SOLOIST: A GENERATIVE MODEL FOR BLIND VOICE SEPARATION IN THE WILD

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world voice separation must recover individual voices from overlapping speech, environmental noise, and music, yet isolated source recordings are rarely available for supervision. We introduce Soloist, a generative model for blind voice separation that autoregressively generates individual voice streams as audio-codec sequences conditioned on the mixture, without requiring a target-speaker query or a predefined voice count. We propose a two-stage training paradigm to learn from synthetic mixtures and adapt to real recordings. The first stage combines supervised separation on synthetic mixtures with isolated source targets and an auxiliary task that predicts speaker timelines from real recordings, learning who speaks when from natural acoustics. The second stage applies reinforcement learning to real recordings, with rewards supplied by a model judge that compares candidate separations based on voice purity and content preservation without isolated source targets. Across diverse scenarios, Soloist recovers clean speech from multi-speaker overlap and noisy recordings, and extracts singing vocals from music accompaniment while maintaining relatively high pitch accuracy and intelligibility. On real-world recordings with unknown voice counts, Soloist also separates voices well. Finally, we integrate Soloist into a full-duplex conversation processing pipeline that assembles separated outputs into continuous speaker tracks, demonstrating its practical utility in real conversations.

1 INTRODUCTION

People speak over one another in conversations, films, and online recordings. Recovering their individual voices requires a separator to identify how many voices are present, retain each voice's content and identity, and exclude unrelated sound. The training signal is less complete: a natural recording often comes with a transcript or speaker timeline, but rarely with the isolated waveforms that produced the mixture.

Supervised separators learn this mapping from mixtures constructed with known sources (Luo & Mesgarani, 2019; Subakan et al., 2021). Their targets are precise, while the acoustics and source combinations available for construction constrain the training distribution. Learning from unpaired mixtures has also been studied: mixture invariant training (MixIT) uses mixture-level reconstruction without isolated sources (Wisdom et al., 2020), including adaptation to real meetings (Sivaraman et al., 2022). Our focus is a complementary setting in which a generative separator can use several incomplete references and can be updated on natural audio through model-generated feedback. Appendix F reviews related work in more detail.

We introduce Soloist, a generative model for blind voice separation in real recordings. Given a single-channel mixture, the model generates one audio-codec sequence per voice without requiring a target-speaker query or a supplied voice count. Each sequence is decoded into an individual voice waveform.

Our training recipe follows the references available for each recording (Figure 1). Constructed mixtures provide source targets; a fixed music separator provides pseudo-vocal targets for songs; and speaker timelines provide text targets for natural recordings without isolated waveforms. These examples share one supervised objective. After supervised training, reinforcement learning adapts the model on real recordings by sampling alternative separations of each mixture. A model judge

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

supplies the reward: it describes tracks independently, compares the resulting reports, and checks consistency across candidate orders.

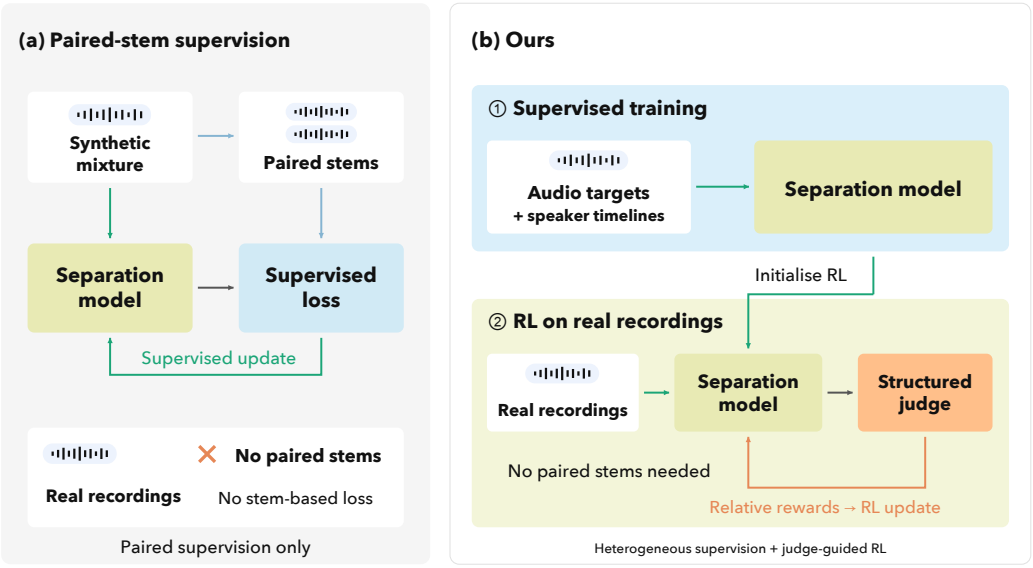

Figure 1: Training with incomplete references. (a) Paired-stem supervision compares predicted audio with known sources; natural recordings usually lack these targets. (b) Our recipe combines audio targets and speaker timelines before adaptation. RL uses model-judge rewards on real recordings. Training details are given in Appendix B.3.

The judge must distinguish useful separation from fluent but incorrect resynthesis. Signal checks detect degenerate outputs and likely copies. The input mixture is included as an anonymous baseline, allowing the judge to compare each separation with returning the input unchanged. To check whether candidate order affects its preferences, the judge ranks the same candidates twice, first in a shuffled order and then in the reverse order. Groups with inconsistent rankings, or reports that cannot distinguish candidate quality, are excluded from RL updates. The remaining groups provide relative rewards that favor purer individual voices and more complete coverage of the audible content.

The paper makes three contributions. First, we formulate blind voice separation as serial codec-sequence generation with an explicit voice count, and show that predicting speaker timelines on real recordings without isolated sources improves count inference. Second, we propose a model-judge reward for RL on real recordings without isolated sources: tracks are described independently, candidates are ranked in text against the unprocessed mixture as an anonymous anchor, and groups whose preferences change substantially when the candidate order is reversed are discarded. Third, we evaluate the model on synthetic, noisy, singing, and real multi-speaker recordings, and show how it is used in a full-duplex conversation processing pipeline.

2 Generative Blind Voice Separation

We formulate blind voice separation as conditional audio generation: given a single-channel recording, the model infers how many voices are present and generates a separate waveform for each. The recording may also contain music or environmental sound, and inference requires neither a supplied voice count nor a target-speaker query. Soloist implements this task by autoregressively generating a variable number of voice streams as sequences of audio-codec frames, which are decoded into waveforms. Its output contains only the separated voices, with no background or residual stream. The formal task definition appears in Appendix B.

Architecture. Soloist uses a Qwen3-4B decoder (Yang et al., 2025) conditioned on both continuous and discrete representations of the mixture (Figure 2). The AuT encoder from Qwen3-Omni (Xu

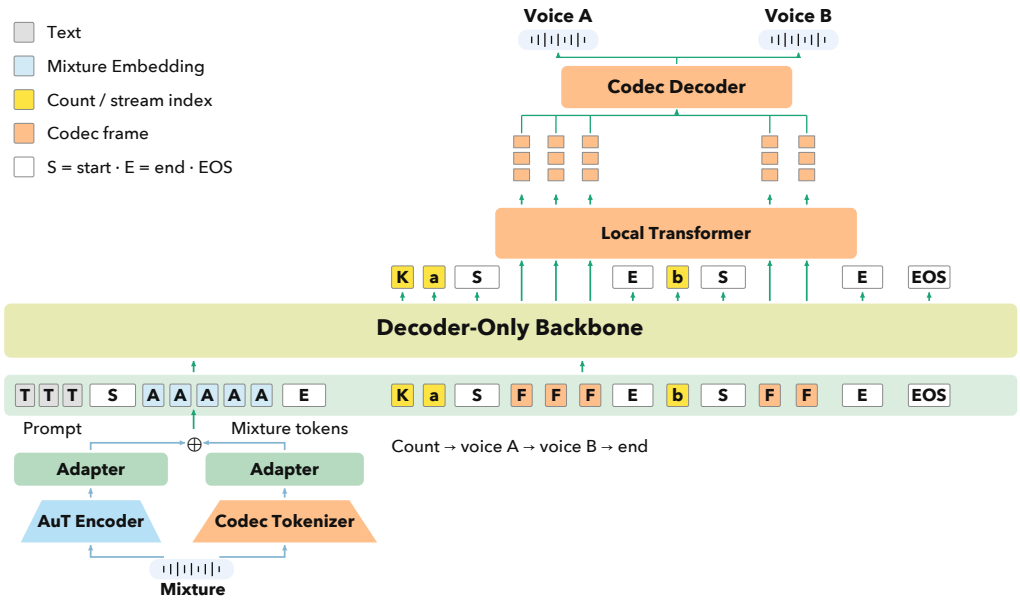

Figure 2: Generative separation architecture of Soloist. Continuous audio features and discrete mixture codes condition the backbone. The text head predicts control tokens, and a local transformer generates the codes within each audio frame.

et al., 2025) supplies continuous features, and MOSS-Audio-Tokenizer-v2 (Gong et al., 2026) supplies twelve residual codec codes per frame at 12.5 frames/s. Gated multilayer perceptrons project the two representations into the backbone space and sum them at input frame positions. At an output frame, a one-layer local transformer predicts the twelve codes sequentially, conditioned on the backbone state. This frame-local decoder contains causal multi-head self-attention and a two-layer feed-forward network, each preceded by layer normalization and wrapped in a residual connection. Codebook-specific linear heads predict each code using the backbone state and the preceding codes within that frame. A separate adaptor feeds the generated codes back into the backbone. The codec is fixed; the audio encoder, backbone, adaptors, and local transformer are trained.

Output format and inference. The backbone's text head predicts tokens that control the output structure, including the voice count, stream indices, and audio boundaries. For separation, it first emits the count and the first stream index, followed by an audio-start token to begin that stream. Whenever the text head emits a frame-request token, the corresponding backbone hidden state conditions the local transformer, which generates the twelve codec codes for that audio frame. The output adaptor maps these codes back into the backbone input space for subsequent autoregressive steps. Frame generation repeats until the text head emits an audio-end token. The model then generates the next stream index and its audio, or emits an end-of-response token to end the response. The codec decoder converts each stream's frame sequence into a waveform.

During supervised training, voice targets are ordered by first onset to define a canonical serial output sequence; evaluation still matches generated streams to references by permutation. For the speaker-timeline task, a separate instruction asks the text head to emit [start][speaker][end] triples sorted by start time, without triggering audio-frame generation. Complete prompts and token roles appear in Appendix B.2.

