

Position-Aware Target Speaker Extraction for Long-Form Multi-Party Conversations: A Diarization-Free Framework for ASR

Yichi Wang¹, Junzhe Chen¹, Wangjin Zhou¹, Tatsuya Kawahara¹

¹ Graduate School of Informatics, Kyoto University, Kyoto, Japan

{yiwang, juchen, zhou, kawahara}@sap.ist.i.kyoto-u.ac.jp

Abstract

In long-form multi-party conversations, highly imbalanced speaker activity and frequent overlap make it difficult to identify “who spoke when and what”. Sliding-window continuous speech separation (CSS) mitigates sparse supervision, but often suffers from cross-window speaker inconsistency and residual crosstalk, which in practice requires diarization for reliable speaker attribution. Motivated by the stability of speakers’ directions of arrival (DOAs) in meetings, we propose **PATSE**, a multi-channel **Position-Aware Target Speaker Extraction** front-end that uses DOA as a spatial prior to directly extract the speech of each target speaker. PATSE combines a DOA-guided spatial encoder and conditioner to generate speaker-attributed streams, from which speaker activity can be inferred via simple post-processing (e.g., VAD) without explicit diarization. Experiments on both replayed and real conversations show consistent ASR gains outperforming CSS and diarization-based pipelines. **Index Terms**: target speaker extraction, direction of arrival, multi-party conversation

1. Introduction

In multi-party conversations such as meetings and discussions, the fundamental problem is to identify “who spoke when and what” [1, 2, 3]. Real-world recordings are typically long and continuous, where spontaneous conversations results in highly imbalanced speaker activity, and back-channel responses and brief interruptions cause frequent speech overlap. Since most automatic speech recognition (ASR) systems are optimized for short single-speaker utterances, a widely-used strategy is to introduce a speaker-aware separation front-end to enable speaker-attributed transcription.

Most separation systems allocate one output per speaker. For long recordings, however, many speakers remain inactive for long periods, yielding temporally sparse supervision and unstable training [4]. Continuous speech separation (CSS) [5] mitigates this issue by applying separation in sliding windows where only a subset of speakers are active. However, consistent speaker identity across discontinuous windows remains difficult: similarity-based linking is local, while separation alone does not enforce long-range identity consistency, so speaker diarization (SD) is typically required for reliable assignment. In addition, CSS is prone to producing crosstalk in single-speaker regions [6, 7], causing insertion errors in downstream ASR.

The interaction between separation and diarization remains an open problem [8, 9]. Some approaches perform diarization prior to separation [10], as in guided source separation [11, 12]. However, accurate estimation of speaker temporal boundaries in overlapping speech remains challenging [13].

Since ideal separation naturally captures speaker activity,

joint modeling has been explored to overcome cascaded limitations [14]. As a form of joint modeling, we formulate the front-end as target speaker extraction (TSE). TSE extracts a specified speaker’s speech conditioned on prior target information [15, 16, 17], such as speaker embeddings or spatial clues. Applying TSE independently to each target yields speaker-attributed streams, which (i) preserves consistent speaker tracking without explicit diarization, and (ii) avoids inter-output crosstalk by producing a single output per target speaker.

Nevertheless, speaker embedding-based TSE has two inherent drawbacks: reliance on enrollment audio [10, 18], which is often unavailable in practice, and mismatch between fixed reference embeddings and time-varying speaker characteristics. Although embeddings can be derived from diarization-based single-speaker segments [19], noise and interference in these segments may distort the resulting embeddings. In contrast, spatial cues such as direction of arrival (DOA) provide a more stable and explicit prior [20]. In typical meetings, speakers remain largely stationary, yielding relatively consistent DOAs that can be obtained via microphone arrays or cameras. These observations motivate multi-channel position-aware TSE as a practical front-end design for long-form multi-party ASR, extending prior DOA-based TSE studies [16, 21, 22] beyond simulated short utterances.

Following are the main contributions of this paper:

- We propose PATSE, a multi-channel position-aware TSE front-end for multi-party ASR, featuring a DOA-guided spatial encoder and conditioning module that injects target-specific spatial features into the separation backbone. PATSE produces speaker-attributed streams, from which speaker activity can be inferred through simple post-processing (e.g., voice activity detection, VAD) without explicit diarization.
- We build and release LibriReplay-DOA¹, a real-room playback dataset with ground-truth DOA annotations to benchmark DOA-based methods. Experiments on LibriReplay-DOA and the real-world conversational TEIDAN dataset [23] demonstrate consistent improvements in downstream ASR. Detailed case studies are available on our demo page².

