G-STAR: End-to-End Global Speaker-Tracking Attributed Recognition

Jing Peng[2,4,*] **Ziyi Chen**[1,3,*] **Haoyu Li**[1,2,*] **Yucheng Wang**[5] **Duo Ma**[4]
Mengtian Li[3] Yunfan Du[3] Dezhu Xu[3] Yu Xi[2] Kai Yu[2] Shuai Wang[1,4,†]
[1] Nanjing University [2] Shanghai Jiao Tong University
[3] Central Media Technology Institute, Huawei
[4] Shenzhen Research Institute of Big Data [5] ETH Zürich
jing.peng@sjtu.edu.cn, zc612997@gmail.com, haoyu.li.cs@sjtu.edu.cn
[*] Equal contribution. [†] Corresponding author.

Abstract

We study timestamped speaker-attributed automatic speech recognition (SA-ASR) for long-form, multi-party speech with overlap. In this setting, chunk-wise inference must preserve meeting-level speaker identity consistency while producing time-stamped, speaker-labeled transcripts. Prior Speech-LLM systems tend to prioritize either local diarization or global labeling, lacking the ability to jointly model fine-grained temporal boundaries and robust cross-chunk identity linking. We propose **G-STAR**, an end-to-end framework that couples a cache-conditioned speaker-tracking module with a Speech-LLM transcription backbone. The tracker provides structured speaker cues with temporal grounding, and the LLM generates attributed text conditioned on these cues. G-STAR supports component-wise optimization and joint end-to-end training, enabling flexible learning under heterogeneous supervision and domain shift. Under chunk-wise decoding protocols, experiments on both oracle-segmented local evaluation and full-meeting global evaluation show strong speaker-attributed transcription performance.

1 Introduction

Large language models (LLMs) have recently become central to speech understanding systems, largely owing to their strong semantic reasoning, discourse-level context modeling, and instruction-following capabilities (Tang et al., 2023; Chu et al., 2023; Xu et al., 2025a; Goel et al., 2025). A prevailing paradigm, often referred to as *Speech-LLM*, combines a pre-trained speech encoder with an LLM through a lightweight projector that maps acoustic representations into the LLM embedding space (Peng et al., 2025; Cui et al., 2025; Xu et al., 2025b). This encoder–projector–LLM recipe has shown promising results across a broad range of speech tasks, especially those requiring higher-level linguistic and pragmatic understanding.

However, real-world conversational audio commonly departs from the single-speaker assumption. In cocktail-party conditions and multi-party meetings, overlapping speech and rapid turn-taking demand not only accurate transcription (Çetin and Shriberg, 2006; Yoshioka et al., 2018; Shriberg et al., 2001), but also reliable attribution of *who spoke what* (Kanda et al., 2022), and increasingly *when* (El Shafey et al., 2019). This motivates *speaker-attributed automatic speech recognition* (SA-ASR), which aims to produce a structured transcript containing both lexical content and speaker attribution (Lu et al., 2021; Raj et al., 2023). In this work, we consider a timestamped SA-ASR formulation: given an input waveform x, the system outputs a sequence of attributed segments

$$\mathcal{Y} = \{(s_n, \tau_n^{\mathrm{st}}, \tau_n^{\mathrm{ed}}, \mathbf{y}_n)\}_{n=1}^{N}, \tag{1}$$

where s_n denotes the speaker identity, $(\tau_n^{\mathrm{st}}, \tau_n^{\mathrm{ed}})$ denotes temporal boundaries, and $\mathbf{y}_n$ is the word sequence. Crucially, for long-form recordings processed in chunks, SA-ASR requires *meeting-level* (or *global*) speaker identity consistency: the same real-world speaker should be assigned the same identity across the entire recording (Raj et al., 2024), rather than being re-indexed independently in each chunk.

The central challenge is therefore not simply to recognize multiple speakers within a short segment. A practical meeting system must keep speaker identities stable across chunks, retain timestamp-level grounding, and avoid relying on post-hoc global clustering that is difficult to integrate with LLM-style generation. This motivates a tracking-conditioned generation view: speaker tracking should provide structured, reusable cues to the transcription model rather than remain a separate downstream diarization step.