3 Supervised Learning from Multiple Sources

The first training stage teaches the model to take a mixed recording as input and serially generate the separated voice streams. To establish this ability, we train a single model with three forms of supervision: source waveforms used to construct speech mixtures, teacher-generated vocal targets for songs, and speaker timelines for natural recordings (Figure 3 and Table 1). The first two supervise

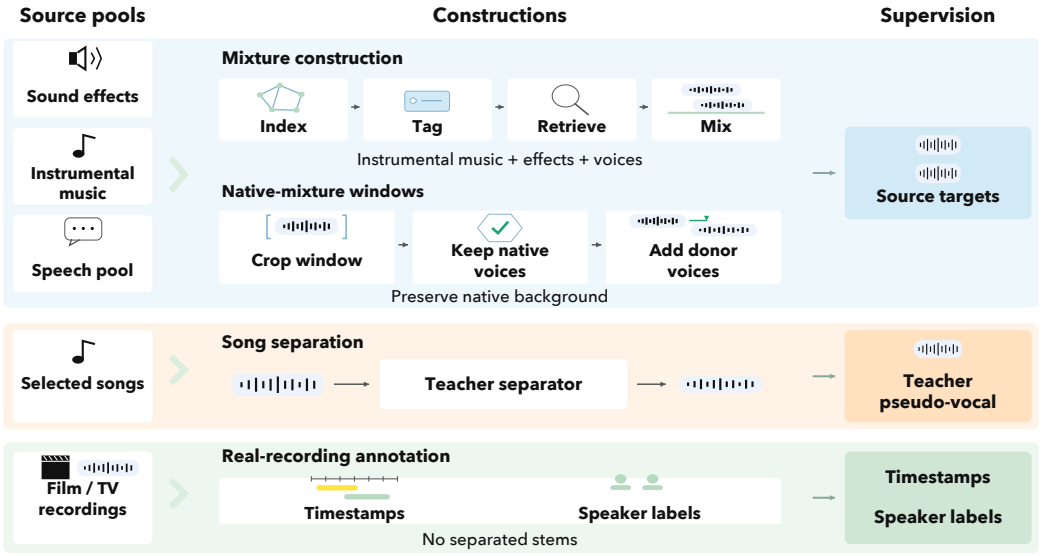

Figure 3: Data construction and the resulting supervision. Known-source mixtures supply audio targets, songs supply teacher pseudo-vocals, and annotated recordings supply speaker timelines. The timeline response contains timestamps and speaker labels.

audio generation, while the third teaches the model who speaks when through text outputs containing timestamps and speaker labels.

Table 1: Training supervision. Timelines supervise text only; pseudo-vocals inherit teacher limitations. Data construction is detailed in Appendix A.

Input family	Available target	Supervised response
Constructed speech mixtures	Source audio from the construction	Count and per-voice codec sequences
Songs	Teacher-generated vocal audio	One vocal codec sequence with count and delimiters
Natural recordings	Speaker-attributed time intervals	Text-only speaker timeline

Constructed mixtures. Scene mixtures combine voice sources with other voices, sound effects, and music beds. Additional constructions overlay donor voices on recording windows that retain background sound (Appendix A).

Songs. A fixed Mel-Band RoFormer separator (Wang et al., 2023b) extracts a vocal pseudo-target from each song. Training uses the song as input and the teacher's vocal output as a single target stream. Accompaniment is not an output target. This is teacher supervision, so agreement with the pseudo-target measures consistency with the teacher rather than recovery of an independently recorded studio vocal.

Natural recordings. Speaker timelines describe who speaks when without providing either words or separated waveforms. These rows train the model to infer speaker activity from natural acoustics through the timeline instruction. They do not provide waveform supervision for separation. We annotate the natural recordings with MOSS-Transcribe-Diarize (MOSI.AI, 2026), retaining speaker labels and start/end times to construct the timeline targets.

Shared objective. Let z contain the assistant text-channel tokens, including frame requests and delimiters, and let a_{tq} denote codebook q of output frame t. The supervised objective combines text

and audio prediction:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{text}} + \lambda \mathcal{L}_{\text{codec}}, \qquad \lambda = 1. \tag{1}$$

Both terms are teacher-forced cross-entropies. The codec term is absent for timeline rows, while the text term supervises the count, stream boundaries, and frame requests for audio-producing rows. No additional count or boundary loss is used. Appendix B.3 gives the training details.

4 Reinforcement Learning on Real Recordings

After supervised training, we perform reinforcement learning on real recordings. Each example contains a real mixed recording and a voice-count annotation from the dataset where available, otherwise from MOSS-Transcribe-Diarize, the model used for speaker-timeline supervision, without isolated source targets. AliMeeting (Yu et al., 2022) counts are obtained from its manually annotated TextGrid files. For each recording, the model samples multiple candidate separations, and a model judge supplies the reward by comparing voice purity, content coverage, and audio quality; Appendix A.3 details the data composition.

4.1 Model-Judge Reward on Real Recordings

In early trials, directly asking Qwen3-Omni to assign rubric scores to separated audio produced large score fluctuations across repeated evaluations of the same clip. We therefore construct relative feedback in two steps (Figure 4): an audio-language model first describes the mixture and each candidate track independently, and a text model then ranks candidate sets from these reports, using an anonymous copy of the input mixture as an anchor. Repeating the ranking under reversed candidate order provides a consistency check for selecting feedback used in policy updates.

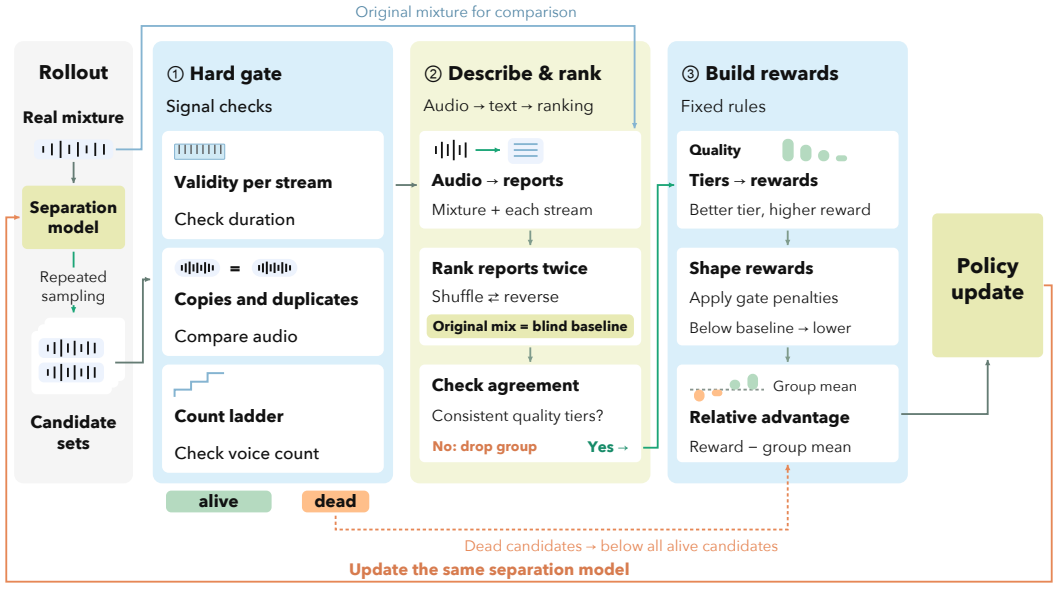

Figure 4: Judge feedback for a group of candidate separations. Signal checks screen stream validity, copies, duplicates, and count errors. An audio-language model describes clips individually, and a text ranker compares their reports with an anonymous mixture anchor in shuffled and reversed orders. Groups that fail the order-consistency check are dropped. Accepted groups receive shaped rewards, with dead candidates below surviving ones; subtracting the group mean gives the policy advantage.

Signal checks. We screen stream duration, signal level, and active fraction, then detect likely input copies and duplicate streams with a log-mel frame-match score. Let $J_{\text{eff},i}$ count the streams in candidate i that pass the duration, level, and activity checks. Given the count annotation K, $J_{\text{eff},i} - K$ determines whether to penalize excess or missing streams. A candidate is marked dead when it emits no usable stream, exceeds the duration bound, receives a zero count penalty, or only passes through

the mixture under the applicable copy rule. These checks are heuristics; perceptually incorrect outputs can still pass them.

Independent perception and an input anchor. Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025) describes the mixture and each surviving track in separate calls. Reports contain audible speaker count, voice traits, content, artifacts, and silence. All clips in a group use the same temporal window. The input mixture is also described with the track prompt and inserted anonymously as a candidate. This anchor lets the ranker compare a separation attempt with returning the input unchanged.

Repeated ranking and consistency. Qwen3.6-27B (Qwen Team, 2026) receives text only: the mixture report, the per-track reports of every candidate, and the count annotation K, without audio input. It assigns candidates to four quality tiers, prioritizing single-voice purity, coverage of the mixture's speakers and content, avoidance of duplicates and invented content, and audio quality. Both ranking calls use the same descriptions and anonymous mixture anchor: candidate labels are shuffled in the first call and presented in reverse order in the second. After mapping results back to the original candidates, let $v_i^{(m)}$ be candidate i's tier score in call m. We compute

$$v_i^{(m)} \in \{q_A, q_B, q_C, q_D\}, \qquad t_i = \frac{1}{M} \sum_{m=1}^{M} v_i^{(m)}, \tag{2}$$

where M is the number of ranking calls and $q_A > q_B > q_C > q_D$ are the tier values. The anonymous anchor is scored in the same way. Their settings and the agreement threshold τ_{min} are given in Table 11.

Since presentation order does not change audio quality, large preference changes after reversal indicate order-dependent or unstable feedback. We measure agreement between the two rankings $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ with Kendall's tie-aware correlation τ_b (Kendall, 1945):

$$\tau_b\left(\mathcal{R}^{(1)}, \mathcal{R}^{(2)}\right) < \tau_{\text{min}} \quad \Longrightarrow \quad r_i = 0 \quad \forall i \text{ in the group.} \tag{3}$$

This rule permits small ranking changes but discards the entire group's feedback when agreement falls below the threshold. We also reject groups when both calls cannot distinguish candidates or disagree on whether they are distinguishable. Rejected groups have zero reward spread and are excluded by the subsequent dynamic filter, so they do not contribute to RL updates. Groups that pass the check use the mean of the two tier scores. The consistency check tests sensitivity to candidate order; the accuracy of content judgments remains dependent on the audio descriptions. Model settings, prompts, complete rejection rules, and a validation on known-source mixtures appear in Appendix C.