2. Proposed Method

2.1. Architecture

As shown in Figure 1, PATSE treats DOA as an explicit cue for “who” and conditions the separation backbone on this spatial prior. Let $\mathbf{x} = \{\mathbf{x}_m\}_{m=1}^M$ denote the multi-channel audio signal captured by M microphones, where m indexes the channels. The azimuth angle θ_{tgt} denotes the DOA of the target speaker.

¹<https://huggingface.co/datasets/real-recordings/LibriReplay-DOA>

²<https://exp-demos.github.io/PATSE-audio-demo>

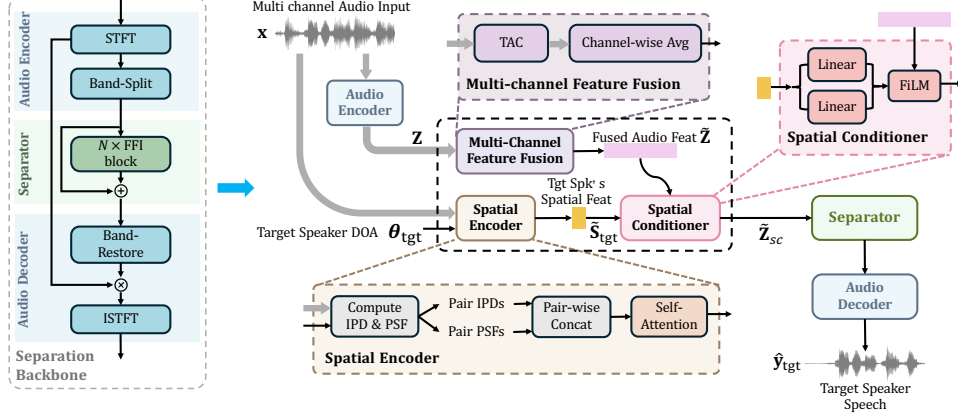


Figure 1: Overall Architecture of the PATSE Framework.

PATSE performs DOA-conditioned extraction via the separation backbone (audio encoder with multi-channel feature fusion (MCFF), separator, and audio decoder), along with a spatial encoder and spatial conditioner:

$$\begin{aligned}\tilde{\mathbf{Z}} &= \text{MCFF}(\text{AudioEnc}(\mathbf{x})), \quad \tilde{\mathbf{S}}_{\text{tgt}} = \text{SpatialEnc}(\mathbf{x}, \theta_{\text{tgt}}), \\ \tilde{\mathbf{Z}}_{sc} &= \text{SpatialConditioner}(\tilde{\mathbf{Z}}, \tilde{\mathbf{S}}_{\text{tgt}}), \\ \hat{\mathbf{y}}_{\text{tgt}} &= \text{AudioDec}(\text{Separator}(\tilde{\mathbf{Z}}_{sc})),\end{aligned}\quad (1)$$

where $\tilde{\mathbf{Z}}$ denotes the encoded audio features, $\tilde{\mathbf{S}}_{\text{tgt}}$ the target-specific spatial features, $\tilde{\mathbf{Z}}_{sc}$ the spatially conditioned features, and $\hat{\mathbf{y}}_{\text{tgt}}$ the reconstructed waveform of the target speaker.

2.1.1. Separation Backbone

In this work, we adopt TIGER [24] as the separation backbone, an encoder–separator–decoder architecture originally designed for single-channel input.

The shared **audio encoder** operates on each channel independently. For channel m , the input waveform $\mathbf{x}_m$ is first transformed into the short-time Fourier transform (STFT) representation $\mathbf{X}_m \in \mathbb{R}^{F \times T}$, where F and T denote the numbers of frequency bins and time frames, respectively. The STFT $\mathbf{X}_m$ is then passed through the band-split module to obtain sub-band features $\mathbf{Z}_m \in \mathbb{R}^{N \times K \times T}$, where K is the number of sub-bands and N is the feature dimension per sub-band. Collecting all channels yields multi-channel audio features $\mathbf{Z} = \{\mathbf{Z}_m\}_{m=1}^M$.