Recent efforts begin to address parts of this problem, yet leave key gaps. SpeakerLM (Yin et al.,

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 36267–36276
October 24-29, 2026 ©2026 Association for Computational Linguistics

2026) augments the Speech LLM pipeline with an additional speaker embedding extractor, enabling robust diarization within a local chunk; however, it does not provide an explicit mechanism to *globally associate* speaker identities across multiple chunks when long recordings must be processed segment-by-segment. JEDIS-LLM (Shi et al., 2025) introduces a *Speaker Prompt Cache* to store and index speaker representations, thereby enabling the model to emit *globally consistent* speaker labels together with the corresponding transcripts under chunk-wise inference; nevertheless, it lacks explicit perception and emission of fine-grained temporal boundaries (timestamps). TagSpeech (Huo et al., 2026) further strengthens temporal grounding by incorporating a dedicated speaker encoder and temporal anchors to obtain speaker and timestamp cues, but it similarly does not resolve the *meeting-level* global speaker identity linking problem under chunk-wise inference. Taken together, these advances highlight the need for an end-to-end framework that simultaneously achieves (i) timestamped, overlap-aware attribution, and (ii) persistent, meeting-level global speaker identity consistency.

To bridge these gaps, we propose **G-STAR: End-to-End Global Speaker-Tracking Attributed Recognition**, an LLM-based end-to-end SA-ASR framework designed for long-form, multi-speaker speech with overlapping regions. G-STAR integrates an interpretable *speaker-tracking module* based on Sortformer (Park et al., 2024; Medennikov et al., 2025)-style modeling with a Speech-LLM transcription backbone. The speaker-tracking module provides structured, time-aware speaker cues, while the LLM generates attributed text conditioned on these cues. Unlike cascaded SA-ASR pipelines that separately perform ASR, diarization, timestamp alignment, and post-hoc fusion, G-STAR injects projected, slot-aligned speaker cues into LLM decoding so lexical tokens, timestamps, and global speaker labels are produced under a shared conditioning context. This modular design enables both *component-wise* optimization (improving tracking and transcription separately) and *joint* end-to-end optimization (integrating speaker and lexical modeling in a unified objective), thereby supporting flexible training strategies under dataset imbalance and domain shifts. Most importantly, G-STAR is built to preserve *global* speaker identity consistency across chunk-wise long-form inference, enabling meeting-level speaker labeling without post-hoc global clustering. Accordingly, our contributions are threefold:

- We present G-STAR, an end-to-end, LLM-based SA-ASR system for multi-party meetings that produces *timestamped* speaker-attributed transcripts with *meeting-level global speaker identity consistency* under chunk-wise long-form inference. We will release the model and code.
- We evaluate G-STAR in both local and global SA-ASR settings on challenging meeting benchmarks, comparing against representative Speech-LLM systems and strong conventional pipelines under clearly specified VAD and collar protocols.
- We conduct analyses and ablations to identify the key design factors behind the gains, including speaker cue fusion strategies, fusion ratio choices for injecting speaker cues into LLM generation, and the impact of hierarchical cross-entropy objectives.

2 Related Work

2.1 Sortformer and AOSC for Speaker Tracking

Sortformer (Park et al., 2024) proposes a *tracking-centric* formulation that connects frame-level speaker activity with token-level speaker attribution. Using notation aligned with Section 3, an audio encoder maps each chunk $x^{(t)}$ to acoustic representations $\mathbf{H}^{(t)} = f_{\text{enc}}(x^{(t)})$, and a Sortformer-style tracker estimates speaker activity over arrival-order slots. Slot k denotes the k-th speaker by first appearance, which makes speaker indices interpretable and reduces the label-permutation ambiguity that arises when multi-speaker outputs are estimated independently. This arrival-order view is particularly useful for chunk-wise processing, where speakers must not be re-indexed independently in each chunk.