Reward shaping. Let t_i be candidate i's mean tier score, t_a the anchor score, and $\gamma_i \in [0, 1]$ its signal-check multiplier. The multiplier combines the mean per-stream copy/duplicate cap with duration, count, and declared-versus-emitted count penalties. We assign

$$R_{\text{judge}}(i) = \begin{cases} \delta\left(\eta + (1-\eta)s_i/s_{\text{max}}\right), & i \text{ is dead,} \\ \max(\epsilon_{\text{alive}}, \min(1, t_i\gamma_i)), & i \text{ survives, } t_i \geq t_a, \\ \max(f_{\text{below}}, \beta_{\text{below}} t_i \gamma_i), & i \text{ survives, } t_i < t_a. \end{cases} \tag{4}$$

Here δ bounds dead-candidate rewards, η sets their minimum fraction, and s_i takes a discrete value determined by the failure category, with maximum s_{max}. The surviving-candidate guard ϵ_{alive} ensures that every surviving reward exceeds the dead-candidate maximum. The below-anchor branch has its own floor f_{below} and scale β_{below}. These floors serve different purposes and need not preserve the ranker's ordering; signal penalties can also change that ordering. Parameter values, failure categories, and multiplier details appear in Appendix C.3.

4.2 Policy Optimization

We sample G candidates per real-recording prompt and center the judge rewards $r_i = R_{\text{judge}}(i)$ within each group:

$$A_i = r_i - \frac{1}{G} \sum_{j=1}^{G} r_j. \tag{5}$$

The update uses a clipped policy-gradient objective (Schulman et al., 2017; Shao et al., 2024), with token averaging, asymmetric clipping, and dynamic group filtering related to recent language-model RL methods (Yu et al., 2025). We do not divide advantages by the group standard deviation. Reward magnitudes therefore affect the update, as well as candidate order. Groups with reward standard deviation at most σ_{min} are filtered to exclude weakly differentiated groups; small nonzero spread would otherwise yield nonzero advantages. A centered per-stream signal, weighted by λ_{stream}, further shapes audio-token advantages. The training loss combines the resulting clipped policy loss with KL regularization and an auxiliary codec cross-entropy term:

$$\mathcal{L} = \mathcal{L}_{\text{PG}} + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{CE}}\mathcal{L}^{+}_{\text{CE}}. \tag{6}$$

RL has two phases: 130 updates at a learning rate of 10^{-6} without the KL term ($\beta_{\text{KL}} = 0$), followed by 90 updates at 2×10^{-6} with it. Training runs in segments, and the KL reference is the policy at the start of the current segment. The codec term applies to positive-advantage rollouts. Appendix B.5 gives G, σ_{min}, the loss weights, and the remaining implementation details.

5 Evaluation

We evaluate separation in three domains: overlapping speech, noisy environments, and singing. Baselines comprise the speech separators SepFormer (Subakan et al., 2021), MossFormer (Zhao & Ma, 2023), ConvTasNet (Luo & Mesgarani, 2019), and SR-CorrNet (Shin & Park, 2026), and the query-based extractors SAM Audio (Shi et al., 2025), AudioSep (Liu et al., 2025), and SoloAudio (Wang et al., 2025), whose target and residual outputs are scored as two streams. With supplied counts, we prefill Soloist's first output token with `2` for LibriMix, or `1` for Voices in the Wild (ViW) and singing vocals, matching baselines with fixed output counts or a single target.

Table 2: LibriMix PI-WER ↓ on overlapping speech with the count supplied. Bold/underline: best/second.

System	Clean	Noised
SepFormer	**0.058**	<u>0.218</u>
MossFormer	0.080	0.289
SAM Audio	0.156	0.351
SR-CorrNet	0.165	0.682
AudioSep	0.435	0.568
ConvTasNet	0.493	0.824
SoloAudio	0.615	0.854
Soloist	<u>0.060</u>	**0.192**

Table 3: ViW WER ↓ and DNSMOS (Reddy et al., 2022) BAK/OVRL ↑ (mean/median). Bold/underline: best/second separator.

System	WER	BAK	OVRL
Unprocessed input	0.261	1.63/1.25	1.43/1.16
SepFormer	0.847	1.87/1.31	1.54/1.19
MossFormer	<u>0.315</u>	<u>2.28</u>/1.71	1.69/1.23
SAM Audio	0.708	2.26/<u>2.06</u>	<u>1.77</u>/<u>1.60</u>
SR-CorrNet	1.074	1.67/1.59	1.56/1.55
AudioSep	0.540	1.94/1.53	1.60/1.35
ConvTasNet	1.082	1.56/1.53	1.53/1.52
SoloAudio	0.882	1.93/1.38	1.52/1.24
Soloist	**0.302**	**3.18/3.45**	**2.45/2.56**

To measure how well spoken content is preserved, we transcribe the separated audio and compute permutation-invariant WER (PI-WER) (Qian et al., 2018; Sklyar et al., 2021). Given K reference transcripts and J output transcripts, we pad the shorter list to $L = \max(K, J)$ with empty transcripts to obtain $\tilde{r}$ and $\tilde{h}$, then compute

$$\text{PI-WER} = \min_{\pi \in \mathcal{P}_L} \frac{\sum_{k=1}^{L} d_{\text{edit}}(\tilde{r}_k, \tilde{h}_{\pi(k)})}{\sum_{k=1}^{K} |r_k|}. \tag{7}$$

Here, $\mathcal{P}_L$ contains all stream permutations, d_{edit} is token edit distance, and the denominator counts all reference tokens. We find the minimum-cost one-to-one assignment with the Hungarian algorithm (von Neumann et al., 2023).

When counts match, PI-WER removes arbitrary output ordering; with one reference and one output, it reduces to ordinary WER. ViW and singing have one reference, so we report WER; for baselines that return two streams, the stream with the lower WER is matched to the reference. When counts differ, unmatched reference tokens count as deletions and unmatched output tokens as insertions. Extra output can therefore push WER above 1, while an extra empty transcript adds no errors. Count

accuracy separately measures whether the number of voiced output streams equals the true count; the model's stated count is recorded separately. Tokenization, corpus aggregation, and the singing metrics are detailed in Appendix D.2.

Overlapping speech. LibriMix (Cosentino et al., 2020) mixes two speakers, with and without added noise (Table 2). WildTV contains 46 natural recordings with one to three speakers (Table 5). On WildTV, SepFormer, MossFormer, and ConvTasNet use a separate checkpoint for each speaker count, selected by the true count; Soloist and SR-CorrNet each infer the count with a single model.

Noisy environments. On ViW (Xie et al., 2026) (Table 3), separation should remove noise and other interference while keeping the voice intact. The unprocessed input yields 0.261 WER; every separator raises it, Soloist the least (0.302), while Soloist obtains the highest DNSMOS background (BAK) and overall (OVRL) quality.

Singing. We evaluate singing vocals on SongYT, MIR-1K (Hsu & Jang, 2010), and MUSDB18-HQ (Rafii et al., 2019), additionally reporting raw pitch accuracy (RPA) and vocal similarity (SingerSim) (Table 4).

Table 4: Singing results: WER ↓, mean RPA ↑, and mean SingerSim ↑. Bold and underline mark the best and second-best comparable results. MUSDB18-HQ has no lyric references. Unprocessed mixtures are omitted: Whisper large-v3 gives very high WER on the original music, so the comparison is not informative.

	SongYT			MIR-1K			MUSDB18-HQ	
System	WER	RPA	SingerSim	WER	RPA	SingerSim	RPA	SingerSim
SepFormer	0.780	0.773	0.955	0.848	0.727	0.890	0.462	0.771
MossFormer	0.389	<u>0.847</u>	**0.982**	0.638	0.783	0.941	<u>0.563</u>	<u>0.845</u>
SAM Audio	0.460	0.846	<u>0.978</u>	0.885	**0.865**	**0.958**	**0.643**	**0.909**
SR-CorrNet	1.187	0.683	0.742	1.082	0.680	0.728	0.431	0.659
AudioSep	<u>0.388</u>	0.457	0.921	<u>0.455</u>	0.779	0.936	0.366	0.788
ConvTasNet	1.195	0.252	0.580	1.239	0.315	0.568	0.093	0.589
SoloAudio	1.006	0.391	0.881	1.058	0.462	0.797	0.251	0.714
Soloist	**0.253**	**0.854**	**0.982**	**0.354**	<u>0.799</u>	<u>0.956</u>	0.438	0.819

Ablation. Speaker-timeline supervision raises SFT count accuracy on every set and lowers PI-WER on noised LibriMix and WildTV, while clean LibriMix PI-WER is nearly unchanged (Table 6).

Table 5: WildTV corpus-level PI-WER ↓ and blind count accuracy (Acc.) ↑. Dashes: inapplicable. <u>Bold/underline</u>: best/second.

	PI-WER				
System	Overall	1 spk	2 spk	3 spk	Acc.
Non-blind					
MossFormer	<u>0.527</u>	<u>0.032</u>	<u>0.371</u>	<u>1.386</u>	—
SepFormer	0.967	0.783	0.801	1.575	—
ConvTasNet	1.541	0.468	0.939	4.113	—
Blind					
SR-CorrNet	1.940	1.572	1.864	2.471	<u>0.217</u>
Soloist	**0.150**	**0.025**	**0.160**	**0.238**	**1.000**

Table 6: Speaker-timeline ablation: PI-WER ↓ and count accuracy (Acc.) ↑. All rows use blind count inference.

		LibriMix		
Recipe	Metric	Clean	Noised	WildTV
SFT w/o timelines	PI-WER	0.139	0.378	0.330
	Acc.	0.890	0.574	0.565
Standard SFT	PI-WER	0.140	0.358	0.287
	Acc.	0.926	0.735	0.761
RL (Soloist)	PI-WER	0.060	0.192	0.150
	Acc.	1.000	1.000	1.000

6 Application: Full-Duplex Conversation Processing

Full-duplex conversations require time-aligned speaker tracks that preserve both overlapping speech and short backchannels. We use Soloist as the separation component of an offline pipeline that

converts diarized recordings into continuous per-speaker audio with aligned transcripts (Figure 5). Speaker timelines determine the active speakers in each inference piece; every piece, including single-speaker regions, passes through the model.

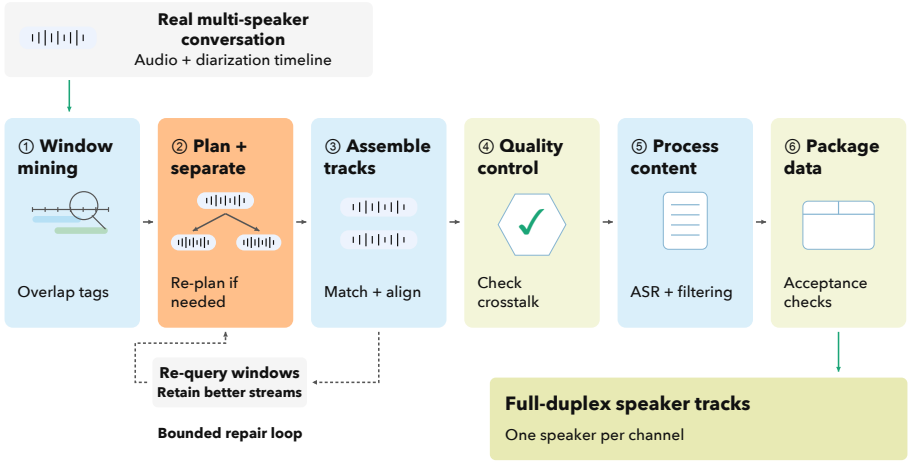

Figure 5: Full-duplex conversation processing. Speaker activity guides window mining and adaptive separation. Voice embeddings link outputs across pieces, and alignment places them on a shared recording timeline. The dashed loop re-queries failed windows and retains improved outputs. Separate identity, missing-speech, and content checks precede transcript-aligned packaging.