The **separator** then stacks frequency–frame interleaved (FFI) blocks with residual connections.

The **audio decoder** finally reconstructs the time-domain waveform $\hat{\mathbf{y}}_{\text{tgt}}$ via band-restore and inverse STFT (iSTFT).

2.1.2. Multi-Channel Feature Fusion (MCFF)

To adapt multi-channel features to a monaural separator, MCFF fuses $\mathbf{Z}$ into a unified single-stream representation $\tilde{\mathbf{Z}}$.

Specifically, MCFF first adopts a transform–average–concatenate (TAC) [25] strategy, where a learnable transform $\mathcal{T}(\cdot)$ is applied to each channel and averaged to form a global descriptor $\mathbf{h}$:

$$\mathbf{h} = \frac{1}{M} \sum_{m=1}^M \mathcal{T}(\mathbf{Z}_m). \quad (2)$$

The per-channel features $\mathbf{Z}_m$ are concatenated with the global

descriptor $\mathbf{h}$ and fused through a mapping $\mathcal{F}(\cdot)$:

$$\mathbf{Z}'_m = \mathcal{F}(\text{Concat}(\mathbf{Z}_m, \mathbf{h})) + \mathbf{Z}_m, \quad (3)$$

where $\mathbf{Z}'_m$ denotes the globally fused feature of channel m . The final single-stream encoded audio features $\tilde{\mathbf{Z}}$ are obtained by channel averaging:

$$\tilde{\mathbf{Z}} = \frac{1}{M} \sum_{m=1}^M \mathbf{Z}'_m, \quad \tilde{\mathbf{Z}} \in \mathbb{R}^{N \times K \times T}. \quad (4)$$

2.1.3. Spatial Encoder

Interaural Phase Difference (IPD): For each microphone pair $p = (i, j)$ ($1 \leq i < j \leq M$), the IPD is defined as:

$$\text{IPD}^{(p)}(t, f) = \angle X_i(t, f) - \angle X_j(t, f). \quad (5)$$

Here $t = 1, \dots, T$ and $f = 1, \dots, F$ denote the time-frame and frequency-bin indices, respectively.

To avoid phase wrapping ambiguity [26], it is encoded as:

$$\phi^{(p)}(t, f) = [\cos(\text{IPD}^{(p)}(t, f)), \sin(\text{IPD}^{(p)}(t, f))]. \quad (6)$$

Theoretical Phase Difference (TPD): TPD denotes the theoretical phase difference induced by a source located at DOA θ_{tgt} under the far-field assumption, defined as [26]:

$$\text{TPD}_{\text{tgt}}^{(p)}(f) = \frac{2\pi f_s f}{2(F-1)c} \cos \theta_{\text{tgt}} d_{i,j}, \quad (7)$$

where f_s denotes the sampling rate of the waveform, $d_{i,j}$ the distance between microphone i and j , and c the speed of sound.

Phase Similarity Feature (PSF): A phase similarity feature between TPD and IPD is defined to quantify the dominance of the target speaker at each time–frequency (t, f) bin [21]:

$$\text{PSF}_{\text{tgt}}^{(p)}(t, f) = \begin{bmatrix} \sin(\text{TPD}_{\text{tgt}}^{(p)}(f) - \text{IPD}^{(p)}(t, f)), \\ \cos(\text{TPD}_{\text{tgt}}^{(p)}(f) - \text{IPD}^{(p)}(t, f)) \end{bmatrix}. \quad (8)$$

It is concatenated with $\phi^{(p)}(t, f)$ to form the spatial feature:

$$\mathbf{S}_{\text{tgt}}^{(p)}(t, f) = [\phi^{(p)}(t, f), \text{PSF}_{\text{tgt}}^{(p)}(t, f)]. \quad (9)$$

Stacking all $P = \frac{M(M-1)}{2}$ pairs yields $\mathbf{S}_{\text{tgt}} = \{\mathbf{S}_{\text{tgt}}^{(p)}\}_{p=1}^P$.

Following the band-split strategy, stacked self-attention is applied within each band for refinement, and the outputs are aggregated to obtain the final target-specific spatial features $\tilde{\mathbf{S}}_{\text{tgt}}$.

