candidate vocabulary for each language.

We build the retrieval objective upon GLCLAP [19], adopting its global–local bidirectional contrastive loss: a global term aligning pooled audio with the transcript and a local term aligning each hotword with per-frame audio via a max-over-time similarity, each summed over the speech→text and text→speech directions. Our only deviation is the positive/negative selection of the local branch. The positive for each utterance is its ground-truth hotword; instead of GLCLAP's in-batch negatives, we sample N=4095 negatives per utterance from the same-language hotword pool, excluding the batch's own hotwords and shared across the batch so that the negative embeddings are encoded only once per step. The text→speech direction keeps in-batch audios as candidates. The temperature is learnable. Training runs for 10 epochs at learning rate 3×10^{-4} with cosine decay, a 5 % warmup ratio, a learnable temperature initialized at 0.07, gradient clipping at 1.0, and effective batch size 384 on six GPUs under fp16. The checkpoint with the best validation Recall@1 on the CommonVoice hotword splits is shipped as the retrieval module.

Table 8: Evaluation sets used in this report.

Task	Test sets	Metric
ASR-zh	AIShell-1/2, KeSpeech, CommonVoice ZH, WenetSpeech (net / meeting)	CER
ASR-en	LibriSpeech (test-clean / -other), GigaSpeech, CommonVoice EN	WER
Hotword ASR	CommonVoice ZH / EN hotword splits	CER / WER
TS-ASR	in-house target-speaker test set	WER
Degradation robustness	Voice-in-the-Wild-Bench [2] (16 subsets, 5,000 clips)	WER
Entity ASR	GigaSpeechBench vertical domains (12×ZH + 12×EN) [1]	B-CER / B-WER
Whispered-speech ASR	wEar (zh) / WTIMIT (en) [3]	CER / WER

5 Experiments

5.1 Setups

Normalization. For English benchmarks we apply the Whisper English text normalizer [11] to both reference and hypothesis before computing WER. For Mandarin we apply a stripped-down normalizer that removes punctuation, normalizes full-width and half-width characters and maps Arabic numerals to their Mandarin equivalents before CER.

Baselines. We compare AmphionASR against eight recent open-source speech-LLM and dedicated ASR systems, organized loosely by training objective: audio-understanding and omni-modal LLMs — Whisper-large-v3 [11], Kimi-Audio-7B-Instruct [15], Step-Audio 2 [14], MiniCPM-o [30], MOSS-Audio [31] — and dedicated ASR-oriented LLM-based systems — Qwen3-ASR-1.7B [16], FireRed-ASR2-LLM [32], and GLM-ASR-nano [33]. The baseline columns in Tables 18 and 19 (Appendix D) are produced by us under the same normalizer as the AmphionASR column.

Hotword conditioning setup. We evaluate two conditions on the CommonVoice ZH/EN hotword splits: without retrieval and with retrieval, where the prompt receives the top-K candidates selected from the full test-split pool by the operator of Section 2.4. CER is reported for Mandarin and WER for English; entity B-WER and B-CER are used for GigaSpeechBench.

Target-speaker ASR setup. The test set is an in-house target-speaker test set (synthesized by the same pipeline as training but on held-out speakers). We report positive-slice WER and the false-alarm rate on the silence negative sub-slice of the in-house set. Baselines are Qwen3-ASR-0.6B, Qwen3-ASR-1.7B [16] and Qwen3-Omni [34]. Qwen3-Omni is evaluated zero-shot with the instruction-enhanced dual-audio prompt described in Section 5.4; the other systems retain the project evaluation prompt.

Degradation robustness setup. We evaluate on Voice-in-the-Wild-Bench [2], a 5,000-clip English/-Mandarin benchmark covering seven atomic acoustic degradations (additive noise, far-field propagation, obstruction, echo/reverberation, recording coloration, electronic distortion, transmission dropout) under both real-world recordings (*Real*) and spectrogram-level simulation (*Sim*), plus compound-degradation Mixed subsets — 16 subsets in total. Baseline numbers for eight comparison systems are reproduced from [2].

Table 9: GigaSpeechBench Chinese vertical-domain entity B-CER (%, lower is better). Columns are twelve vertical domains (AGR–MIL); AVG averages over domains with equal weight. Bold marks the lowest value in each column.

System	AGR	AIT	ART	BIO	ECM	ENG	ENT	FIN	HUM	LAW	MED	MIL	AVG
Azure [37]	35.98	32.83	27.17	30.12	37.71	37.76	34.52	9.97	37.97	14.21	26.04	7.44	27.64
Chirp 3 [38]	33.06	32.92	25.96	30.05	42.10	25.39	47.54	11.30	31.00	42.92	24.02	11.50	29.81
ElevenLabs Scribe v2 [39]	29.40	26.86	18.59	27.90	37.66	25.20	32.79	6.90	27.15	12.31	24.93	6.04	22.98
Meta OmniASR 3B [40]	54.40	58.14	35.54	45.07	46.85	48.02	52.78	19.24	41.88	19.11	47.62	21.15	40.82
Qwen3-ASR-Flash [16]	15.55	26.75	17.95	20.02	27.78	15.78	27.89	4.69	14.98	16.06	15.54	7.68	17.56
Qwen3-ASR-1.7B [16]	20.48	23.28	14.57	19.47	30.28	19.72	34.33	5.07	22.84	10.06	18.13	5.29	18.63
NVIDIA NeMo [41]	69.15	78.49	55.91	65.51	68.51	65.91	74.94	40.25	56.88	33.29	65.70	46.80	60.11
GPT-4o Transcribe [42]	38.49	32.46	30.67	33.27	53.67	48.55	46.97	17.90	29.30	18.87	30.93	11.06	32.68
Gemini 3.0 Flash [43]	21.50	18.43	17.44	20.42	38.12	18.25	39.90	7.26	18.40	14.71	15.62	6.79	19.74
Whisper Lv3 [11]	47.07	36.10	33.15	37.45	45.13	39.39	50.03	16.95	39.91	19.67	42.22	13.81	35.07
Dolphin Small [44]	44.02	55.23	32.93	38.09	42.12	41.58	44.83	17.82	39.02	14.05	41.90	20.59	36.02
Dolphin Base [44]	51.83	62.50	38.04	45.86	46.77	46.53	53.57	25.05	43.06	17.48	47.62	26.71	42.09
FunASR-MLT-Nano [45]	27.91	31.69	18.38	26.47	30.86	25.85	31.25	6.81	31.44	10.97	22.92	6.37	22.58
FunASR-Realtime [20]	**6.11**	**15.75**	9.03	14.07	22.54	**9.87**	20.21	**2.45**	10.56	9.20	**8.74**	**2.41**	10.91
Qwen3.5-Omni-Plus [46]	8.70	18.09	9.36	14.10	22.79	12.51	20.52	2.95	10.99	9.85	9.74	3.08	11.89
BigASR [47]	12.88	20.26	12.01	20.50	23.09	17.26	22.46	4.61	15.89	10.72	14.42	5.05	14.93
SeedASR [47]	12.71	20.10	11.94	19.81	23.09	17.33	22.08	4.55	15.89	10.63	14.26	5.10	14.79
AmphionASR (w/o RAG)	23.58	47.49	16.52	26.40	31.71	23.14	33.94	6.70	24.15	10.08	20.91	13.49	23.18
AmphionASR (w/ RAG)	10.79	19.10	**5.81**	**12.12**	**19.85**	10.99	**10.01**	3.05	**7.00**	**8.98**	**8.74**	8.28	**10.39**