The central difficulty is maintaining identity and coverage when independently generated pieces are assembled. We plan short inference pieces around speaker activity, assign their outputs using voice embeddings and a cross-piece consistency term, and place each stream relative to its speaker's timeline onset. Missing or rejected streams trigger bounded repair: a replacement window must contain the original interval, and for a piece that already had enough streams, a re-query replaces the previous output only when separation improves.

Quality control separates speaker purity from content preservation. Diarization and speaker verification identify intrusive speech; source-timeline checks detect missing speech; and energy-envelope agreement in exclusive-speaker regions detects utterances whose voice identity is plausible but whose content is mismatched. Accepted audio, speaker intervals, utterances, and word timestamps share one time axis. Appendix E gives details; the supplementary material has audio examples.

7 Conclusion

We introduced Soloist, a generative model that separates individual voices from a mixture without a target-speaker query or a supplied voice count. Its training combines source-audio and speaker-timeline supervision with reinforcement learning on real recordings, where a model judge compares independently obtained track descriptions under repeated candidate orderings.

With supplied voice counts, Soloist achieves the lowest PI-WER among the compared separators on noisy overlapping speech, speech in noisy environments, and singing vocals. On real-world multi-speaker recordings with unknown voice counts, it also separates voices well, while the singing results show that lyric content, pitch accuracy, and vocal similarity capture different aspects of separation quality. We also integrate the model into an offline pipeline that assembles continuous speaker tracks and aligned transcripts from real conversations. Together, the model and recipe provide a way to learn voice separation from constructed mixtures and natural recordings with incomplete references.

Ethics Statement

Voice separation can make speech more intelligible and accessible, but it can also expose speech that participants did not intend to isolate. Generative separation may introduce words or alter voice characteristics, so its outputs should not be treated as faithful forensic records. Dataset use and any redistribution of audio, annotations, or weights require an assessment of the source licenses and applicable permissions.

Reproducibility Statement

We will release the synthetic-data construction pipeline, the model weights, the inference code, and the part of the training data that can be redistributed. Section 2 and Appendix B describe the model architecture and output protocol; Section 3 and Appendix A describe how each supervision source is built; Section 4 and Appendix C give the RL procedure and the judge reward; and Appendix D specifies the evaluation data and metrics. The supplementary material contains demo audio from the full-duplex pipeline (Appendix E).

Statement of AI Use

We used AI tools to assist with writing and polishing the manuscript, including revising its wording and organization and checking consistency and formatting. We also used AI tools for literature retrieval and discovery and for checking and formatting bibliographic references. The authors reviewed all AI-assisted content and take full responsibility for the paper.

References

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2023. doi: 10.1109/TASLP.2023.3288409.

Chih-Ning Chen, Jen-Cheng Hou, Hsin-Min Wang, Shao-Yi Chien, Yu Tsao, and Fan-Gang Zeng. LLM-guided reinforcement learning for audio-visual speech enhancement. *arXiv preprint arXiv:2603.13952*, 2026.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022. doi: 10.1109/JSTSP.2022.3188113.

Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Qian Chen, and Jiajun Qi. An enhanced Res2Net with local and global feature fusion for speaker verification. In *Proceedings of Interspeech*, pp. 2228–2232, 2023. doi: 10.21437/Interspeech.2023-1294.

Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. LibriMix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ivCd8z8zR2.

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

Hakan Erdogan, Scott Wisdom, Xuankai Chang, Zalán Borsos, Marco Tagliasacchi, Neil Zeghidour, and John R. Hershey. TokenSplit: Using discrete speech representations for direct, refined, and

transcript-conditioned speech separation and recognition. In *Proc. Interspeech 2023*, pp. 3462–3466, 2023. doi: 10.21437/Interspeech.2023-2069.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives. In *Proceedings of Interspeech*, pp. 4300–4304, 2019. doi: 10.21437/Interspeech.2019-2899.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Yitian Gong, Kuangwei Chen, Zhaoye Fei, Xiaogui Yang, Ke Chen, Yang Wang, Kexin Huang, Mingshu Chen, Ruixiao Li, Qingyuan Cheng, Shimin Li, and Xipeng Qiu. MOSS-Audio-Tokenizer: Scaling audio tokenizers for future audio foundation models. *arXiv preprint arXiv:2602.10934*, 2026. v2 weights: https://huggingface.co/OpenMOSS-Team/MOSS-Audio-Tokenizer-v2.

Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2010. doi: 10.1109/TASL.2009.2026503.

M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. doi: 10.1093/biomet/33.3.239.

Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1901–1913, 2017. doi: 10.1109/TASLP.2017.2726762.

Anurag Kumar, Andrew Perrault, and Donald S. Williamson. Using RLHF to align speech enhancement approaches to mean-opinion quality scores. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10888446.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR – half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630, 2019. doi: 10.1109/ICASSP.2019.8683855.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26874–26901, 2024.

Xubo Liu, Qiuqiang Kong, Yan Zhao, Haohe Liu, Yi Yuan, Yuzhuo Liu, Rui Xia, Yuxuan Wang, Mark D. Plumbley, and Wenwu Wang. Separate anything you describe. *IEEE Transactions on Audio, Speech and Language Processing*, 33:458–471, 2025. doi: 10.1109/TASLP.2024.3520017.

Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27 (8):1256–1266, 2019. doi: 10.1109/TASLP.2019.2915167.

MOSI.AI. MOSS Transcribe Diarize Technical Report, 2026. URL https://arxiv.org/abs/2601.01554.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Yanmin Qian, Xuankai Chang, and Dong Yu. Single-channel multi-talker speech recognition with permutation invariant training. *Speech Communication*, 104:1–11, 2018. doi: 10.1016/j.specom.2018.09.003.

Qwen Team. Qwen3.6-27B. https://huggingface.co/Qwen/Qwen3.6-27B, 2026.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ—an uncompressed version of MUSDB18, 2019. URL https://doi.org/10.5281/zenodo.3338373.

Chandan K. A. Reddy, Vishak Gopal, and Ross Cutler. DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 886–890, 2022. doi: 10.1109/ICASSP43922.2022.9746108.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, Christoph Feichtenhofer, Piotr Dollár, Wei-Ning Hsu, and Ann Lee. SAM Audio: Segment anything in audio. *arXiv preprint arXiv:2512.18099*, 2025.

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. *arXiv preprint arXiv:2601.21337*, 2026. URL https://arxiv.org/abs/2601.21337.

Ui-Hyeop Shin and Hyung-Min Park. Asymmetric encoder-decoder based on time-frequency correlation for speech separation. *arXiv preprint arXiv:2603.29097*, 2026.

Aswin Sivaraman, Scott Wisdom, Hakan Erdogan, and John R. Hershey. Adapting speech separation to real-world meetings using mixture invariant training. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 686–690, 2022. doi: 10.1109/ICASSP43922.2022.9747855.

Ilya Sklyar, Anna Piunova, and Yulan Liu. Streaming multi-speaker ASR with RNN-T. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6903–6907, 2021. doi: 10.1109/ICASSP39728.2021.9413471.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25. IEEE, 2021. doi: 10.1109/ICASSP39728.2021.9413901.

Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji. Recursive speech separation for unknown number of speakers. In *Proceedings of Interspeech*, pp. 1348–1352, 2019. doi: 10.21437/Interspeech.2019-1550.

Efthymios Tzinis, Yossi Adi, Vamsi Krishna Ithapu, Buye Xu, Paris Smaragdis, and Anurag Kumar. RemixIT: Continual self-training of speech enhancement models via bootstrapped remixing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1329–1341, 2022. doi: 10.1109/JSTSP.2022.3200911.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006. doi: 10.1109/TSA.2005.858005.

Thilo von Neumann, Christoph Boeddeker, Keisuke Kinoshita, Marc Delcroix, and Reinhold Haeb-Umbach. On word error rate definitions and their efficient computation for multi-speaker speech recognition systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094784.

Helin Wang, Jiarui Hai, Yen-Ju Lu, Karan Thakkar, Mounya Elhilali, and Najim Dehak. SoloAudio: Target sound extraction with language-oriented audio diffusion transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10890066.

Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, and Qian Chen. CAM++: A fast and efficient network for speaker verification using context-aware masking. In *Proceedings of Interspeech*, pp. 5301–5305, 2023a. doi: 10.21437/Interspeech.2023-1513.

Ju-Chiang Wang, Wei-Tsung Lu, and Minz Won. Mel-Band RoFormer for music source separation. *arXiv preprint arXiv:2310.01809*, 2023b. ISMIR 2023 Late-Breaking/Demo.

Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe. TF-GridNet: Integrating full- and sub-band modeling for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3221–3236, 2023c. doi: 10.1109/TASLP.2023.3304482.

Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J. Weiss, Kevin Wilson, and John R. Hershey. Unsupervised sound separation using mixture invariant training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3846–3857, 2020.

Zhifei Xie, Kaiyu Pang, Haobin Zhang, Deheng Ye, Xiaobin Hu, Shuicheng Yan, and Chunyan Miao. Mega-ASR: Towards in-the-wild^2 speech recognition via scaling up real-world acoustic simulation. *arXiv preprint arXiv:2605.19833*, 2026. URL https://arxiv.org/abs/2605.19833.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

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, Jianxin Yang, Jiaxi Yang, Jing Zhou, 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 arXiv:2505.09388*, 2025.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering complex control in MOBA games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6672–6679, 2020. doi: 10.1609/aaai.v34i04.6144.

Takuya Yoshioka, Zhuo Chen, Changliang Liu, Xiong Xiao, Hakan Erdogan, and Dimitrios Dimitriadis. Low-latency speaker-independent continuous speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6980–6984, 2019. doi: 10.1109/ICASSP.2019.8682274.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245. IEEE, 2017. doi: 10.1109/ICASSP.2017.7952154.

Fan Yu, Shiliang Zhang, Yihui Fu, Lei Xie, Siqi Zheng, Zhihao Du, Weilong Huang, Pengcheng Guo, Zhijie Yan, Bin Ma, Xin Xu, and Hui Bu. M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6167–6171, 2022. doi: 10.1109/ICASSP43922.2022.9746465.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. doi: 10.1109/TASLP.2021.3129994.

Zihan Zhang, Xize Cheng, Zhennan Jiang, Dongjie Fu, Jingyuan Chen, Zhou Zhao, and Tao Jin. MARS-Sep: Multimodal-aligned reinforced sound separation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AhvApZghHf.