To carry this ordering across chunks, Streaming Sortformer (Medennikov et al., 2025) introduces the *Arrival-Order Speaker Cache (AOSC)*. For chunk t, the tracker conditions on the previous cache state and produces frame-synchronous speaker cues,

$$\mathbf{S}^{(t)} = f_{\text{trk}}(x^{(t)}, \mathcal{C}^{(t-1)}), \tag{2}$$

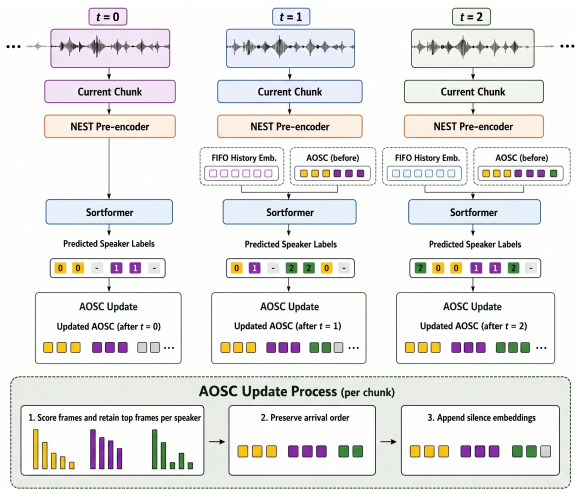

Figure 1: Illustration of Streaming Sortformer with AOSC. The cache stores arrival-order speaker states across chunks and reuses them to keep speaker identities consistent when speakers reappear.

after which the cache is updated as

$$\mathcal{C}^{(t)} = \mathrm{UpdateCache}(\mathcal{C}^{(t-1)}, \mathbf{S}^{(t)}). \quad (3)$$

Conceptually, AOSC stores compact evidence for speakers in first-arrival order: new speakers receive new slots, while returning or temporarily silent speakers can be linked back to existing slots. This mechanism (Fig. 1) provides the cross-chunk identity state needed by G-STAR, where speaker tracking serves as structured conditioning for Speech-LLM generation.

2.2 Serialized Output Training

Serialized Output Training (SOT) (Kanda et al., 2020) addresses the *label permutation* issue in overlapped and multi-speaker transcription by converting multi-stream supervision signals into a single serialized target sequence. Instead of predicting K parallel speaker-dependent streams, SOT defines a deterministic serialization function that maps the set of speaker-labeled utterances (or token sequences) into a single sequence by inserting explicit *speaker tags* (e.g., <spk1>, <spk2>) between lexical tokens (Kanda et al., 2020). This enables standard autoregressive or sequence-to-sequence training while avoiding explicit permutation search at inference time.

Crucially, SOT provides an *output protocol* for speaker-attributed text that is compatible with LLM-style generation: the model emits a single token stream that interleaves speaker identifiers and words, and can be naturally extended with additional structural tokens (e.g., timestamps) when temporal grounding is required. However, under chunk-wise inference, SOT alone does not guarantee *meeting-level* identity consistency: speaker tags can still permute across chunks unless the model maintains a persistent speaker concept (e.g., via an explicit tracking state). Therefore, in our framework, SOT is adopted as the structured output interface, while global identity persistence is provided by cache-conditioned Sortformer-style tracker outputs used as speaker cues.

Overall, these lines of work suggest complementary strengths: Sortformer-style models provide stable tracking cues, whereas SOT-style generation provides a flexible text interface for attributed transcripts. G-STAR (Fig. 2) combines these two perspectives by treating speaker tracking as structured conditioning for generation, which is the bridge from the background reviewed here to the method introduced next.

3 Method

3.1 Problem Setup and Notation

Let a long-form recording be a waveform x partitioned into T consecutive chunks $\{x^{(t)}\}_{t=1}^{T}$. In practice, chunk durations are *approximately stable but not fixed* due to chunking constraints, utterance boundaries, or buffering policies; thus the number of encoder frames L_t and tracker frames M_t may vary with t.

Our goal is *timestamped speaker-attributed ASR with meeting-level consistent speaker identities*. For each chunk t, the model emits a serialized sequence in an SOT-style format:

$$\mathbf{z}^{(t)} = \big[\texttt{<t_st>},\ \mathbf{w},\ \texttt{<t_ed>},\ \texttt{<spk=k>}\big]^{*}, \quad (4)$$

where $\mathbf{w}$ denotes lexical tokens, and <spk=k> denotes a *global* speaker-ID token that is consistent across all chunks within the same meeting.