5.2 Evaluation Datasets and Metrics

All evaluations use the same metric implementation for offline and batched inference. Table 8 summarizes the test sets reported in this paper.

5.3 Hotword Conditioning

Hotword pools. We evaluate the retriever under two pool regimes. For the GigaSpeechBench vertical domains, we collect every hotword that the benchmark annotates in each domain, deduplicate it, and build a separate retrieval pool per domain and language (Mandarin and English). These per-domain pools range from a few hundred to a few thousand entries. To evaluate larger candidate vocabularies, we additionally build CommonVoice-based pools using Doubao-Seed-2.0-Lite [35] to annotate hotword candidates on CommonVoice Mandarin and English, yielding 10,112 Mandarin and 7,561 English entries, and annotate extra hotwords from GigaSpeech [36] to fill the English pool to 10,000.

GigaSpeechBench vertical-domain entity recognition. GigaSpeechBench [1] measures recognition of domain entities—professional terms and named entities annotated in each reference—rather than full-sentence accuracy. We follow the benchmark protocol: on Mandarin domains the primary metric is **biased CER** (B-CER), which counts substitution, deletion and insertion errors only on entity tokens in the reference; on English domains the counterpart is **biased WER** (B-WER). Each of the twelve vertical domains (agriculture, AI technology, arts, biology, e-commerce, engineering, entertainment, finance, humanities, law, medicine, military) provides an in-the-wild test split, and retrieval uses the corresponding per-domain pool. Every baseline is reported under general ASR (no prompt hotwords). We add two AmphionASR protocols on the same entity metric: without retrieval and with retrieval (K=50, per-domain pool). Baseline numbers are reproduced from the benchmark [1].

Entity-recognition results. Tables 9 and 10 list per-domain entity error for every system plus the domain-average (AVG) column. Without retrieval, AmphionASR reaches 23.18 % AVG B-CER on

Table 10: GigaSpeechBench English vertical-domain entity B-WER (%, lower is better). Columns are twelve vertical domains (AGR–MIL); AVG averages over domains with equal weight. Bold marks the lowest value in each column.

System	AGR	AIT	ART	BIO	ECM	ENG	ENT	FIN	HUM	LAW	MED	MIL	AVG
Azure [37]	9.84	28.56	12.02	18.52	19.45	14.74	21.19	13.17	7.79	17.78	14.88	19.76	16.48
Chirp 3 [38]	7.92	27.63	8.58	15.12	16.13	13.28	21.16	12.05	6.25	15.97	13.10	19.06	14.69
ElevenLabs Scribe v2 [39]	9.54	29.95	9.46	16.46	18.31	13.46	20.32	14.33	8.56	17.18	14.19	22.15	16.16
Meta OmniASR 3B [40]	22.69	42.15	22.60	28.45	27.60	26.53	40.29	22.74	12.22	27.87	24.59	21.03	26.56
Qwen3-ASR-Flash [16]	6.91	26.54	14.61	15.85	14.54	11.32	17.51	10.27	8.48	19.93	14.65	18.15	14.90
Qwen3-ASR-1.7B [16]	7.36	26.79	8.20	15.67	15.71	11.70	19.42	10.35	6.12	15.13	13.52	15.79	13.81
NVIDIA NeMo [41]	14.50	31.73	17.18	24.07	23.98	18.50	36.01	13.34	10.19	20.66	14.36	**11.53**	19.67
GPT-4o Transcribe [42]	15.06	44.01	18.79	17.09	36.64	29.04	29.87	19.75	10.48	20.85	32.10	23.87	24.80
Gemini 3.0 Flash [43]	7.68	27.05	7.04	14.56	16.05	11.56	19.15	11.55	5.66	13.03	12.45	18.24	13.67
Whisper Lv3 [11]	7.60	28.53	9.00	16.34	16.26	12.64	19.47	11.64	6.11	15.64	13.78	18.99	14.67
FunASR-MLT-Nano [45]	10.02	28.24	10.96	19.33	17.45	12.93	25.42	12.78	7.79	18.75	15.30	18.14	16.43
FunASR-Realtime [20]	7.80	22.51	8.92	14.92	13.34	9.21	20.48	10.27	5.05	14.95	12.35	15.47	12.94
Qwen3.5-Omni-Plus [46]	6.82	25.83	6.47	14.33	14.45	10.23	16.93	10.24	5.39	12.77	12.31	17.77	12.79
BigASR [47]	11.05	29.01	10.94	16.74	17.27	11.70	24.58	13.30	7.13	17.45	15.45	19.42	16.17
SeedASR [47]	12.79	29.48	11.13	16.83	17.17	11.62	24.97	13.20	7.19	17.48	15.36	19.46	16.39
AmphionASR (w/o RAG)	8.16	26.60	9.59	15.63	16.65	11.61	22.35	12.72	6.76	16.26	13.66	17.31	14.78
AmphionASR (w/ RAG)	**4.13**	**20.93**	**3.82**	**11.37**	**12.04**	**6.15**	**9.35**	**8.36**	**3.52**	**9.83**	**10.27**	12.39	**9.35**

Table 11: Retrieval-based contextual biasing entity recognition on CommonVoice with K=50 and the 10,112-entry Mandarin and 10,000-entry English pools. Recall@K is retriever-level ("–" without RAG); B-CER and B-WER count errors on annotated entity tokens only.