Shengkui Zhao and Bin Ma. MossFormer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096646.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. Datasets and Benchmarks Track.

A Training Data and Construction

A.1 Supervised Data

Constructed mixtures. The constructed corpus comprises roughly 200k hours of synthesized audio built on a cleaned Pixabay sound-effect library, together with about 50k hours of music. Mixtures are built in three steps. *Semantic tagging* labels each background recording with a scene category derived from pre-computed scene topics. *Candidate retrieval* queries each scene label for semantically similar audio and keeps a pool of 20 candidate clips that mixes long and short recordings at a fixed ratio. *Mixture synthesis* lays a long recording as the acoustic bed when a bed is used, overlays short sound effects at random positions, and adds the voice sources as independent streams. Because every source is known by construction, the isolated voice waveforms serve as separation targets without manual annotation. Scene mixtures place one to four Chinese, English, or Japanese voices on a retrieved background: a music or ambient bed, a bed with transient effects, transient effects alone, or silence. Overlay and background-preserving windows instead add donor voices to 5–30 s windows that already carry an acoustic background. Table 7 lists these constructions together with the timeline and song data described below. Indexed rows can reuse recordings, so counts describe training examples rather than unique audio sources.

Table 7: Supervised data composition.

Construction	Rows	Voices	Target type
Scene mixtures	12,786,360	1–4	Constructed voice sources
Overlay windows	5,000,000	3–5	Voice sources
Background-preserving windows	5,000,000	2–6	Voice sources
Timeline windows	299,642	2–6	Speaker time intervals
Song segments	5,368,771	1	Teacher pseudo-vocal
Total	28,454,773		

Speaker timelines. We select multi-speaker recordings from in-domain data spanning film, television, animation, short video, podcasts, radio, audiobooks, and sports commentary to provide speaker-activity supervision. The nominal windows span 10–120 s and contain two to six speakers. Targets are time-interval triples ordered by start time, with speaker IDs assigned by first appearance. Upstream transcripts are excluded from the prediction target; the output format is specified in Appendix B.2.

Singing vocals. Song segments are cut from in-house song recordings into clips of at most two minutes before teacher extraction. The teacher's vocal stream can contain backing vocals or multiple singers, so it forms a combined vocal target rather than individual singer streams.

A.2 Supervised Sampling

The sampler groups examples by source count and duration, with duration boundaries at 10 and 20 s. Timelines form a separate administrative class indexed by zero; their shortest-duration cell has zero weight, leaving twenty active classes. The sampling-budget shares are 4.81% for timelines and 4.0/26.6/39.5/19.8/4.5/0.8% for one through six source streams.

A.3 RL Data

The RL prompt set contains 7,563 real recordings without isolated source targets. Every recording has a voice-count annotation. AliMeeting uses its manual TextGrid annotations; other recordings use dataset annotations where available and MOSS-Transcribe-Diarize otherwise. All prompts use the model-judge reward described in Section 4.1, with the annotated count used in the signal checks and ranking request.

The recordings include far-field meetings, film, television, animation, podcasts, short video, and audiobooks.

B Model and Training Details

B.1 Architecture and Sequence Representation

Table 8 lists the model components. For a single-channel recording x containing $K \geq 1$ voices, a response contains a declared count $\widehat{K}$ and J generated streams:

$$(\widehat{K}, \hat{y}_{1:J}) \sim \pi_\theta(\cdot \mid x), \qquad \widehat{K} \in \{1, \ldots, K_{\text{max}}\}, \tag{8}$$

Here K_{max} is the decoding cap. The declared count and emitted stream count are distinct so that malformed responses can be measured. The vocabulary covers voice-containing recordings and has no zero-voice response. With T input frames and T_k frames in output stream k, each codec frame contains twelve codes $a_{tq} \in \{0, \ldots, 1023\}$; code 1024 denotes padding. The signal gate additionally computes a valid-stream count J_{eff} (Appendix C.3). Signal validity is independent of whether a stream isolates the correct speaker.

Table 8: Model configuration.

Component	Configuration
Backbone	Qwen3-4B (Yang et al., 2025)
Continuous encoder	Qwen3-Omni AuT; trainable
Adaptors	Gated MLPs: $2560 \to 8192 \to 2560$ for codec embeddings and $2048 \to 8192 \to 2560$ for AuT features; separate output-code adaptor
Local transformer	One layer; hidden size 2560; 32 heads of dimension 80; twelve codebooks with 1024 entries plus padding
Codec	MOSS-Audio-Tokenizer-v2; first 12 of 32 residual quantizers; frozen
Parameter count	5,385,239,424

B.2 Output Interface and Supervision

The model retains the Qwen3 vocabulary and chat template, assigning audio roles to existing tokens (Table 9). At each input-frame position, it sums the projected continuous feature and the embedded mixture codes. At an output-frame request, the local transformer generates twelve codes and the output adaptor embeds them for subsequent backbone steps.

Table 9: Vocabulary tokens reused for the audio interface.

Token	Role
`<|im_start|>`, `<|im_end|>`	Turn start and turn end / stop
`<|vision_start|>`, `<|vision_end|>`	Audio boundaries for input and output
`<|vision_pad|>`	Input frame, replaced by the mixture embedding
`<|video_pad|>`	Output frame request, replaced by the generated-code embedding
`<|endoftext|>`	Sequence padding

Teacher forcing supervises all assistant text-channel positions, including indices, counts, delimiters, frame requests, newlines, and the stop token. Audio-code cross-entropy acts only at output-frame positions. Separation targets follow first onset; the timeline task uses the text channel alone.

The two requests below specify the separation and timeline tasks. The fixed Chinese strings are the actual model inputs. $\times T$ denotes repeated frame placeholders, and the response values illustrate the format.

Separation.

```
<|im_start|>system
你是一个人声分离助手。<|im_end|>
<|im_start|>user
```

请先输出这段混合音频中的说话人个数，再分别分离并输出每一段人声，只输出编号和对应分离音频：<|vision_start|><|vision_pad|>×T<|vision_end|><|im_end|>
<|im_start|>assistant
2
1<|vision_start|><|video_pad|>×T_1<|vision_end|>
2<|vision_start|><|video_pad|>×T_2<|vision_end|><|im_end|>

The system instruction asks the model to act as a voice separation assistant. The user instruction requests the speaker count followed by indexed separated audio. The mixture immediately follows the colon. Each audio placeholder denotes one frame of twelve codes.

Timeline.

<|im_start|>system
你是一个人声分离助手。<|im_end|>
<|im_start|>user
请输出这段混合音频中每个说话人的发声时间段，格式为 [开始时间][说话人编号][结束时间]，按开始时间排序，不要输出语音内容：
<|vision_start|><|vision_pad|>×T<|vision_end|><|im_end|>
<|im_start|>assistant
[0.06][S01][1.22][1.44][S02][3.80]⋯ <|im_end|>

The user instruction requests speaking intervals in [start][speaker][end] form, sorted by start time, without speech content. Times are seconds to two decimal places. Speaker IDs follow first appearance and can recur across intervals. This response has no count token or audio-frame placeholders.

B.3 Training Stages

Training has two stages. Supervised training on the corpus in Table 7 establishes separation and speaker-activity prediction. RL then adapts the model on real recordings through model-judge rewards.

Supervised training uses a peak learning rate of 10^{-4}. RL has two phases. Phase 1 runs for 130 updates with a learning rate of 10^{-6} and no KL regularization. Phase 2 runs for 90 updates with a learning rate of 2×10^{-6} and KL regularization. Training runs in segments, and the KL reference is the policy at the start of the current segment. Table 10 lists the RL configuration; the two phases differ only in learning rate and KL regularization.

B.4 Supervised Objective

The source and timeline tasks share the objective in Equation (1). Text positions u and output codebook positions (t, q) contribute the negative log probabilities

$$-\log p_\theta(z_u \mid z_{<u}, x), \qquad -\log p_\theta(a_{tq} \mid a_{t,<q}, h_t), \tag{9}$$

where h_t conditions output frame t. Text and codec losses have equal configured weights; additional count and boundary losses have weight zero.

B.5 Policy Objective and Optimization

For the token likelihood ratio $\rho_{it} = \pi_\theta(z_{it} \mid z_{i,<t}, x)/\pi_{\text{old}}(z_{it} \mid z_{i,<t}, x)$ and advantage A, the clipped surrogate is

$$b(\rho, A) = \min(\rho A, \text{clip}(\rho, 0.8, 1.28)A)\,, \tag{10}$$

$$u(\rho, A) = \begin{cases} \max(b(\rho, A), 3A), & A < 0, \\ b(\rho, A), & A \geq 0. \end{cases} \tag{11}$$

The policy loss averages the negative surrogate over eligible tokens, with a KL penalty and a codec cross-entropy auxiliary term. Audio likelihood ratios use truncated importance sampling. Centered per-stream feedback augments audio-token advantages as specified in Appendix C.4.

Table 10: RL configuration for both phases. Only the learning rate and KL coefficient change between phase 1 and phase 2.

Setting	Value
Group and batch	$G = 8$ candidates/prompt; 48 retained groups/update; 384 sequences
Dynamic sampling	Initial 72 attempted groups; cap 400; filter reward std ≤ 0.005
Advantage	Group-centered reward; no standard-deviation normalization
Policy clipping	Lower/upper ranges 0.2/0.28; negative-advantage dual clip 3.0 (Ye et al., 2020)
KL and entropy	k_1 KL estimator; coefficient 0 in the first phase and 0.001 in the second; no entropy bonus
Optimizer	Adam; learning rate 10^{-6} in the first phase and 2×10^{-6} in the second; 10 warm-up updates then constant; $(\beta_1, \beta_2) = (0.9, 0.98)$; $\epsilon = 10^{-8}$; no weight decay
Rollout sampling	Temperature 1.0; top-$p = 1.0$; maximum response 2048 tokens
Audio policy term	Coefficient 1.0; truncated importance sampling enabled
Stream feedback	Coefficient 0.5 on centered stream scores
Codec auxiliary	Weight 0.05; positive-advantage rollouts only; codebook decay 0.8

In Equation (6), the enabled KL coefficient is $\beta_{\text{KL}} = 0.001$ with the k_1 estimator, the auxiliary coefficient is $\lambda_{\text{CE}} = 0.05$, and centered stream feedback has weight $\lambda_{\text{stream}} = 0.5$. The group size is $G = 8$ and the spread threshold is $\sigma_{\text{min}} = 0.005$. Phase 1 uses $\beta_{\text{KL}} = 0$; phase 2 uses the stated coefficient.

C Judge and Reward Details

C.1 Audio Perception and Candidate Ranking

Each perception call processes one clip. For long recordings, the mixture, anchor, and candidate streams share a uniformly positioned 30 s window. Mixture descriptions provide global context; independently obtained track descriptions provide the evidence for each candidate. The mixture anchor uses the same track prompt as generated streams. The text ranker receives the mixture report, the candidate track reports, and the count annotation K; it receives no audio. The perception requests do not receive the count annotation.