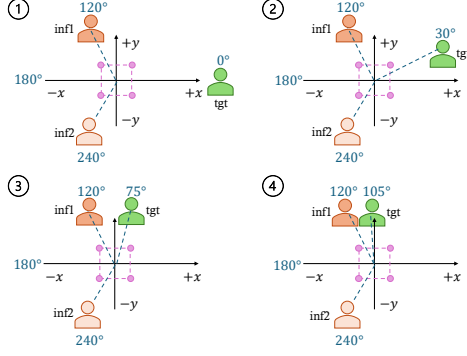


Figure 2: Four Spatial Configurations for LibriReplay-DOA.

2.1.4. Spatial Conditioner

We adopt feature-wise linear modulation (FiLM) [27], where a linear generator $\mathcal{L}$ produces modulation parameters γ and β from $\hat{\mathbf{S}}_{\text{tgt}}$, which are used to modulate $\hat{\mathbf{Z}}$ to obtain $\hat{\mathbf{Z}}_{\text{sc}}$:

$$(\gamma, \beta) = \mathcal{L}(\hat{\mathbf{S}}_{\text{tgt}}), \quad \gamma, \beta \in \mathbb{R}^{N \times K \times T}, \quad (10)$$

$$\hat{\mathbf{Z}}_{\text{sc}} = \gamma \odot \hat{\mathbf{Z}} + \beta, \quad (11)$$

where $\odot$ denotes element-wise multiplication.

2.2. Training Objective

To better supervise the model under different target activity states (speech vs. silence), we adopt the activity-aware loss in [28]. Based on the reference waveform $\mathbf{y}_{\text{tgt}}$, it is segmented into silent and non-silent regions, and the same segmentation is applied to the estimated waveform $\hat{\mathbf{y}}_{\text{tgt}}$, yielding aligned pairs $(\hat{\mathbf{y}}_{\text{tgt}}^{\text{sil}}, \mathbf{y}_{\text{tgt}}^{\text{sil}})$ and $(\hat{\mathbf{y}}_{\text{tgt}}^{\text{n-sil}}, \mathbf{y}_{\text{tgt}}^{\text{n-sil}})$.

Silent regions: The residual log-energy loss L_E is defined as:

$$L_E = 10 \log_{10}(\|\hat{\mathbf{y}}_{\text{tgt}}^{\text{sil}}\|^2 + \epsilon), \quad (12)$$

where ϵ is a small constant for numerical stability.

Non-silent regions: The signal-to-noise ratio (SNR) loss L_S is defined as:

$$L_S = -10 \log_{10}\left(\frac{\|\mathbf{y}_{\text{tgt}}^{\text{n-sil}}\|^2}{\|\hat{\mathbf{y}}_{\text{tgt}}^{\text{n-sil}} - \mathbf{y}_{\text{tgt}}^{\text{n-sil}}\|^2 + \epsilon}\right). \quad (13)$$

The **overall loss** L is defined as:

$$L = \alpha L_E + L_S, \quad (14)$$

where α is a scaling factor balancing the L_E and L_S .

3. Experiments

3.1. Datasets

LibriReplay-DOA Dataset. Most speech separation benchmarks rely on simulated Room Impulse Responses (RIRs) that may not fully reflect real-world conditions, while real-world corpora often lack ground-truth DOA labels. To evaluate DOA-based methods reliably, we construct LibriReplay-DOA, a real-room playback dataset with ground-truth DOA annotations.

Multi-party conversations were simulated by concatenating LibriSpeech utterances [29] with inserted silences for each speaker and replaying them through loudspeakers in a real room. A total of 114 playback sessions (about 1 minute each) were created, each involving two or three speakers, with overlap ratios of 0–25% (30), 25–50% (28), 50–75% (30), and 75–100% (26). Recordings were captured using a four-channel circular array (ReSpeaker Mic Array V2.0, radius 0.032 m), and DOA labels were given by the physical loudspeaker directions.

Table 1: Comparison of different front-end methods.

Method	DOA	Chunk	Stitch	Spk. ID	VAD
DSB + Gate	✓	None	–	–	Silero
FastMNMF	×	None	–	Oracle	Silero
Sortformer + GSS	×	SD	–	SD	–
CSS (FasNet-TAC)	×	Uniform	MSE	Oracle	–
CSS (TIGER)	×	Uniform	MSE	Oracle	–
PATSE	✓	Uniform	Direct	–	Silero

To evaluate spatial robustness, four angular configurations were designed by varying the target–interferer angle (15°, 45°, 90°, 120°) while fixing the interferer–interferer angle at 120°, as shown in Figure 2 (in two-speaker sessions, inf2 remained silent throughout). Each session was recorded under all four angular configurations, yielding 456 recordings (≈ 7 hours in total).