G-STAR couples three components in a chunk-wise fashion: (i) an ASR acoustic branch that produces LLM-conditioned acoustic embeddings, (ii) an SD/speaker-tracking branch that produces time-synchronous speaker cues, and (iii) a persistent *Arrival-Order Speaker Cache* (AOSC) that maintains meeting-level speaker identities. At chunk t, the SD branch consumes the previous cache state $\mathcal{C}^{(t-1)}$, produces speaker cues, and updates the cache to $\mathcal{C}^{(t)}$. The LLM then performs SOT decoding conditioned on the fused acoustic/speaker embeddings.

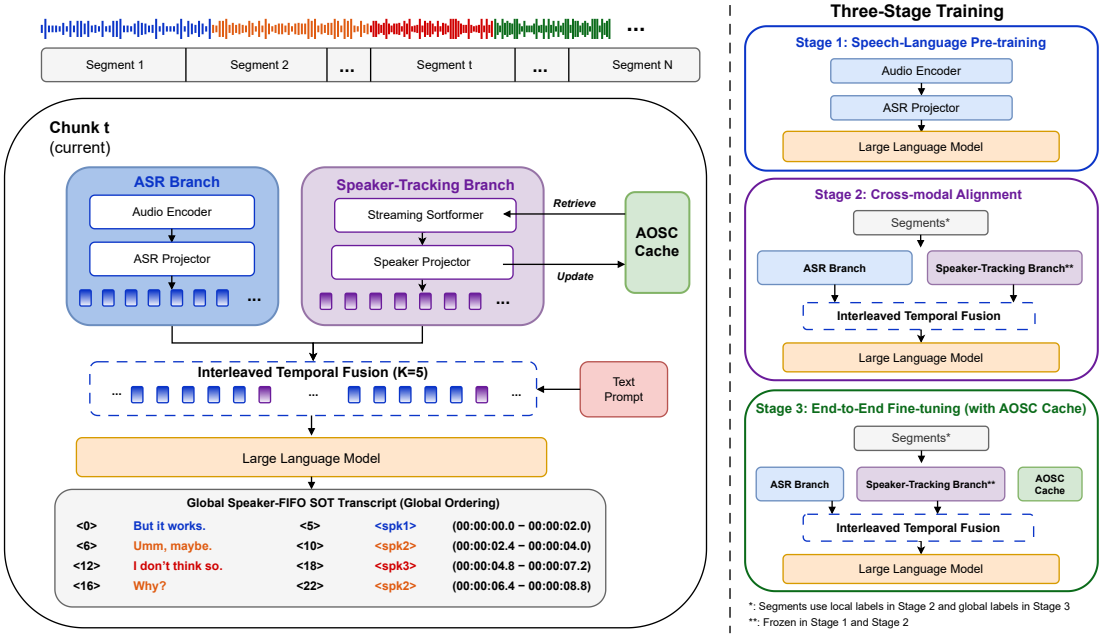

Figure 2: Overview of G-STAR Structure. The left panel illustrates the proposed G-STAR model architecture, which features a dual-branch encoder design comprising ASR and speaker diarization modules to support chunk-wise inference. The right panel outlines the three-stage training strategy for G-STAR, progressing sequentially from Speech-Language Pre-training to local SA-ASR (Cross-modal Alignment) and finally to global SA-ASR (End-to-End Fine-tuning).

3.2 Chunk-Wise Dual-Branch Modeling with AOSC

3.2.1 ASR Acoustic Branch

Given chunk $x^{(t)}$, an audio encoder $f_{\text{enc}}(\cdot)$ (e.g., Conformer/Whisper-style) produces frame-level representations:

$$\mathbf{H}^{(t)} = f_{\text{enc}}(x^{(t)}) \in \mathbb{R}^{L_t \times d_h}, \tag{5}$$

where L_t varies with chunk duration. A projector $g_1(\cdot)$ maps them into the LLM embedding space:

$$\mathbf{U}^{(t)} = g_1(\mathbf{H}^{(t)}) \in \mathbb{R}^{L_t \times d_{\text{llm}}}. \tag{6}$$