CommonVoice test set	Condition	Recall@K (%)	B-CER / B-WER (%)
zh hotword test	w/o RAG	–	9.27
zh hotword test	w/ RAG	94.52	2.30
en hotword test	w/o RAG	–	28.93
en hotword test	w/ RAG	95.19	9.81

Mandarin, worse than Qwen3-ASR-1.7B (18.63 %). With the complete retrieval-conditioned pipeline (K=50), AVG B-CER falls to 10.39 % and AVG entity B-WER to 9.35 %, relative reductions of 55 % and 37 %, respectively, relative to the no-hotword-context condition.

The retrieval condition outperforms every listed general-ASR baseline on both language averages. Because those baselines receive no hotword input, this comparison measures the complete retrieval-conditioned pipeline with added context rather than an isolated contribution of retrieval or a ranking under identical inputs.

Per domain, retrieval beats Qwen3-ASR-1.7B on eleven of twelve Chinese domains and all twelve English domains. The largest residual entity errors occur in the English AIT domain and the Mandarin FIN and MIL domains.

CommonVoice large-pool evaluation. Table 11 reports Recall@K and entity B-CER and B-WER on the CommonVoice hotword test splits with and without retrieval. In the retrieval condition, the top-50 candidates come from the 10,112-entry Mandarin and 10,000-entry English pools.

With the complete retrieval-conditioned pipeline, Recall@50 exceeds 94 % in both languages and entity error is lower than in the corresponding no-hotword-context setting. This comparison does not isolate retrieval from the effect of supplying hotword context.

Table 12: Steady-state end-to-end latency with full candidate pools and K=50. Each condition contains 30 utterances per language and runs as single requests on a shared H20. Values are in ms; Precomputed freezes the online Top-50 list outside timing.

Lang.	Condition	Mean	P50	P95	P99
EN	No hotword	67.80	67.63	85.63	88.15
EN	Precomputed	72.67	72.54	87.89	133.36
EN	Online	71.11	72.46	89.24	94.21
ZH	No hotword	66.68	64.40	88.26	100.09
ZH	Precomputed	70.15	66.74	95.04	107.64
ZH	Online	72.07	69.00	96.23	105.52

Latency protocol. We next measure the cost of the frame-level retriever, pairing recognition quality with inference efficiency as in recent ASR and large-pool contextual-biasing evaluations [48, 49]. For each language, we select 30 utterances stratified equally across short, medium, and long duration bands. All requests run serially on one NVIDIA H20 shared with other services, after three warm-up requests. The audio projection path is identical in all conditions, and each decoding request uses a unique identifier to prevent multimodal-cache reuse. We compare (i) no hotword context, (ii) a precomputed Top-50 list frozen outside the timed region, and (iii) online frame-level retrieval followed by the same Top-50 prompt. The latter two conditions therefore differ only in whether candidate scoring is timed. The 10,000-entry English pool contains 7,566 corpus-derived terms and deterministic fillers; the 10,112-entry Mandarin pool contains 10,088 corpus-derived terms and 24 fillers. The fillers do not occur in the references, so this experiment supports within-system latency comparisons rather than cross-system quality comparisons.

Latency results. Relative to precomputed Top-50, online retrieval adds a paired median of 1.37 ms for English and 3.25 ms for Mandarin; the paired P95 increments are 7.92 and 7.17 ms, respectively. Relative to no hotword context, the paired median increments of the complete online path are 2.82 ms and 5.61 ms. The English online mean is lower than the precomputed mean because one precomputed request raises its P99 to 133.36 ms; we therefore base the isolated retrieval comparison on paired percentiles rather than a difference of marginal means. Within this shared-GPU run, the measured online overhead is thus on the order of a few milliseconds, but the observed tail includes shared-resource variation.

Candidate-pool and Top-K scaling. On a fixed 30-utterance subset with K=50, increasing the pool from 100 to approximately 10,000 entries changes the complete audio-projection-and-retrieval segment mean from 19.43 to 20.46 ms in English and from 18.94 to 19.58 ms in Mandarin. Recall@50 rises from 55.56 % to 94.44 % and from 43.66 % to 97.18 %, respectively, largely reflecting the greater target-term coverage of the larger pools. Because this timing segment also includes the audio tower and projector, it is not a measurement of similarity scoring alone. With the full pools, increasing K from 10 to 50 raises retrieval recall from 83.33 % to 93.75 % in English and from 75.93 % to 92.59 % in Mandarin; latency is not monotonic over the three tested values ($K \in \{10, 25, 50\}$). A cold retriever request measures 800.91 ms versus 18.04 ms for the immediately repeated request, so all values in Table 12 exclude cold start.

Taken together, the evaluated pipeline pairs substantial entity-level accuracy gains with a measured warm-state retrieval cost of a few milliseconds. The accuracy claim applies to the complete retrieval-conditioned pipeline, whereas the isolated latency claim compares online retrieval with identical precomputed Top-50 prompts; it does not assign the complete quality gain to the retriever alone.

These measurements are an initial latency characterization rather than a serving-capacity result: each

configuration has only 30 utterances, no concurrent requests are tested, the GPU is shared, and the deployed default retriever remains the utterance-pooled variant. The available summary does not provide aggregate audio duration for an RTF calculation. An exclusive-GPU run with the frame-level retriever and a larger evaluation slice is required before making a production-latency or throughput claim.

5.4 Target-Speaker ASR

Zero-shot Qwen3-Omni protocol. We evaluate Qwen3-Omni-30B-A3B-Instruct without task-specific fine-tuning. Each request provides an enrollment utterance followed by a mixture and uses the instruction in Listing 2. The prompt assigns the two audio roles explicitly and requests `<NO_TARGET_SPEECH>` when the target is absent. This sentinel is deterministically mapped to an empty hypothesis before scoring. Decoding is greedy and capped at 200 output tokens.

Comparison with open baselines. On the positive sub-slice of our in-house target-speaker test set, AmphionASR reaches 13.02 % WER, compared with 31.91 % for zero-shot Qwen3-Omni and 84.97 % for Qwen3-ASR-0.6B.