TEIDAN Dataset. TEIDAN is a real-world triadic dialogue corpus [23], where three participants sit around a circular table (about 120° apart) with a four-channel circular microphone array (radius 0.032 m) placed at the center. Experiments are conducted on the English subset (Sessions 01–05), which comprises approximately 2.5 hours of spontaneous dialogue, with each recording lasting about 10 minutes and overlapping speech accounting for 31.98% of the total duration on average.

Training Data. Training (50 h) and validation (6 h) data were generated following the same dialogue simulation procedure as in LibriReplay-DOA. For spatialization, RIRs were simulated using the `gpuRIR` toolbox [30]. Rooms were randomly sampled with sizes ranging from $3 \times 3 \times 2.4$ m to $10 \times 8 \times 4$ m and RT60 values between 0.2–0.8 s. A fixed four-channel circular array (radius 0.032 m) was placed with random position and orientation. Speakers were positioned around a virtual table with inter-speaker distances (≥ 0.3 m), and relative speaker SNRs were sampled from 0–5 dB. Noise samples from the DNS corpus [31] were added as additional point sources, and the mixture SNR was uniformly sampled from 0–20 dB.

3.2. Experimental Setup

Baselines. Representative front-end and pipeline baselines are compared. *DSB+Gate* applies delay-and-sum beamforming followed by frame-level energy gating. *FastMNMF* [32] is a multi-channel nonnegative matrix factorization (MNMF)-based blind source separation method. *Sortformer+GSS* performs speaker diarization using Sortformer [33] followed by guided source separation (GSS) [34]. For CSS-based systems, only the separation backbone is employed, excluding the proposed spatial encoder and conditioning modules (Figure 1). FasNet-TAC [25] and TIGER extended with MCFF for multi-channel input are adopted as separation backbones, and oracle speaker assignment is provided for all CSS-based methods.

System Configuration. Table 1 summarizes the configurations of all compared systems. The *DOA* column indicates whether prior speaker DOA information is required. *Chunk* describes how long-form recordings are segmented: no segmentation (*None*), where the full recording is processed as a whole; diarization-based segmentation (*SD*); or uniform segmentation using a 4 s window with 2 s overlap (*Uniform*). *Stitch* denotes cross-chunk permutation alignment, which is based on mean squared error (*MSE*) for CSS systems, whereas *Direct* refers to permutation-free concatenation, as adopted in PATSE. In this setup, CSS is configured to produce three output streams, enabling separation of up to three overlapping speakers. *Spk. ID*

Table 2: WER (%)↓ on the LibriReplay-DOA dataset. Results are reported under four target–interferer angles (15°, 45°, 90°, 120°). Each entry in the form xx/xx/xx/xx corresponds to overlap ratios of 0–25%, 25–50%, 50–75%, and 75–100%, respectively.

Method	Training Strategy	Speaker angle				Overall
		15°	45°	90°	120°	
DSB + Gate	NT	37.1/ –/ –/ –	33.3/ –/ –/ –	28.3/ –/ –/ –	27.3/ –/ –/ –	(31.5)
FastMNMF	NT	31.3/ 39.4/ 40.9/ 34.7	14.3/ 16.5/ 15.0/ 12.9	10.4/ 22.2/ 20.1/ 19.4	12.7/ 14.7/ 15.8/ 14.1	21.1
Sortformer + GSS	NT	28.3/ 28.9/ 45.5/ 56.5	25.1/ 23.1/ 42.4/ 49.1	22.3/ 24.4/ 44.2/ 51.2	22.0/ 24.7/ 41.5/ 48.6	38.4
CSS (FasNet-TAC)	Scratch	46.0/ 49.9/ 60.9/ 61.6	42.6/ 39.7/ 56.2/ 57.0	34.9/ 40.4/ 56.1/ 57.3	30.2/ 34.2/ 48.6/ 48.9	49.5
CSS (TIGER)	Scratch	25.8/ 32.5/ 41.6/ 50.5	24.8/ 27.1/ 38.5/ 46.6	22.5/ 29.0/ 39.3/ 49.1	19.0/ 24.8/ 35.0/ 44.4	36.3
	PT+FT	19.9/ 25.9/ 39.8/ 49.1	19.8/ 22.2/ 36.1/ 45.1	18.6/ 24.4/ 36.3/ 45.6	15.7/ 20.1/ 30.9/ 40.9	32.8
PATSE	Scratch	15.6/ 19.6/ 24.7/ 28.0	12.9/ 22.5/ 22.7/ 22.5	10.9/ 20.0/ 15.4/ 19.7	10.1/ 12.6/ 14.4/ 20.2	18.9
	PT+FT	15.1/ 15.3/ 21.4/ 22.8	11.0/ 9.6/ 13.3/ 14.7	8.9/ 10.0/ 13.5/ 15.3	8.8/ 9.5/ 12.2/ 14.6	14.0