Table 11: Judge inference settings; tier scores are averaged over two rankings.

Component	Setting
Perception	Qwen3-Omni-30B-A3B-Instruct; greedy; maximum 400 tokens; repetition penalty 1.05
Text ranker	Qwen3.6-27B; thinking disabled; temperature 0.3; top-$p = 0.9$; maximum 1024 tokens; repetition penalty 1.05
Candidate presentation	$M = 2$ calls; randomly shuffled labels, then reversed order; anonymous mixture anchor included
Tier values (q_A, q_B, q_C, q_D)	A/B/C/D $\mapsto$ 1.0/0.7/0.4/0.1
Consistency threshold	$\tau_{\text{min}} = 0.3$; accept Kendall's $\tau_b \geq \tau_{\text{min}}$
Group rejection	At most one candidate surviving the signal checks; failed mixture or anchor description; fewer than two fully described surviving candidates; failed ranking call; low rank agreement; both calls indistinguishable; conflicting indistinguishable flags

The serving backend disables thinking through the chat-template option. If the primary text service fails, the implementation falls back to the perception model for text ranking. This branch uses a different ranker. A description fails when the perception model returns no valid JSON in three attempts. A surviving candidate with a failed track description is left out of the ranking and receives a floor reward of 0.05; if fewer than two fully described candidates remain, the whole group is rejected.

C.2 Judge Validation on Known-Source Mixtures

The judge never observes isolated sources, reference transcripts, or evaluation metrics; it only orders candidates of the same recording. We therefore test whether this ordering agrees with a reference ordering on mixtures whose sources are known. We take 40 LibriMix mixtures (20 clean and 20 noised) and, for each, eight separation outputs randomly drawn from the inference results of intermediate supervised checkpoints. LibriMix sources are clean read speech, so PI-WER reflects most quality differences between separation outputs and serves as the reference ordering. The judge describes each track with Qwen3-Omni, ranks each group twice in text with the candidate order reversed, and applies the order-consistency check. In this test the text ranker runs at temperature 0, and dead and below-anchor candidates receive a common floor score; pairs are compared by the final judge scores.

With Qwen3.6-27B as the text ranker, 36 of the 40 groups pass the order-consistency check. These groups contain 307 output pairs whose PI-WER differs by more than 0.2. The judge prefers one output in 207 of them and agrees with the PI-WER order in 184, or 88.9% (95% confidence interval 80.8–95.6% from 10,000 bootstrap resamples of groups). Per group, Kendall's τ_b between the judge scores and the PI-WER order has a median of 0.40 (interquartile range 0.16–0.57) over the 31 passing groups whose judge scores are not all tied, and it is positive in 29 of them.

Table 12: Judge validation on 40 LibriMix groups of eight outputs. Agreement is measured on pairs whose PI-WER differs by more than 0.2, excluding pairs the judge ties.

Ranking protocol	Groups passing check	Agreement with PI-WER order
Direct ranking from audio (Qwen3-Omni)	0/40	—
Describe, then rank in text (Qwen3.6-27B)	36/40	88.9%

The same test compared ranking protocols. Ranking all outputs of a group directly from audio with Qwen3-Omni passed the order-consistency check in none of the 40 groups, whereas describing each clip in isolation and ranking the descriptions in text passed in 36 (Table 12). Together with the rubric-score fluctuations noted in Section 4.1, this result motivated the describe-then-rank design. The reference ordering reflects transcript accuracy only, so the test does not measure voice purity or audio quality beyond their effect on transcription. Separately, the RL model has lower PI-WER than standard SFT on both LibriMix conditions and on WildTV (Table 6), although the reward uses no reference transcripts.

C.3 Signal Gates and Reward Shaping

Signal checks operate at 16 kHz with the parameters in Table 13. The number of streams passing the duration, level, and activity tests is J_{eff}. Count penalties relative to the annotation K use this quantity; protocol consistency compares the declared count $\widehat{K}$ with the emitted count J.

Table 13: Signal-gate parameters for reward shaping.

Check	Setting
Stream duration	$d_k > 0.2$ s; duration ratio lower bound 0.35, upper bound 1.05
Signal level and activity	RMS > -45 dBFS; active fraction ≥ 0.05
Activity frames	50 ms; active above -40 dBFS
Log-mel representation	FFT 512; hop 256; 40 mel bands; frame-match tolerance 1.5
Input copy	Frame-match threshold 0.8; per-stream cap 0.1
Duplicate stream	Frame-match threshold 0.8; per-stream cap 0.3
Excess valid streams	Multipliers 1.0/0.7/0.4 for excess 0/1/2; 0 beyond
Missing valid streams	Multipliers 1.0/0.5/0.2 for deficit 0/1/2; 0 beyond
Declared/emitted count mismatch	Multiplier 0.6
Dead categories	No streams; no valid streams; overlong; zero count penalty; mixture passthrough

Copy and duplicate checks compare log-mel frames, allowing tolerance for codec reconstruction. The configured copy exemption permits a mixture-like output when the declared count is one and the count annotation is at most one. This accommodates single-source reconstruction, while leaving background retention and missed speakers for the subsequent quality assessment.

For the dead-candidate branch of Equation (4), severity is $s_i = 0$ for passthrough, no streams, or no valid streams; $s_i = 0.4$ for overlong output; and $s_i = 0.6$ for a zero count penalty. These values put dead candidates below surviving candidates within an accepted group. Rejected groups receive zero reward throughout.

Table 14: Reward-mapping constants (Equation (4)).

Parameter	Value	Role
δ	0.005	Dead-candidate maximum
η	0.25	Minimum fraction for dead rewards
s_{max}	0.6	Maximum category severity
ϵ_{alive}	0.006	Surviving-candidate guard
f_{below}	0.05	Below-anchor floor
β_{below}	0.3	Below-anchor scale

Thus $s_i \in \{0, 0.4, 0.6\}$. The multiplier γ_i is the mean per-stream copy/duplicate cap multiplied by the duration and count step penalties, and by the declared/emitted-count factor (0.6 on mismatch). The guard ϵ_{alive} is a constructive safeguard: it exceeds δ, while the unguarded product can fall below δ; for example, $0.1 \times 0.1 \times 0.4 \times 0.6 = 0.0024$. It was inactive for surviving candidates in the recorded runs. The separate floor f_{below} is larger, so this mapping is not globally monotone in tier score.

C.4 Per-Stream Feedback

The track report also supplies local feedback for audio-token updates. For stream k in candidate i, let n_{ik} denote the audible-speaker count, $\alpha_{ik} \in \{0, 0.05, 0.2\}$ the penalty for none, mild, or severe artifacts, and σ_{ik} the mostly-silence indicator. The stream score is

$$q_{ik} = \text{clip}(1 - 0.35 \min(\max(n_{ik} - 1, 0), 2) - \alpha_{ik} - 0.3\sigma_{ik}, 0, 1)\, \text{cap}_{ik}. \tag{12}$$

The score uses $n_{ik} = 1$ for a missing count field and defaults to 0.5 before the cap when the entire report is missing. The trainer centers stream scores and adds them to the corresponding audio-token advantages with coefficient 0.5.

C.5 Exact Judge Prompts

We list the exact prompts sent to the judge models. In the ranking prompt, `{mix_report}` and `{reports}` are replaced by the JSON reports for the mixture and for every candidate track, `{ids}` by the anonymous candidate labels, and `{gt_line}` by a line stating the annotated voice count.

Track description (`DESCRIBE_TRACK_PROMPT`).

```
Listen to this single audio clip. Report ONLY what you
actually hear. Answer STRICT JSON, no other text:
{"speakers_audible": <int>, "speaker_traits":
["<gender/voice, <=6 words each>", ...], "content": "<what
is said, verbatim fragments or brief summary, <=30 words>",
"language": "<language(s)>", "artifacts":
"none|mild|severe", "mostly_silence": <bool>, "notes": "<=10
words>"}

"speakers_audible" counts DISTINCT speakers you can hear in
this clip (0 if no speech). "artifacts" = distortion,
robotic/garbled audio, clicks, unnatural sound.
```

Mixture description (`DESCRIBE_MIX_PROMPT`).

```
Listen to this audio: it is a MIXTURE that may contain
several people speaking, possibly overlapping. Report ONLY
what you actually hear. Answer STRICT JSON, no other text:
{"num_speakers": <int>, "speakers": [{"traits":
"<gender/voice, <=6 words>", "content": "<what this speaker
says, <=20 words>"}, ...], "background": "<=8 words",
"notes": "<=10 words"}
```

Text ranking (`TEXT_RANK_PROMPT`).

```
You are ranking candidate results of blind speech separation
performed on the same input mixture. You cannot hear the
audio. An independent listener described the MIXTURE, and
separately described every track of every candidate (each
track heard IN ISOLATION, without context).

MIXTURE report:
{mix_report}
{gt_line}
Candidates, labeled {ids} (order is arbitrary and
meaningless). Each candidate is one attempt to split the
mixture into one-speaker tracks:
{reports}

Rank ALL candidates into quality tiers A (clearly best) > B
> C > D (clearly worst). Priorities, most important first:
1. A track with 2+ speakers audible is NOT separated - the
more such tracks, the worse. A candidate whose single track
sounds like the whole mixture is worst.
2. The candidate's tracks together should cover all mixture
speakers/content; missing speakers or missing content is
bad.
3. Two tracks with the same speaker/content = duplicates,
bad.
4. Track content that does NOT appear in the mixture report
= likely fabricated, serious defect.
5. Fewer artifacts is better.
Ties allowed. Use multiple tiers whenever the reports differ
materially.

Answer STRICT JSON only:
{"indistinguishable": <bool>, "ranking": [{"id": "C0",
"tier": "A"}, ...one entry per candidate, every id exactly
once...], "notes": "<=20 words"}
```

D Evaluation Datasets and Metrics

D.1 Evaluation Data

LibriMix. Clean and noised two-speaker mixtures are generated with the official LibriMix scripts (Cosentino et al., 2020).

Voices in the Wild (ViW) subset. We select 3,000 single-speaker clips from Voices-in-the-Wild-2M (Xie et al., 2026). From its 54 compound acoustic scenarios, we retain twelve: five single conditions (`noise`, `echo`, `far_field`, `obstructed`, and `recording`) and seven pairs (`echo_noise`, `echo_recording`, `far_field_noise`, `far_field_recording`, `obstructed_noise`, `obstructed_recording`, and `recording_noise`). Scenarios involving `distortion` or `dropout` are excluded because they corrupt or remove speech content beyond the denoising and over-splitting behavior evaluated here. The retained conditions cover noise, reverberation, distance, occlusion, and recording-device artifacts.