Target-absent silence behavior. The in-house test set additionally contains $n_{\text{neg}} = 164$ target-absent silence negatives (no speaker active). On these negatives, both Qwen3-ASR baselines yield $\text{FA}_{\text{sil}} = 100.00\,\%$. Zero-shot Qwen3-Omni lowers FA_{sil} to 9.15 %, and AmphionASR reaches 6.10 % (93.90 % correct silence). Its positive-slice miss rate is 4.13 %, so its WER and FA should be read jointly.

These results support speaker-selection capability when the requested speaker is present, and the in-house negatives provide a first target-absent silence measurement. They do not establish competitiveness with dedicated TS-ASR systems under a matched protocol. The available experiments also do not isolate corpus or synthesis effects behind the in-house silence behaviour.

5.5 Degradation Robustness

Table 14 reports the sixteen-subset breakdown. We compare AmphionASR against eight publicly reported systems: Gemini 3-Flash [43], Seed-ASR [47], GPT-4o-transcribe [42], Whisper-large-v3 [11], Qwen2.5-Omni [50], Kimi-Audio [15], Qwen3-ASR-1.7B [16], and Mega-ASR [2] (with its environment-aware routing variant).

Subset-level results. AmphionASR records the lowest WER on four *Real* subsets and four *Sim* subsets. Its leading conditions include obstruction, echo and reverberation, recording coloration, transmission dropout, and simulated noise.

On the Mixed subsets, AmphionASR is second on Real-Mixed and first on Sim-Mixed. Echo and recording coloration remain its highest-error simulated conditions despite leading the corresponding rows.

5.6 Whispered-Speech ASR

We evaluate whispered-speech ASR on the Mandarin wEar and English WTIMIT test sets [3]. We report CER for Mandarin and WER for English. All baseline numbers are produced by us under the same normalizer as Section 5.1; every system is decoded greedily with no whisper-specific prompt or audio preprocessing.

AmphionASR records the lowest error on both whispered-speech test sets (Table 15). On the Mandarin wEar set it reaches 0.58 % CER, ahead of FireRed-ASR2-LLM (1.01 %) and Qwen3-ASR-1.7B (1.22 %);

Table 13: Target-speaker ASR results on the in-house test set. All values are percentages; lower is better. WER is computed on the positive sub-slice. FA_{sil} is the utterance-level false-alarm rate on the target-absent silence condition ($n = 164$). The Qwen3-ASR baselines emit a non-empty transcript on every target-absent sample. †Qwen3-Omni is evaluated zero-shot with the instruction-enhanced prompt in Listing 2. AmphionASR denotes our model. Bold marks the best entry in each column.

	In-house TS-ASR	
System	WER	FA_{sil}
Qwen3-ASR-0.6B [16]	84.97	100.00
Qwen3-ASR-1.7B [16]	78.99	100.00
Qwen3-Omni-30B-A3B† [34]	31.91	9.15
AmphionASR	**13.02**	**6.10**

Table 14: Voice-in-the-Wild-Bench [2] results, WER (%, lower is better). The benchmark contains 5,000 clips covering seven atomic acoustic effects plus a Mixed subset, in two splits: *Real* (public-internet recordings, 150 clips per atomic effect plus a 450-clip Mixed) and *Sim* (spectrogram-level simulation, 350 clips per atomic effect plus a 1,050-clip Mixed). The AmphionASR column uses the same normalizer as Section 5.1. Bold marks the lowest WER in each row.

Subset	Gem3-F	Seed	GPT-4o	Wh-Lv3	Q2.5-O	Kimi-A	Q3-ASR	Mega	**Amph.**
Real (real-world recordings)									
Noise	7.63	8.21	13.19	16.57	11.92	35.10	7.51	**6.12**	6.64
Far-field	5.14	3.06	**1.87**	3.38	2.35	2.71	2.23	2.33	2.27
Obstructed	3.73	3.10	1.57	3.06	2.40	2.49	1.73	1.80	**1.38**
Echo & reverberation	8.75	16.55	15.62	25.34	20.01	24.00	10.40	8.66	**7.91**
Recording coloration	8.38	18.48	13.37	18.33	13.71	8.73	9.57	6.91	**6.32**
Electronic distortion	3.15	3.89	3.70	3.74	2.46	1.83	**1.54**	1.60	1.92
Transmission dropout	5.47	7.97	8.76	7.04	6.34	4.54	4.16	2.72	**2.32**
Mixed	7.99	6.88	5.62	8.91	6.40	4.44	3.30	**2.63**	2.85
Sim (spectrogram-level simulation)									
Noise	10.61	8.11	45.78	18.19	17.88	14.59	9.52	8.09	**7.24**
Far-field	1.90	3.19	2.39	6.85	2.44	1.92	**1.54**	1.69	2.40
Obstructed	2.65	2.76	2.77	6.01	2.08	1.64	**1.27**	1.41	1.97
Echo & reverberation	14.86	18.21	28.76	39.87	32.64	26.58	14.61	12.22	**11.49**
Recording coloration	19.85	23.33	22.60	31.81	30.09	18.09	19.42	13.23	**11.55**
Electronic distortion	7.56	5.71	8.43	8.77	5.96	**2.78**	3.41	3.35	3.29
Transmission dropout	7.65	7.46	7.71	8.05	5.88	6.33	4.19	**2.88**	2.98
Mixed	9.62	9.29	11.00	14.79	10.29	6.19	5.39	4.53	**4.19**

the margin over every baseline indicates that the Stage 1 whisper SFT contributes a genuine gain on whispered phonation rather than reflecting general recognition quality. On the English WTIMIT set AmphionASR reaches 6.11 % WER, a 3.45-point absolute reduction over Qwen3-ASR-1.7B (9.56 %) and 4.84 points below the best non-whisper-trained baseline (Kimi-Audio-7B-Instruct, 10.95 %).