3.2.2 SD Branch and Cache-Conditioned Speaker Cues

The SD/speaker-tracking module maintains a persistent cache $\mathcal{C}^{(t)}$ across chunks. For chunk t, the tracker takes the current audio and the previous cache state as input, and outputs frame-synchronous cache-conditioned speaker cues:

$$\begin{aligned} \mathbf{S}^{(t)}, \mathcal{C}^{(t)} &= f_{\text{trk}}(x^{(t)}, \mathcal{C}^{(t-1)}), \\ \mathbf{S}^{(t)} &\in \mathbb{R}^{M_t \times d_s}, \end{aligned} \tag{7}$$

where M_t is the number of tracker frames, also varying with chunk duration and the tracker frame rate. A projector $g_2(\cdot)$ maps the speaker cues into the LLM embedding space:

$$\mathbf{V}^{(t)} = g_2(\mathbf{S}^{(t)}) \in \mathbb{R}^{M_t \times d_{\text{llm}}}. \tag{8}$$

AOSC stores compact speaker evidence ordered by *first-arrival time*, thereby defining a stable meeting-level indexing rule. When a new speaker appears, AOSC assigns the next arrival-order index; when a previously seen speaker reappears, AOSC retrieves the corresponding index. This renders speaker identity *stateful* across chunks and eliminates cross-chunk permutation drift. The detailed cache retrieval, allocation, and update procedure of AOSC is illustrated in Fig. 1, where returning speakers are linked to existing arrival-order slots and newly detected speakers are assigned to the next available slot.

3.3 Interleaved Temporal Fusion and Global SOT Decoding

3.3.1 Interleaved Temporal Fusion

We fuse acoustic embeddings $\mathbf{U}^{(t)}$ and speaker embeddings $\mathbf{V}^{(t)}$ *along time* by inserting sparse speaker tokens into the acoustic token stream. Let K be the insertion stride. We construct a fused embedding sequence:

$$\mathbf{E}^{(t)} = \text{Interleave}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)}; K) \in \mathbb{R}^{N_t \times d_{\text{llm}}}, \tag{9}$$

Algorithm 1 G-STAR chunk-wise inference with cache-conditioned tracking and global SOT decoding

Require: Meeting chunks $\{x^{(t)}\}_{t=1}^{T}$; insertion stride K; optional prompt $\mathbf{p}$.
Ensure: Timestamped transcript with meeting-level speaker IDs.
Initialize AOSC cache $\mathcal{C}^{(0)} \leftarrow \varnothing$ and transcript list $\mathcal{Z}$.
for $t = 1$ to T **do**
  Encode acoustic frames: $\mathbf{H}^{(t)} \leftarrow f_{\text{enc}}(x^{(t)})$.
  Project ASR features: $\mathbf{U}^{(t)} \leftarrow g_1(\mathbf{H}^{(t)})$.
  Track speakers with history: $(\mathbf{S}^{(t)}, \mathcal{C}^{(t)}) \leftarrow f_{\text{trk}}(x^{(t)}, \mathcal{C}^{(t-1)})$.
  Maintain arrival-order speaker indices for <spk=k>.
  Project speaker cues: $\mathbf{V}^{(t)} \leftarrow g_2(\mathbf{S}^{(t)})$.
  Resample $\mathbf{V}^{(t)}$ to the ASR insertion grid.
  Build $\mathbf{E}^{(t)} \leftarrow \text{Interleave}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)}; K)$ in temporal order.
  Decode $\mathbf{z}^{(t)} \leftarrow \text{LLMDecode}(\mathbf{p}, \mathbf{E}^{(t)})$.
  Append $\mathbf{z}^{(t)}$ to $\mathcal{Z}$; retain $\mathcal{C}^{(t)}$.
end for
return $\text{Concat}(\mathcal{Z})$.

where $N_t \approx L_t + \lceil L_t/K \rceil$. Operationally, for $i = 1, \ldots, L_t$, we append $\mathbf{U}_i^{(t)}$ and additionally insert a speaker embedding whenever $i \equiv 0 \pmod{K}$. When M_t does not match the insertion schedule, we apply a deterministic resampling (nearest-neighbor or linear) to align speaker frames to insertion locations. This yields a single time-ordered embedding stream that periodically carries explicit speaker evidence.