We require decodable audio, a duration of 1–30 s, and a nonempty transcript, then randomly sample 250 clips per scenario (125 Mandarin and 125 English) with a fixed seed. Audio is resampled to 16 kHz mono. The resulting set spans 1.0–19.65 s per clip (median 5.11 s), totaling 5.54 h.

WildTV. WildTV consists of 46 real-world speech clips from several domains, including film and TV drama and podcasts, each with one to three speakers: 11 with one speaker, 25 with two, and 10 with three. Reference labels (speaker counts and per-speaker transcripts) are annotated with MOSS-Transcribe-Diarize (MOSI.AI, 2026) and then manually reviewed and corrected. Soloist and SR-CorrNet (the released SR-CorrNet-SS checkpoint) each use a single model with blind count inference. SepFormer, MossFormer, and ConvTasNet switch checkpoints using the true count: enhancement for one speaker, Libri2Mix for two, and WSJ0Mix for three. Their perfect output-count agreement follows from this construction.

Singing benchmarks. SongYT contains 3,000 song recordings in Chinese, English, and Japanese, so it tests whether separation preserves vocals across languages and regional singing styles. SongYT uses vocal pseudo-references produced by Mel-Band RoFormer (Wang et al., 2023b). Agreement with these references measures consistency with the teacher estimate. MIR-1K (Hsu & Jang, 2010) contains 1,000 clips of 4–13 s cut from 110 Chinese karaoke pop songs sung by 19 mostly amateur singers. Accompaniment and singing voice are recorded on the left and right channels, so every clip has an isolated vocal reference. MUSDB18-HQ (Rafii et al., 2019) is the uncompressed version of MUSDB18: 150 full-length stereo tracks of various genres (about 10 h), each with isolated vocals, drums, bass, and other stems. MUSDB18-HQ has no lyric references, so we report only RPA and SingerSim on it.

D.2 Metrics and Aggregation

Transcription and PI-WER. Separated outputs are resampled to 16 kHz and transcribed with Whisper large-v3 (Radford et al., 2023). PI-WER uses the padded one-to-one assignment in Equation (7). A real reference paired with a dummy output contributes its entire token count as deletions; a real output paired with a dummy reference contributes its entire token count as insertions. The denominator contains reference tokens only. Text is normalized with Unicode NFKC and case folding. CJK characters, kana, and Hangul are tokenized character by character, while Latin text is tokenized into words. Punctuation separates tokens, except that internal apostrophes are retained. Thus, the reported WER uses script-dependent word/character units. Count accuracy compares the number of voiced output streams with the reference speaker count; the declared count is stored separately as `stated`.

Corpus-level WER is a micro-average: total edit errors divided by total reference tokens, rather than the mean of clip-level WERs.

Speech quality. We use DNSMOS on ViW to measure how much background noise remains after separation. ViW clips carry noise, reverberation, and other acoustic degradations, and WER alone cannot show whether a system removes the noise or only keeps the speech intelligible. DNSMOS (Reddy et al., 2022) is a non-intrusive neural predictor of ITU-T P.835 subjective ratings. Its background-noise score (BAK) rates how intrusive the remaining background is, and its overall score (OVRL) rates overall quality; both use a 1–5 scale with higher values better. We report means and medians over the ViW clips.

Music similarity and pitch. SingerSim uses layer-14 features from WavLM-large (Chen et al., 2022), with L2 normalization on RMS-active frames followed by mean pooling. RPA (raw pitch accuracy) is the fraction of voiced reference frames whose estimated F0 lies within 50 cents of the reference F0. Transcript-based metrics are insensitive to melody: a separated vocal can keep every lyric while its pitch drifts. This matters for a generative separator, which re-synthesizes the vocal rather than masking the mixture. RPA therefore tests whether separation preserves the sung melody, complementing WER for lyric content and SingerSim for vocal similarity.

E Full-Duplex Application

The pipeline converts a conversation recording with K speakers into K continuous speaker tracks with word-aligned transcripts; our examples use $K = 2$ and $K = 3$. Its input is the recording and a speaker timeline with transcripts from MOSS-Transcribe-Diarize (MOSI.AI, 2026). Processing follows four stages: window mining and planning (Appendix E.1), separation and assembly (Appendix E.2), quality control (Appendix E.3), and acceptance (Appendix E.4). Table 15 lists the pipeline parameters. The supplementary material contains audio examples processed by the pipeline: eight two-speaker conversations (four Chinese and four English) and six Chinese three-speaker conversations with three output tracks, each with the input mixture, the separated speaker tracks, and aligned transcripts.

Table 15: Full-duplex pipeline parameters.

Stage	Parameter	Value
Window mining	Window length	60–180 s; K speakers
	Minimum speech per speaker	0.2 s
	Minimum exclusive speech per speaker	1 s
	Minimum speech density	0.45
	Boundary silence gap / outer padding	$\geq$0.25 s / 0.2 s
	Excluded transcript tail	Final 2 s
	Overlap stratum ($K = 2$)	Longest annotated intersection $\geq$0.30 s
	Near-duplicate rejection	Temporal IoU $>$0.30 within a recording
	Single-block rejection	$\leq$2 intervals cover $>$90% of a span that is $\geq$15 s or $>$20% of the window
Piece planning	Maximum piece length / streams	22 s / min$(K, 3)$
	Context and speaker-inclusion margin	0.5 s per side
	Preferred minimum length (1/2/3 streams)	3/6/8 s; penalty 4.0 per missing second per stream
Assembly	Cross-piece consistency weight	2.0
	Mixed-stream rejection	Highest similarity $\geq$0.45 and second-highest $\geq \max(0.45,$ reference similarity $+ 0.10)$
	Alignment refinement	$\pm$6 s; confidence $\geq$0.05
	Timeline-mask expansion	0.2 s per side
Repair	Stopping rule	Round limit, or empty-slot reduction $<$10%
	Minimum exclusive region for comparison	0.5 s
Intrusive speech	Intrusion margin	Other-speaker similarity higher by $\geq$0.10
	Utterance removal	$\geq$70% scrubbed
	Purity rejection	Unresolved minority speech (verifier score gap within 0.10) $\geq$2.0 s
	Noise-floor fill (all zeroed regions)	2nd-percentile speech-frame energy + 3 dB (source frames below + 6 dB)
Content	Envelope-correlation threshold	0.30 (amplitude fallback 0.12)
	Minimum exclusive speech for assessment	0.30 s
Acceptance	Per-speaker completeness	$\geq$0.80
	Least-speaking speaker	$\geq$8 s and $\geq$5 utterances
	Hole gate	Fraction $>$0.20 and missing duration $>$1.0 s

E.1 Window Mining and Planning

Window mining. The pipeline first selects conversation windows that are long enough to capture turn-taking and reliable enough to annotate. Each window contains exactly K speakers, gives each speaker enough speech to keep short backchannels, including at least 1 s of exclusive speech, and is dense in speech. Boundaries fall in silence gaps and avoid the end of a possibly truncated transcript. Windows with ambiguous multi-speaker labels, near-duplicates of another window from the same recording, or a suspicious single-block speaker annotation are discarded. For $K = 2$, each retained window is assigned to the overlap or turn-taking stratum according to whether the timeline contains

a sufficiently long speaker intersection; overlap that the timeline misses is therefore not reflected in the strata. Windows with $K = 3$ form a single stratum.

Piece planning. A conversation window is too long to separate in one pass, so it is split into inference pieces. A dynamic program chooses pieces that cover all annotated speech while minimizing the sum of piece duration times requested stream count, a proxy for the number of generated audio tokens. Each piece requests one stream per speaker active in it, rather than one per speaker in the whole window, and adds context on both sides for later cross-fading. Long monologues are split before planning, and a soft penalty, proportional to the missing duration and the stream count, discourages pieces shorter than the preferred length for their stream count. A piece requests at most $\min(K, 3)$ streams.

E.2 Separation and Assembly

Separation and speaker assignment. Soloist separates every planned piece, including single-speaker pieces. CAM++ embeddings (Wang et al., 2023a) then assign the local streams to the K conversation speakers, and a cross-piece consistency term keeps assignments stable across neighboring pieces. A stream is rejected as mixed when its second-highest speaker similarity is also high; the margin adapts to the similarity between the speaker references, so that speakers with similar voices are not rejected wholesale.

Placement. Each assigned stream is placed on the recording time axis relative to its speaker's timeline onset. When its confidence is high enough, energy-envelope alignment refines the offset within a bounded range. Neighboring pieces are cross-faded over their shared context. Timeline masks, slightly expanded but never entering other speakers' intervals, remove residual speech outside the speaker's own activity, and all zeroed regions are filled with a low-level noise floor sampled from the track's quietest frames. Speaker verification over each speaker's full timeline intervals then checks the assembled assignment. For $K = 2$, the two tracks are swapped only when both assignments are reversed; for $K \geq 3$, the Hungarian algorithm reassigns tracks whenever a better permutation exists. Streams that remain misassigned after this check are discarded.

Repair. Pieces that return too few usable streams, and streams rejected during assignment, are re-queried. The repair window keeps the active-speaker set and strictly contains the previous interval, so a replacement cannot open new coverage gaps. For a piece that already had enough streams, a replacement is accepted only if it improves separation, judged by the energy contrast between speaker streams and by envelope agreement in their exclusive regions; when those regions are too short to compare, the existing output is kept. For a piece with missing streams, a replacement is accepted once it provides enough voiced streams. Repair stops at a round limit or when the number of empty slots stops decreasing appreciably, and unrecoverable streams remain explicitly marked as missing.

E.3 Quality Control

Each assembled track is checked for three failure modes, each requiring different evidence.

Intrusive speech. MOSS-Transcribe-Diarize re-diarizes each track. When it detects more than one speaker, speaker verification localizes the intrusive intervals with CAM++ and ERes2Net (Chen et al., 2023); a region is kept only when both verifiers accept the target speaker, and scrubbed intervals are filled with the noise floor. An utterance that is mostly scrubbed is removed from the timeline. The track fails the purity check if minority-speaker speech that the verifiers cannot resolve (score gap within 0.10) accumulates beyond the threshold, or if verification cannot cover it. Combining diarization with verification separates true intrusions from diarization over-segmenting a single speaker.

Missing speech. Hole detection compares the track with the mixture over the source speaker timeline. The hole fraction is measured over frames in the speaker's exclusive regions where the mixture has speech-level energy.

Mismatched content. A track can carry the right voice with the wrong words. Qwen3-ASR-1.7B and Qwen3-ForcedAligner-0.6B (Shi et al., 2026) produce the track transcript and word timestamps, and the word count is compared with MOSS-Transcribe-Diarize. Within exclusive-speaker regions, an utterance whose normalized energy envelope correlates poorly with the mixture is marked as mismatched, with an amplitude-domain fallback. Instead of discarding the whole conversation, the pipeline cuts out mismatched utterances by splitting all tracks and the mixture at those utterances into separate samples; pieces shorter than 20 s are dropped, so final samples can be shorter than 60 s. Regions with too little exclusive speech are marked unassessable and reported separately.