specifies how speaker identities are assigned (*SD* or *Oracle*). In contrast, PATSE produces target-conditioned outputs and does not require an additional speaker assignment step.

ASR and Segmentation. For ASR evaluation, estimated waveforms are segmented into utterances and transcribed using Whisper Large-v3 [35]. For long-form outputs (e.g., *DSB+Gate*, *FastMNMF*, and PATSE after stitching), Silero-VAD [36] is applied to obtain utterance boundaries, whereas *Sortformer+GSS* directly produces utterance-level outputs. For CSS-based systems, segmentation is derived from oracle speaker assignment, which defines the speaker boundaries.

Implementation Details. $\mathcal{T}(\cdot)$ is implemented as a two-layer Conv2D with PReLU, and $\mathcal{F}(\cdot)$ is a Conv2D with PReLU. All separation backbone configurations, including F , T , N , and K , follow TIGER-Large³. In the loss, the scaling factor α is set to 0.005. The model is trained with a learning rate of $5e-4$. ASR is evaluated using word error rate (WER), and speaker diarization using diarization error rate (DER).

4. Results

LibriReplay-DOA. Table 2 reports WER on LibriReplay-DOA across four target–interferer angles (15°, 45°, 90°, 120°) and four overlap ranges (0–25%, 25–50%, 50–75%, 75–100%). In the *Training Strategy* column, *NT* denotes no training; *Scratch* denotes training from scratch on the training data in Section 3.1; and *PT+FT* denotes initialization from a pretrained model (TIGER-Large) followed by finetuning. DOA serves as an explicit cue to the target speaker in *DSB+Gate* and PATSE, whereas CSS-based methods and *FastMNMF* are provided with oracle target speaker identities for comparison.

PATSE is applied per target DOA to separate all speakers. Since *DSB+Gate* cannot handle heavy overlap, it is evaluated only for overlap <25%. As expected, smaller target–interferer angles and higher overlap ratios increase task difficulty. Overall, PATSE consistently achieves the lowest WER across nearly all configurations, demonstrating the benefit of DOA-conditioned target extraction. The improvement of PATSE arises not only from DOA conditioning, as *DSB+Gate* also exploits DOA cues yet performs substantially worse, but also from the architectural advantages of the proposed framework. By extracting a single target stream per speaker, PATSE avoids inter-output crosstalk and cross-chunk permutation ambiguity in CSS systems. Even with oracle

Table 3: WER (%)↓ and DER (%)↓ on TEIDAN dataset.

Method	WER (%)	DER (%)
DSB + Gate	41.33	35.72
FastMNMF	26.82	28.42
Sortformer+GSS	45.03	36.15
CSS(TIGER)	37.43	–
PATSE	20.50	13.83

speaker assignment—an unrealistic upper bound that entirely eliminates diarization errors—*CSS(TIGER)* remains substantially worse, implying that the gap would be even larger under realistic conditions. The diarization-based pipeline *Sortformer+GSS* performs particularly poorly under high overlap conditions, as speaker temporal boundary estimation becomes unreliable, causing cascaded diarization–separation pipelines to struggle. Interestingly, *FastMNMF* shows competitive performance only under extremely high overlap ratios (75–100%), a scenario that is rarely encountered in real-world meetings.