3.3.2 Cache-Consistent SOT Decoding with Global Speaker IDs

We feed $\mathbf{E}^{(t)}$ to an autoregressive LLM decoder. Optionally, a text prompt $\mathbf{p}$ is prepended. The decoder generates the SOT sequence conditioned on the fused embeddings:

$$p(\mathbf{z}^{(t)} \mid x^{(t)}, \mathcal{C}^{(t-1)}) = \prod_{m=1}^{|\mathbf{z}^{(t)}|} p(z_m^{(t)} \mid \mathbf{p}, \mathbf{E}^{(t)}, z_{<m}^{(t)}). \quad (10)$$

The speaker token <spk=k> in Eq. (4) corresponds to the k-th arrival-order index maintained by AOSC. The tracker update enables end-to-end meeting-level identity consistency during chunk-wise inference (Alg. 1).

4 Experimental Setups

We design the experiments to answer two complementary questions. The local setting isolates speaker-attributed recognition under controlled utterance-level segmentation, whereas the global setting evaluates full-meeting behavior where segmentation, speaker tracking, and attribution interact over long contexts. This separation is important because local and global results reflect different sources of difficulty. We implemented our model and conducted experiments using the WEST (Zhang et al., 2025) framework.

4.1 Data

We trained and evaluated our models on four conversational and meeting-style speech datasets: MLC (Mu et al., 2025), AMI (Carletta et al., 2005), Fisher (Cieri et al., 2004), and Candor (Reece et al., 2023). For MLC, only the English subset was used; for Candor, only the audio modality was adopted, excluding video information. All training audio was segmented into clips of up to 20 seconds. For evaluation, two settings were considered: in the local setting, the test set was processed in the same way as the training data; in the global setting, the test set comprised full-length audio, with the model performing chunk-by-chunk inference.

In addition to the public benchmarks, we also include internal data for model training and evaluation. For training, we added an internal Chinese conversational corpus, which is dominated by two-speaker conversations.The only training-data difference between the internal model and the open-source model is whether internal Chinese conversational corpus and the open-source AISHELL-4 (Fu et al., 2021) and AliMeeting (Yu et al., 2022) datasets are used. The internal test set is collected from real out-of-domain meeting audio and contains two-speaker, three-speaker, and four-speaker recordings of 0.50, 0.49, and 1.86 hours, respectively.

4.2 Modeling

We have detailed the proposed architecture in Section 3.2. The audio encoder and projector are initialized from FireRed-LLM (Xu et al., 2025b). The LLM uses Qwen2-7B-Instruct (Yang et al., 2025) with LoRA weights inherited from FireRed-LLM. For the SD branch, the speaker diarization module is initialized with Streaming Sortformer[1], whereas its projector consists of a 1D convolutional downsampling module with a stride of 5, followed by a two-layer MLP, and is randomly initialized.

4.3 Training Procedure

The overall optimization followed three consecutive stages.

[1] nvidia/diar_streaming_sortformer_4spk-v2

- First, we performed meeting-style ASR pre-training to adapt the speech encoder–projector–LLM pipeline to conversational and meeting-style audio.
- Second, we conducted local SA-ASR training on segmented utterances, where the model learned to generate timestamped transcriptions with speaker-attribution tokens.
- Third, we performed global SA-ASR training to adapt the system to meeting-level inference, where speaker identities must remain consistent across chunks rather than being assigned independently within each segment.

During this global stage, we also tuned the Sortformer module on 90-second chunks from the same four datasets to improve long-form speaker tracking under realistic meeting-level inference. This component can be tuned modularly because the diarization module is coupled to the downstream Speech-LLM generator through the SD projector.