The cross-system spread is markedly larger on Mandarin than on English. On wEar, three systems exceed 15 % CER (Whisper-large-v3 16.33 %, GLM-ASR-nano 15.41 %), indicating that whispered Mandarin is broadly challenging for backbones never exposed to whisper-style acoustic excitation, whereas the whisper-aware systems (AmphionASR, FireRed-ASR2, Qwen3-ASR, Kimi-Audio) cluster below 1.5 %. On WTIMIT the spread is compressed at the top—six of nine baselines fall in 9.56–11.99 %—but two systems degrade sharply: Step-Audio 2 reaches 37.45 % WER and MiniCPM-o / MOSS-Audio exceed 21 %, consistent with insertion-heavy failure modes on atypical excitation. AmphionASR is the only system that is simultaneously best on both languages, which supports the claim that the Stage 1 whisper recipe generalizes across the Mandarin and English whisper sub-corpora without a language-specific trade-off.

Table 15: Whispered-speech ASR on two test sets. WER (%) for English and CER (%) for Mandarin; lower is better. Bold marks the lowest value in each column. "–" means the system has not been evaluated on that test set in this revision.

System	wEar (zh)	WTIMIT (en)
Whisper-large-v3 [11]	16.33	11.99
Kimi-Audio-7B-Instruct [15]	1.48	10.95
Step-Audio 2 [14]	2.40	37.45
MiniCPM-o [30]	4.71	21.77
MOSS-Audio [31]	4.02	21.78
Qwen3-ASR-1.7B [16]	1.22	9.56
FireRed-ASR2-LLM [32]	1.01	15.18
GLM-ASR-nano [33]	15.41	15.50
Fun-ASR-Nano [51]	2.75	11.11
AmphionASR	**0.58**	**6.11**

6 Conclusion

We presented **AmphionASR**, a 1.7 B SpeechLLM that unifies general, hotword-conditioned, target-speaker, degradation-robust, and whispered-speech ASR in a single model. The system combines staged fine-tuning with optional task conditions and retrieval from larger hotword vocabularies. The current evaluation covers Mandarin and English benchmarks.

The complete retrieval-conditioned hotword pipeline provides the clearest observed improvement over the no-hotword-context setting. It attains Recall@50 above 94 % on CommonVoice and reduces average entity error on GigaSpeechBench by 55 % in Mandarin and 37 % in English. Under the measured single-request, steady-state protocol, online retrieval adds a paired median of 1.37 ms in English and 3.25 ms in Mandarin relative to precomputed Top-50 prompts. These results pair substantial contextual benefit with a few milliseconds of measured retrieval cost.

The target-speaker evidence covers positive slices and target-absent silence; distractor-condition false alarms and transfer across public mixture sets remain outside the conclusions supported by the current experiments.

Acknowledgments

Portions of this manuscript were drafted with the assistance of large language models. All technical content, numerical results, and citations were verified by the authors against primary sources before inclusion.

References

[1] Y. Tu, Y. Yang, T. Wang *et al.*, "GigaSpeechBench: A real-world multilingual speech-to-text benchmark," https://github.com/SpeechColab/GigaSpeechBench, 2026.

[2] Y. Xie *et al.*, "Mega-ASR: Towards in-the-wild^2 speech recognition via scaling up real-world acoustic simulation," *arXiv preprint arXiv:2605.19833*, 2026.

[3] Z. Fang, Y. Shen, Z. Guan *et al.*, "WhispEar: A bi-directional framework for scaling whispered speech conversion via pseudo-parallel whisper generation," in *Proc. Interspeech*, 2025.

[4] G. Hinton, L. Deng, D. Yu *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] A. Graves, S. Fernández, F. Gomez *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[6] W. Chan, N. Jaitly, Q. V. Le *et al.*, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[7] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[8] A. Gulati, J. Qin, C.-C. Chiu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[9] A. Baevski, H. Zhou, A. Mohamed *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[10] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[11] A. Radford, J. W. Kim, T. Xu *et al.*, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023.

[12] Y. Chu, J. Xu, X. Zhou *et al.*, "Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[13] C. Tang, W. Yu, G. Sun *et al.*, "SALMONN: Towards generic hearing abilities for large language models," in *ICLR*, 2024.

[14] Step-Audio Team, "Step-Audio 2 technical report," *arXiv preprint arXiv:2507.16632*, 2025.

[15] Kimi Team, "Kimi-Audio technical report," *arXiv preprint arXiv:2504.18425*, 2025.

[16] X. Shi, X. Wang, Z. Guo *et al.*, "Qwen3-ASR technical report," *arXiv preprint arXiv:2601.21337*, 2026. [Online]. Available: https://github.com/QwenLM/Qwen3-ASR

[17] Y. Kong, J. Hou, J. Tang *et al.*, "Contextual biasing for LLM-based ASR with hotword retrieval and RL," *arXiv preprint arXiv:2512.21828*, 2025.

[18] B. Elizalde, S. Deshmukh, M. A. Ismail *et al.*, "CLAP: Learning audio concepts from natural language supervision," in *ICASSP*, 2023.

[19] Y. Kong, F. Cui, L. Guo *et al.*, "GLCLAP: A novel contrastive learning pre-trained model for contextual biasing in ASR," in *Interspeech*, 2025, pp. 5173–5177.

[20] K. An, Y. Chen, Z. Chen *et al.*, "Fun-ASR Technical Report," 2025. [Online]. Available: https://arxiv.org/abs/2509.12508

[21] J. Du, X. Na, X. Liu *et al.*, "AISHELL-2: Transforming Mandarin ASR research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[22] V. Panayotov, G. Chen, D. Povey *et al.*, "LibriSpeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[23] R. Ardila, M. Branson, K. Davis *et al.*, "Common Voice: A massively-multilingual speech corpus," in *LREC*, 2020.

[24] MagicData Tech, "MagicData-RAMC: A rich annotated Mandarin conversational speech dataset," https://www.magicdatatech.com/, 2019.

[25] H. Bu, J. Du, X. Na *et al.*, "AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline," in *O-COCOSDA*, 2017.

[26] Y. Shi, H. Bu, X. Xu *et al.*, "AISHELL-3: A multi-speaker Mandarin TTS corpus," in *Interspeech*, 2021.

[27] V. Pratap, Q. Xu, A. Sriram *et al.*, "MLS: A large-scale multilingual dataset for speech research," in *Interspeech*, 2020.

[28] D. Wang and X. Zhang, "THCHS-30: A free Chinese speech corpus," arXiv preprint arXiv:1512.01882, 2015.

[29] ModelScope Team, "ms-swift: Scalable lightweight infrastructure for fine-tuning," https://github.com/modelscope/ms-swift, 2024.