TEIDAN Dataset. Table 3 reports ASR (measured by WER) and speaker diarization (measured by DER) results on the real-world conversational dataset TEIDAN. The same experimental setup is used, and the best-performing *CSS(TIGER)* setting is selected to represent CSS-based systems. Specifically, for *Sortformer+GSS*, the DER is computed directly from the diarization outputs of Sortformer. PATSE achieves the best overall performance in terms of both WER and DER, demonstrating improved robustness for speaker-attributed transcription in real-world three-party conversations. The presence of filler words and informal expressions can increase recognition difficulty; thus, the WER may be partly limited by the ASR backend.

5. Conclusion

In this paper, we presented PATSE, a position-aware target speaker extraction framework for addressing the “who spoke when and what” problem in multi-party conversations. By producing target-conditioned, speaker-attributed streams, PATSE derived speaker activity via simple post-processing without explicit diarization. To facilitate evaluation of DOA-based methods, we introduced and released LibriReplay-DOA, a real-room dataset with DOA annotations to address the lack of DOA labels in real-room recordings. Experiments on LibriReplay-DOA and the real-world conversational TEIDAN dataset demonstrated consistent improvements in downstream ASR compared to conventional pipelines, even under oracle speaker assignment.

³<https://huggingface.co/JusperLee/TIGER-speech>

6. Acknowledgments

This work was supported by JST BOOST JPMJBS2407 and JST Moonshot R&D JPMJMS2011.

7. Generative AI Use Disclosure

Generative AI tools (Gemini and ChatGPT) were used for language editing and improving the phrasing of this manuscript.