E.4 Acceptance and Output

A conversation is accepted when every speaker has a track and transcript, every track passes the purity check, each speaker's completeness reaches the threshold, the speaker with the least speech contributes enough, and no track fails the hole gate. The hole gate requires both a large hole fraction and a long missing duration, so very short exclusive regions are not rejected on their ratio alone. Completeness is an automatic proxy that can exceed one, so it is used only for coarse filtering. A batch must also stay within the failure-rate limits in Table 16, and every delivered track must be non-silent. Listening to stress cases complements these automatic checks.

Table 16: Batch acceptance thresholds. Denominators are specified per row; limits are configured thresholds, not measured rates.

Check	Failure event	Denominator	Limit
Inference	Fewer usable streams than requested	Planned pieces	3%
Assembly: missing	No assigned stream after repair	Speaker slots per piece	4%
Assembly: placement	Assigned stream silent in its exclusive region	Speaker slots per piece	6%
Speech coverage	Track more than 40 dB below mixture during timeline speech	Source-timeline speech duration	5%
Speaker coverage	Unprocessed speaker with more than 3 s of speech	Evaluated conversation windows	1%
Content matching	Envelope below 0.30 or fallback discrepancy above 0.12	Assessable utterances	8%
Silent output	Track RMS below -60 dBFS	Delivered tracks	0

Each accepted conversation provides one 24 kHz mono track per speaker (and, for two speakers, a stereo version with one speaker per channel), together with speaker intervals, utterances, and word timestamps on the same time axis. Speaker assignments are marked as model-derived. A filtered export applies all gates, while an unfiltered export keeps the hole, purity, and completeness metadata for downstream selection.

F Extended Related Work

F.1 Supervised Separation and Variable Source Counts

Paired-source separation learns to recover isolated waveforms from constructed mixtures. Conv-TasNet operates in a learned time-domain representation (Luo & Mesgarani, 2019); SepFormer and MossFormer develop attention-based sequence models (Subakan et al., 2021; Zhao & Ma, 2023); and TF-GridNet combines full-band and sub-band modeling in the time-frequency domain (Wang et al., 2023c). These approaches establish strong reconstruction baselines, while the choice of training mixtures determines the acoustic conditions and source combinations available to supervision. LibriMix provides a reproducible construction for studying speech separation and generalization (Cosentino et al., 2020).

Output assignment and source count are distinct problems. Permutation-invariant training handles the arbitrary ordering of reference sources (Yu et al., 2017; Kolbæk et al., 2017), whereas recursive separation extracts speakers successively to accommodate unknown counts (Takahashi et al., 2019). SR-CorrNet instead adapts the number of output streams with an attractor-based split module (Shin & Park, 2026). Soloist represents count, stream boundaries, and voice audio in one serial response. Its

supplied-count and blind evaluations therefore probe different capabilities: recovering voices under a known count and deciding how many streams to produce (Section 5).

F.2 Neural Audio Codecs and Generative Separation

SoundStream and EnCodec learn compressed audio representations that can be decoded back into waveforms (Zeghidour et al., 2022; Défossez et al., 2023). AudioLM shows how discrete audio representations support language-model-style generation (Borsos et al., 2023). This representation makes separation amenable to conditional sequence prediction, alongside tasks such as synthesis and recognition. TokenSplit is a direct precedent: it jointly handles speech separation, transcription, and speech generation through discrete representations, with both direct separation and refinement of a conventional separator's outputs, and with or without transcript conditioning (Erdogan et al., 2023).

Soloist uses codec sequences as its separation output, conditioned on continuous audio features and discrete mixture codes (Section 2). Its backbone predicts the response structure and conditions a local decoder for each frame's codec codes. The relevant distinction is the combination of variable voice-stream generation, speaker-timeline supervision, and adaptation on real mixtures through candidate feedback. Generating plausible audio and preserving the particular voices and words in a mixture remain separate requirements, motivating the content and identity checks used in our evaluation and application.

F.3 Prompted Extraction and Singing-Voice Separation

Target extraction specifies which sound to recover through an external cue. AudioSep uses natural-language descriptions (Liu et al., 2025); SoloAudio performs generative extraction with a diffusion Transformer and supports audio and language conditioning (Wang et al., 2025); and SAM Audio supports text, visual, and temporal prompts (Shi et al., 2025). These interfaces allow a user to select a sound of interest. Blind voice separation instead requires enumerating the audible voices without a target query. Our comparisons make the supplied-count setting explicit because count information and target selection change what a system must infer.

Music separation introduces sustained pitch, accompaniment, and potentially layered vocals. Mel-Band RoFormer models overlapping mel-frequency bands for music source separation (Wang et al., 2023b). MIR-1K and MUSDB18-HQ provide established singing and music evaluation resources (Hsu & Jang, 2010; Rafii et al., 2019). We use a fixed music separator to produce song pseudo-vocals and train these as one vocal stream (Section 3). This supervision can represent multiple singers in a combined vocal target, whereas speech examples teach individual voice streams. The two target definitions determine what each evaluation can establish.

F.4 Learning from Unpaired Recordings and Teacher Targets

MixIT removes the need for isolated training sources by separating a mixture of mixtures and assigning estimated components back to the observed mixtures (Wisdom et al., 2020). Its application to real meetings demonstrates adaptation to far-field, overlapping, reverberant speech without isolated in-domain references (Sivaraman et al., 2022). RemixIT follows a teacher–student approach: estimated speech and noise are remixed to create training pairs, and the teacher is updated from successive students (Tzinis et al., 2022). These methods provide alternatives to relying entirely on constructed mixtures with known sources.

Our supervision is organized by the reference available for each recording. Constructed mixtures provide source audio, songs provide fixed-teacher pseudo-vocals, and natural recordings provide speaker timelines. Subsequent RL compares candidate separations of the same real mixture using a model judge (Section 4). The distinction concerns the training signal: mixture reconstruction, teacher-generated audio, speaker-activity labels, and relative quality feedback impose different constraints on the outputs.

F.5 Speaker Activity, Diarization, and Identity

Diarization estimates who speaks when, including overlapping activity. End-to-end neural diarization formulates this as multi-label prediction with a permutation-free objective and supports adapta-

tion from speaker-segment labels (Fujita et al., 2019). Such labels describe temporal activity without specifying each speaker's waveform. MOSS-Transcribe-Diarize supplies the speaker-attributed intervals used in our timeline supervision (MOSI.AI, 2026). We encode those intervals as a text response within the same model that generates separated audio, allowing natural recordings to contribute supervised examples without isolated targets.

Activity labels and speaker identity serve different roles in long recordings. Speaker-verification systems such as CAM++ and ERes2Net learn representations for comparing voices (Wang et al., 2023a; Chen et al., 2023). Our conversation pipeline uses speaker embeddings and consistency across pieces to associate local outputs with recording-level speakers, then combines activity and identity checks to detect missing or intruding speech (Appendix E). Agreement on identity alone does not establish that the original words have been preserved, which motivates the separate content checks.

F.6 AI Feedback, Policy Optimization, and Judge Reliability

RLHF uses human preferences to train language models beyond supervised demonstrations (Ouyang et al., 2022), while RLAIF studies preferences supplied by AI systems (Lee et al., 2024). PPO supplies a clipped policy-gradient objective (Schulman et al., 2017); GRPO in DeepSeekMath and the optimization components of DAPO develop group-based feedback and practical training procedures for language-model RL (Shao et al., 2024; Yu et al., 2025). Our update uses relative rewards within a group of candidate separations, with an unnormalized group-centered advantage and signal-based shaping (Section 4.2).

Audio-specific feedback learning provides closer precedents. MOS-based RLHF fine-tunes speech enhancers using a learned perceptual-quality reward (Kumar et al., 2025). MARS-Sep optimizes a mask policy with a multimodal preference reward for query-conditioned sound separation (Zhang et al., 2026). LLM-guided audio-visual enhancement converts audio-language descriptions into scalar ratings for PPO updates (Chen et al., 2026). These studies connect feedback to audio processing through quality prediction, cross-modal alignment, or textual descriptions. Our setting combines blind generation of multiple voice streams with comparison of complete candidate separations of the same recording. The feedback must account for voice purity and coverage across all outputs, as well as each track's quality.

The quality of the feedback is a separate concern from the optimizer. LLM-as-a-judge studies identify presentation-order bias and other limitations in model judgments (Zheng et al., 2023). Reward-model overoptimization can increase a proxy score while degrading the quality it is intended to measure (Gao et al., 2023). Our judge first describes clips independently, then compares their reports alongside an anonymous mixture anchor and filters inconsistent rankings under shuffled and reversed candidate orders. The consistency check targets sensitivity to candidate order, and the signal checks target degenerate outputs.

F.7 Evaluating Content, Acoustics, Pitch, and Identity

Waveform-based measures such as BSS Eval and SI-SDR quantify aspects of reconstruction fidelity relative to source signals (Vincent et al., 2006; Le Roux et al., 2019). Recognition-based evaluation instead asks whether the separated speech retains recoverable words. Permutation-aware multi-speaker ASR and work on WER definitions provide the basis for matching streams before counting transcription errors (Qian et al., 2018; Sklyar et al., 2021; von Neumann et al., 2023). Our PI-WER protocol pads unequal stream sets with empty transcripts and evaluates the minimum-cost assignment; count accuracy separately assesses the number of voiced outputs.

Speech representations such as WavLM are trained to support tasks involving both content and speaker information (Chen et al., 2022). Our SingerSim metric uses WavLM features, while raw pitch accuracy measures pitch preservation (Appendix D.2). These quantities complement transcription recovery: correct words do not imply correct pitch or voice identity, and high embedding similarity need not imply exact words. Reference provenance also matters. Comparisons against teacher-extracted vocals measure agreement with that teacher, while isolated-source references support a different fidelity interpretation. We therefore distinguish these dimensions and reference types rather than treating one score as a complete measure of separation.

F.8 Continuous Separation and Full-Duplex Conversation Data

Continuous speech separation converts long recordings containing overlap into continuous output channels suitable for recognition. Speaker-independent CSS allows successive utterances to occupy different channels while keeping each utterance intact (Yoshioka et al., 2019). Our application requires an additional association step: independently generated pieces must be linked to persistent speaker identities and placed on the original recording timeline. Window planning, stream assignment, alignment, and targeted repair jointly determine whether the assembled tracks preserve coverage (Section 6).

Full-duplex dialogue models such as Moshi model user and system speech in parallel streams to support overlap and conversational timing (Défossez et al., 2024). This creates a need for temporally aligned conversational audio. Our pipeline prepares continuous speaker tracks and aligned transcripts offline from existing recordings. It checks speaker purity, missing speech, and content consistency before packaging the outputs.