[30] OpenBMB Team, "MiniCPM-o 4.5 technical report," https://huggingface.co/papers/2604.27393, 2026.

[31] OpenMOSS Team, "MOSS-Audio technical report," https://github.com/OpenMOSS/MOSS-Audio, 2026.

[32] K. Xu, Y. Jia, K. Huang *et al.*, "FireRedASR2S: A state-of-the-art industrial-grade all-in-one automatic speech recognition system," *arXiv preprint arXiv:2603.10420*, 2026.

[33] Z.ai (Zhipu AI) Team, "GLM-ASR-nano," https://huggingface.co/zai-org/GLM-4.6, 2026.

[34] Qwen Team, "Qwen3-Omni technical report," https://github.com/QwenLM/Qwen3-Omni, 2025.

[35] ByteDance Doubao Team, "Doubao-Seed-2.0-Lite large language model," https://www.volcengine.com/product/doubao, 2025.

[36] G. Chen, S. Chai, G. Wang *et al.*, "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," in *Interspeech*, 2021.

[37] Microsoft, "Microsoft Azure Speech to text," https://learn.microsoft.com/azure/ai-services/speech-service/, 2025.

[38] Google Cloud, "Google Chirp 3 speech-to-text," https://docs.cloud.google.com/speech-to-text, 2025.

[39] ElevenLabs, "ElevenLabs Scribe v2 speech-to-text," https://elevenlabs.io/, 2025.

[40] Meta AI, "OmniASR-LLM-3B," https://huggingface.co/facebook/omniASR-LLM-3B, 2025.

[41] NVIDIA, "NVIDIA NeMo Canary (canary-1b)," https://huggingface.co/nvidia/canary-1b, 2025.

[42] OpenAI, "GPT-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[43] G. Comanici, E. Bieber, M. Schaekermann *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[44] DataoceanAI, "Dolphin Mandarin ASR models (base/small)," https://modelscope.cn/models/DataoceanAI/dolphin-base, 2025.

[45] FunAudioLLM Team, "Fun-ASR-MLT-Nano-2512," https://huggingface.co/FunAudioLLM/Fun-ASR-MLT-Nano-2512, 2025.

[46] Qwen Team, "Qwen3.5-Omni technical report," *arXiv preprint arXiv:2604.15804*, 2026.

[47] Y. Bai *et al.*, "Seed-ASR: Understanding diverse speech and contexts with LLM-based ASR," *arXiv preprint arXiv:2407.04675*, 2024.

[48] V. Srivastav, S. Zheng, E. Bezzam *et al.*, "Open ASR leaderboard: Towards reproducible and transparent multilingual and long-form speech recognition evaluation," *arXiv preprint arXiv:2510.06961*, 2025.

[49] X. Gong, A. Lv, Z. Wang *et al.*, "BR-ASR: Efficient and scalable bias retrieval framework for contextual biasing ASR in speech LLM," *arXiv preprint arXiv:2505.19179*, 2025.

[50] J. Xu *et al.*, "Qwen2.5-Omni technical report," *arXiv preprint arXiv:2503.20215*, 2025.

[51] FunAudioLLM Team, "Fun-ASR-Nano-2512: End-to-end real-time ASR model," https://huggingface.co/FunAudioLLM/Fun-ASR-Nano-2512, 2025.

[52] T. Ko, V. Peddinti, D. Povey *et al.*, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017.

[53] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," arXiv preprint arXiv:1510.08484; OpenSLR SLR17, 2015.

Appendix

A Hotword Extraction Prompt

Hotword pools for both training and evaluation (Section 3.3) are mined offline by prompting a large language model with the instruction below. Each utterance transcript is sent as the user message and the model is constrained to return a strict JSON object whose `hotwords` field lists the extracted proper-noun entities. The prompt is reproduced in Listing 1; non-Latin example tokens are romanized (pinyin for Mandarin, RTGS for Thai) for the Latin-only typesetting, while the deployed prompt keeps the original scripts.

```
You are an ASR (Automatic Speech Recognition) data optimization expert. Your
    task is to extract **proper-noun hotwords** from the given text. These
    hotwords help the ASR model correctly recognize words that are rare, easily
    confused, or carry specific reference.

# Core Criterion
Extract words that **could easily be mis-transcribed by an ASR system based on
    pronunciation alone, without context**.
- Extract: personal names, specific place names, organization/brand names,
    domain-specific terms.
- Discard: common nouns, everyday verbs, adjectives, function words.

# Extraction Categories (ONLY the following entity types)

## 1. Personal names / specific titles -- HIGHEST PRIORITY
**Always extract personal names.** This is the most important category. Names
    are the hardest for ASR to transcribe correctly.
   - Extract: full names, nicknames, name+title combinations, transliterated
    foreign names, historical figure names.
   - Examples: "Wang Zhiqiang", "Buster Keaton", "Dr. Schmidt", "Dvorak", "
    Murong Ren", "Fuxi", "Joko Widodo", "Soekarno", "Megawati"
   - Filter out ONLY: generic titles used alone without a name ("teacher", "sir
    ", "laoshi", "xiansheng", "Bapak", "Ibu")

## 2. Specific place names / buildings
   - Extract: named locations that are **uncommon or easily mis-transcribed**
    -- specific streets, parks, districts, landmarks, neighborhoods, provinces/
    states, lesser-known cities, transliterated foreign place names.
   - Examples: "Chaoyang Park", "Arkansas", "Bavaria", "Jaipur", "Nanjing Road
    ", "Yogyakarta", "Surabaya", "Kalimantan", "Bandung", "Borobudur"
   - Filter out: generic location words ("home", "office", "gongsi", "louxia",
    "rumah", "kantor")
   - Filter out: **only the most common country names and world capitals** (e.g
    . "China", "USA", "Japan", "Germany", "France", "UK", "Beijing", "Tokyo", "
    London", "Paris", "Indonesia", "Jakarta"). When in doubt, keep the place
    name.

## 3. Organizations / brands / teams
   - Extract: named companies, schools, institutions, product brands, sports
    teams.
   - Examples: "Tsinghua University", "Starbucks", "NATO", "Lenovo", "Xinjiang
    Tianshan Snow Leopards", "Pertamina", "Garuda Indonesia", "Tokopedia", "
    Universitas Indonesia"
```

```
   - Filter out: generic category words ("bank", "hospital", "xuexiao", "
    chaoshi", "rumah sakit", "sekolah")

## 4. Domain-specific terms / proper nouns
   - Extract: technical terms, legal terms, scientific names (species, taxonomy
    ), historical proper nouns.
   - Examples: "blockchain", "Pythagorean theorem", "Civil Code", "Poaceae"

# Strict Negative Constraints (NEVER extract)
These constraints apply ONLY to common/generic words, NOT to proper nouns
    listed above.
1. **High-frequency common nouns**: "phone", "computer", "clothes", "telepon",
    "komputer", "makanan"
2. **Common verbs / adjectives**: "like", "happy", "expensive", "suka", "senang
    ", "mahal"
3. **Numbers and time expressions**: "two hundred", "3 o'clock", "Monday",
    "2023", "dua ratus", "Senin"
4. **Filler words, pronouns, particles**: "um", "that", "it", "na ge", "en", "
    itu", "ya", "kan", "sih"
5. **Single-character / single-letter words**: isolated single characters --
    too ambiguous as hotwords
6. **Only the most common country/capital names**: "China", "USA", "Japan", "
    Germany", "France", "Beijing", "London", "Indonesia", "Jakarta" -- ASR
    already handles these well. Do NOT over-apply this rule to less common
    places.

# Important: When in Doubt, EXTRACT
If you are unsure whether a word qualifies as a hotword, **extract it**. It is
    better to include a borderline hotword than to miss a genuine one. The only
    exception is the negative constraints above.

# Mixed-Language Handling
When the input contains multiple languages (code-switching), extract hotwords
    in their **original script** as they appear in the text. Do NOT translate
    them.

# Output Format
Return a JSON object with key "hotwords" and value as a list of strings (List[
    str]).
If no hotwords are found, return {"hotwords": []}.
```