8. References

- [1] S. Cornell, J.-w. Jung, S. Watanabe, and S. Squartini, “One model to rule them all? towards end-to-end joint speaker diarization and speech recognition,” in *Proc. ICASSP*, 2024, pp. 11 856–11 860.
- [2] M. Shi, X. Xiao, R. Fan, S. Ling, and J. Li, “Train short, infer long: Speech-llm enables zero-shot streamable joint asr and diarization on long audio,” in *Proc. ICASSP*, 2026, pp. 17 442–17 446.
- [3] M. Shi, Z. Du, Q. Chen, F. Yu, Y. Li, S. Zhang, J. Zhang, and L.-R. Dai, “Casa-asr: Context-aware speaker-attributed asr,” in *Proc. Interspeech*, 2023, pp. 411–415.
- [4] S.-T. Niu, J. Du, R.-Y. Wang, G.-B. Yang, T. Gao, J. Pan, and Y. Hu, “DCF-DS: Deep cascade fusion of diarization and separation for speech recognition under realistic single-channel conditions,” *IEEE/ACM Trans. ASLP*, 2025.
- [5] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, “Continuous speech separation: Dataset and analysis,” in *Proc. ICASSP*, 2020, pp. 7284–7288.
- [6] D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, S. Watanabe, J. Du, T. Yoshioka, Y. Luo *et al.*, “Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis,” in *Proc. SLT*, 2021, pp. 897–904.
- [7] H. Taherian and D. Wang, “Multi-resolution location-based training for multi-channel continuous speech separation,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [8] T. Yoshioka, Z. Chen, C. Liu, X. Xiao, H. Erdogan, and D. Dimitriadis, “Low-latency speaker-independent continuous speech separation,” in *Proc. ICASSP*, 2019, pp. 6980–6984.
- [9] T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, and R. Haeb-Umbach, “Graph-PIT: Generalized permutation invariant training for continuous separation of arbitrary numbers of speakers,” in *Proc. Interspeech*, 2021, pp. 3490–3494.
- [10] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, and T. Nakatani, “Speaker activity driven neural speech extraction,” in *Proc. ICASSP*, 2021, pp. 6099–6103.
- [11] D. Raj, D. Povey, and S. Khudanpur, “Gpu-accelerated guided source separation for meeting transcription,” in *Proc. Interspeech*, 2023, pp. 3507–3511.
- [12] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny *et al.*, “The STC system for the chime-6 challenge,” in *ChIME 2020 Workshop on Speech Processing in Everyday Environments*, 2020.
- [13] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, “End-to-end neural speaker diarization with permutation-free objectives,” in *Proc. Interspeech*, 2019, pp. 4300–4304.
- [14] H. Taherian and D. Wang, “Multi-channel conversational speaker separation via neural diarization,” *IEEE/ACM Trans. ASLP*, vol. 32, pp. 2467–2476, 2024.
- [15] A. Briegleb, M. M. Halimeh, and W. Kellermann, “Exploiting spatial information with the informed complex-valued spatial autoencoder for target speaker extraction,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [16] M. Elminshawy, S. R. Chetupalli, and E. A. Habets, “Beamformer-guided target speaker extraction,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [17] M. Delcroix, S. Watanabe, T. Ochiai, K. Kinoshita, S. Karita, A. Ogawa, and T. Nakatani, “End-to-End speakerbeam for single channel target speech recognition,” in *Proc. Interspeech*, 2019, pp. 451–455.
- [18] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, “Voice-Filter: Targeted voice separation by speaker-conditioned spectrogram masking,” in *Proc. Interspeech*, 2019, pp. 2728–2732.
- [19] C. Boeddeker, A. S. Subramanian, G. Wichern, R. Haeb-Umbach, and J. Le Roux, “TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings,” *IEEE/ACM Trans. ASLP*, vol. 32, pp. 1185–1197, 2024.
- [20] S. Zhang, J. Zhang, Y. Wang, and H. Yan, “DOA or Speaker embedding: Which is better for multi-microphone target speaker extraction,” *IEEE Signal Processing Letters*, 2025.
- [21] Y. Wang, J. Zhang, S. Chen, W. Zhang, Z. Ye, X. Zhou, and L. Dai, “A study of multichannel spatiotemporal features and knowledge distillation on robust target speaker extraction,” in *Proc. ICASSP*, 2024, pp. 431–435.
- [22] Y. Wang, J. Zhang, C. Jiang, W. Zhang, Z. Ye, and L. Dai, “Leveraging boolean directivity embedding for binaural target speaker extraction,” in *Proc. ICASSP*, 2025, pp. 1–5.
- [23] M. Elmers, K. Inoue, D. Lala, and T. Kawahara, “Triadic multi-party voice activity projection for turn-taking in spoken dialogue systems,” in *Proc. Interspeech*, 2025, pp. 3015–3019.
- [24] M. Xu, K. Li, G. Chen, and X. Hu, “Tiger: Time-frequency interleaved gain extraction and reconstruction for efficient speech separation,” *arXiv preprint arXiv:2410.01469*, 2024.
- [25] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, “End-to-end microphone permutation and number invariant multi-channel speech separation,” in *Proc. ICASSP*, 2020, pp. 6394–6398.
- [26] Z.-Q. Wang and D. Wang, “Combining spectral and spatial features for deep learning based blind speaker separation,” *IEEE/ACM Trans. ASLP*, vol. 27, no. 2, pp. 457–468, 2018.
- [27] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Proc. AAAI*, vol. 32, no. 1, 2018.
- [28] Z. Pan, M. Ge, and H. Li, “USEV: Universal speaker extraction with visual cue,” *IEEE/ACM Trans. ASLP*, vol. 30, pp. 3032–3045, 2022.
- [29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *Proc. ICASSP*, 2015, pp. 5206–5210.
- [30] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, “gpuRIR: A python library for room impulse response simulation with GPU acceleration,” *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 5653–5671, 2021.
- [31] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrarni, R. Cheng, H. Dubey, S. Matushevych, R. Aichner, A. Aazami, S. Braun *et al.*, “The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results,” *arXiv preprint arXiv:2005.13981*, 2020.
- [32] K. Sekiguchi, Y. Bando, A. A. Nugraha, K. Yoshii, and T. Kawahara, “Fast multichannel nonnegative matrix factorization with directivity-aware jointly-diagonalizable spatial covariance matrices for blind source separation,” *IEEE/ACM Trans. ASLP*, vol. 28, pp. 2610–2625, 2020.
- [33] T. Park, I. Medennikov, K. Dhawan, W. Wang, H. Huang, N. R. Koluguri, K. C. Puvvada, J. Balam, and B. Ginsburg, “Sortformer: A novel approach for permutation-resolved speaker supervision in speech-to-text systems,” *arXiv preprint arXiv:2409.06656*, 2024.
- [34] D. Raj, D. Povey, and S. Khudanpur, “GPU-accelerated guided source separation for meeting transcription,” *arXiv preprint arXiv:2212.05271*, 2022.
- [35] A. Radford *et al.*, “Robust speech recognition via large-scale weak supervision,” 2022.
- [36] S. Team, “Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier,” <https://github.com/snakers4/silero-vad>, 2024.