Listing 1: Hotword extraction prompt used to annotate training and evaluation hotword pools.

B Zero-Shot Qwen3-Omni TS-ASR Prompt

The zero-shot Qwen3-Omni baseline in Table 13 receives the enrollment and mixture as two ordered audio inputs. The language placeholder is set to zh or en from sample metadata. The exact sentinel is mapped to an empty hypothesis before scoring.

```
Target-speaker transcription task. You will receive exactly two audio clips.

Audio 1 -- ENROLLMENT REFERENCE: Use it only to learn the target speaker's
    voice identity (timbre, pitch, accent, and speaking style). Do not
    transcribe or copy any words from Audio 1.
```

```
Audio 1:
<AUDIO_1_ENROLLMENT>

Audio 2 -- MIXTURE TO TRANSCRIBE: Identify the same speaker by voice
    characteristics, then transcribe only that speaker's intelligible words from
     Audio 2. Ignore every other speaker, background speech, noise, music, and
    overlap.

Output rules:
1. Output only the verbatim transcript; no explanation, label, speaker name,
    quotation marks, or markdown.
2. Never include words spoken only in Audio 1 or by a different speaker in
    Audio 2.
3. If the target speaker is absent or says nothing intelligible in Audio 2,
    output exactly <NO_TARGET_SPEECH>.
Expected transcript language: <LANGUAGE>.
Audio 2:
<AUDIO_2_MIXTURE>
```

Listing 2: Instruction-enhanced prompt used for zero-shot target-speaker ASR with Qwen3-Omni-30B-A3B-Instruct.

C TS-ASR Synthesis Recipe

This appendix details the offline synthesizer used to build the target-speaker training sets of Section 3.5 (Figure 4).

Given single-speaker source manifests, each output sample contains a target waveform mixed with K interferers, an enrollment utterance from the same speaker but *a different recording*, and the target's transcript:

$$\text{mixed}[n] = s_{\text{tgt}}[n] + \sum_{k=1}^{K} \alpha_k \, s_{\text{int},k}[n - o_k], \qquad \alpha_k = \frac{\rho_{\text{tgt}}}{\rho_{\text{int},k}} \cdot 10^{-\text{SNR}_k/20}, \tag{5}$$

where ρ denotes RMS energy, o_k is a uniformly sampled time offset that places interferer k inside the target, and the overlap fraction $|\text{int}_k|/|\text{tgt}| \sim \text{Uniform}[0.1, 1.0]$ controls how much of the target the interferer covers. The default recipe draws $K \sim \text{Cat}(\{1{:}0.8, 2{:}0.2\})$, $\text{SNR} \sim \mathcal{N}(5, 7^2)$ dB clipped to $[-5, 20]$, and 3–5 s enrollments. After mixing we apply an RIR with probability 0.5 (Table 16) and an additive background noise segment with $\text{SNR} \sim \text{Uniform}(0, 25)$ dB. RIR convolutions are truncated to 1.5 s and applied via an FFT-based overlap-add; noise is mixed at the configured SNR and peak-normalized to avoid clipping.

Table 16: Room impulse response and noise pools used by the TS-ASR synthesizer. Within each pool, RIRs are sampled uniformly; noise sources are sampled with the listed weights per training example.

Resource	Source	Weight
SLR26 sim RIRs	OpenSLR-26 [52]	uniform
SLR28 sim RIRs	OpenSLR-28 [52]	uniform
SLR28 real RIRs	OpenSLR-28 (AIR / RWCP / REVERB-2014) [52]	uniform
MUSAN noise	SLR17 [53]	0.30
MUSAN music	SLR17 [53]	0.20
AudioSet road traffic	AudioSet road-traffic subset	0.50

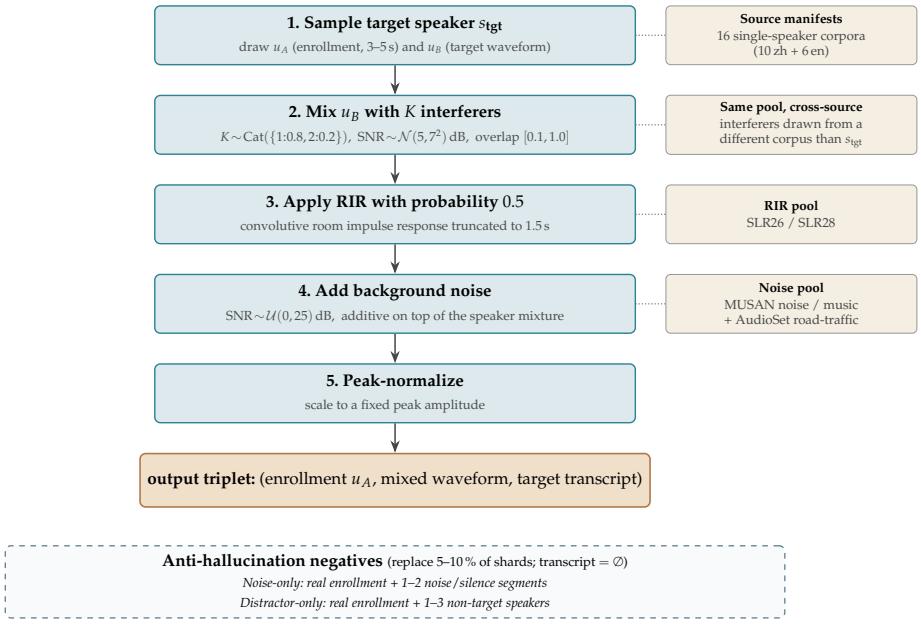

Figure 4: Offline TS-ASR synthesizer. From a target speaker s_{tgt} the pipeline draws two different recordings – one becomes the 3–5 s enrollment, the other becomes the target waveform. Cross-source interferer sampling adds $K \in \{1, 2\}$ competing speakers under Eq. (5), after which an RIR is applied with probability 0.5 (truncated to 1.5 s) and an additive noise segment from the MUSAN / AudioSet pool is mixed at SNR $\sim$ Uniform$(0, 25)$ dB. Five to ten percent of every shard is instead generated by the dashed anti-hallucination recipe shown below the pipeline: a real enrollment is paired with either a noise-only or distractor-only mixture and an empty target transcript, so the decoder learns to remain silent when the requested speaker is absent.

Anti-hallucination negatives. Five to fifteen percent of each target-speaker partition is replaced by *negative* samples whose ground-truth transcript is the empty string. Two sub-types (Table 17) are sampled with equal weight; both preserve a real enrollment but remove the target speaker from the mixed channel. These samples are intended to expose the model to cases where the requested speaker is absent.

Table 17: Anti-hallucination negative sub-types.

Sub-type	Composition	Key parameters
Noise-only	Real enrollment + 1–2 noise / silence segments	duration $[3, 15]$ s; gain $[-30, 0]$ dB; silence inclusion probability 0.1; sources weighted road-traffic 0.6, MUSAN noise 0.3, MUSAN music 0.1
Distractor-only	Real enrollment + 1–3 non-target speakers	SNR $[-5, 20]$ dB; overlap fraction $[0.1, 1.0]$

D General ASR Retention Details

This appendix details the general-ASR retention results motivated by Section 3.4: we examine whether the added personalization capabilities — hotword conditioning, target-speaker selection, degradation

Table 18: English ASR (WER %, lower is better) on four standard English test sets. All numbers are produced by us under the Whisper English normalizer of Section 5.1. Bold marks the lowest WER in each column. "–" means the baseline has not been evaluated on that test set in this revision.

System	LibriSp. clean	LibriSp. other	GigaSpeech	CV en
Whisper-large-v3 [11]	1.88	3.68	10.00	9.77
Kimi-Audio-7B-Instruct [15]	**1.30**	**2.56**	9.59	7.79
Step-Audio 2 [14]	1.41	2.75	9.90	**6.77**
MiniCPM-o [30]	1.46	3.57	9.31	7.96
MOSS-Audio [31]	4.66	7.26	8.96	12.72
Qwen3-ASR-1.7B [16]	1.61	3.39	**8.68**	7.24
FireRed-ASR2-LLM [32]	1.43	2.98	9.19	10.03
GLM-ASR-nano [33]	2.13	4.42	9.43	9.52
AmphionASR	1.59	3.35	9.53	7.87

Table 19: Mandarin ASR (CER %, lower is better) on six standard Mandarin test sets. All numbers are produced by us under the Mandarin normalizer of Section 5.1. Bold marks the lowest CER in each column. "–" means the baseline has not been evaluated on that test set in this revision.

System	AIShell-1	AIShell-2	KeSpeech	CV zh	Wenet. net	Wenet. mtg.
Whisper-large-v3 [11]	5.59	5.23	29.03	12.65	9.98	19.00
Kimi-Audio-7B-Instruct [15]	0.81	2.62	5.36	5.63	5.82	6.03
Step-Audio 2 [14]	0.83	**2.35**	4.16	4.80	5.40	5.51
MiniCPM-o [30]	1.05	2.91	7.65	5.40	6.14	7.10
MOSS-Audio [31]	2.31	3.05	14.97	6.98	5.97	7.61
Qwen3-ASR-1.7B [16]	1.58	2.77	5.21	5.06	5.04	5.85
FireRed-ASR2-LLM [32]	**0.65**	2.37	**3.19**	**4.34**	**4.43**	**4.49**
GLM-ASR-nano [33]	2.54	3.53	9.36	6.84	6.43	8.54
AmphionASR	1.34	2.65	5.71	5.21	5.17	6.30

robustness, and whispered-speech recognition — come at the cost of general transcription quality. Tables 18 and 19 compare AmphionASR with eight open-source systems on four English and six Mandarin test sets under the normalizers of Section 5.1. AmphionASR remains within 0.85 absolute WER/CER points of Qwen3-ASR-1.7B on every test set and leads on four of the ten, indicating that multi-condition fine-tuning preserves general recognition quality rather than materially degrading it; we therefore treat this grid as a retention check rather than a state-of-the-art claim.

Qwen3-ASR-1.7B is the closest comparison because it matches AmphionASR in parameter scale and is likewise an ASR-specialized SpeechLLM: AmphionASR leads on four of the ten test sets, while Qwen3-ASR-1.7B leads on the remaining six by 0.13–0.85 absolute points. Among the other open systems, FireRed-ASR2-LLM [32] leads five Mandarin columns and Step-Audio 2 leads AIShell-2, while the English column leaders vary across Kimi-Audio, Qwen3-ASR-1.7B, and Step-Audio 2; AmphionASR does not lead any column in these two tables, consistent with the retention-check framing above. The highest AmphionASR errors occur on GigaSpeech and the WenetSpeech meeting split, both of which contain more spontaneous speech than the read-speech sets [36].