4.4 Training Hyperparameters

We used packed training examples with a maximum pack size of 12,000 tokens in all Speech-LLM training stages. The trainable modules were kept consistent across these stages: the ASR projector, the SD projector, and the LoRA adapters. The LoRA rank, scaling factor, and dropout were fixed to 64, 16, and 0.05, respectively. We optimized the Speech-LLM components with AdamW using a weight decay of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping. For meeting-style ASR pre-training and local speaker-attributed ASR (SA-ASR) training, the learning rate was linearly warmed up with a warmup ratio of 0.01 to a peak value of 5×10^{-5} and then decayed with cosine annealing; each of the two stages was trained for 20,000 steps. For global (meeting-level) SA-ASR training, we kept the same pack size and learning-rate schedule, but reduced the peak learning rate to 2×10^{-5} and trained for 5,000 steps. Timestamps and speaker labels were added as special LLM tokens. We optimized the generation objective with hierarchical cross-entropy, assigning a $1.5\times$ loss weight to timestamp tokens and a $2\times$ loss weight to speaker-label tokens. For Sortformer tuning in the global stage, we used AdamW with a learning rate of 1×10^{-4} and a batch size of 4, and trained for 5 epochs on 90-second segments constructed from AMI, Fisher, MLC, and Candor.

4.5 Evaluation Setup

Unless otherwise noted, all reported inference uses chunk-wise decoding. In the local setting, all systems are evaluated on utterance-level inputs of up to 20 seconds with oracle VAD/segmentation, which isolates the back-end speaker-attributed recognition capability from segmentation errors. In the global setting, systems are evaluated on full-meeting recordings with each model's own VAD or segmentation front-end, which better reflects realistic long-form usage but also means that VAD behavior is part of the evaluated system. Unless otherwise specified, the collar for DER is set to 0. When hallucinated hypothesis segments appear, we score them consistently for all systems: the hallucinated outputs are treated as errors and are reflected in cpWER and DER.

To assess whether G-STAR offers benefits beyond post-hoc integration, we implement a cascaded pipeline as the closest non-end-to-end counterpart. For this baseline, we used the same Speech-LLM back-end, FireRedASR-LLM, for the ASR component. A VAD model[2] was employed to remove silence segments and generate input utterances for ASR while preserving global timestamps. As FireRedASR-LLM does not provide word-level timestamps, we applied a pre-trained CTC-based forced alignment model[3] to align the ASR outputs and obtain word-level timestamps. For speaker diarization, we adopted the same Sortformer-style front-end as **G-STAR** to handle long-form speech and produce global speaker timestamps. Finally, the global word-level timestamps from ASR were integrated with the speaker timestamps to generate the final speaker-attributed transcription with global timing information.

For the internal test set, we also compare with a cluster-based internal baseline, which performs segment-level recognition followed by global speaker clustering to produce speaker-attributed transcripts.

5 Evaluation

Throughout this paper, we use *meeting-level* and *global* to refer to speaker identity consistency over the full recording.

[2] https://github.com/snakers4/silero-vad
[3] MahmoudAshraf/mms-300m-1130-forced-aligner

Table 1: Performance comparison in the local setting (audio up to 20s) with oracle VAD/segmentation, reported in cpWER/DER (%). The NVIDIA default cascade is shown as a two-row group: Sortformer for SD/DER and Parakeet for ASR/cpWER. DER is computed with collar 0. **Bold** indicates the best result.

Group	Component	AMI	Fisher	MLC	Candor
		cpWER/DER↓	cpWER/DER↓	cpWER/DER↓	cpWER/DER↓
NVIDIA cascade	Sortformer (SD) (Park et al., 2024)	– / 29.87	– / 18.33	– / 17.76	– / 30.92
	Parakeet (ASR) (Wang et al., 2025)	**24.62** / –	27.73 / –	25.90 / –	27.44 / –
E2E SA-ASR	Vibevoice-ASR (Peng et al., 2026)	30.51 / 31.99	15.18 / 17.68	21.74 / 14.01	22.12 / 30.89
	MOSS-Diarizen (MOSI.AI et al., 2026)	25.13 / 32.20	11.69 / 21.61	14.16 / 10.58	16.38 / 31.76
G-STAR	Ours	24.86 / **19.00**	**10.29 / 8.18**	**13.90 / 6.49**	**14.54 / 17.56**

5.1 Chunk-Level Evaluation

The chunk-level evaluation in Table 1 uses utterance-level inputs of up to 20 seconds with oracle VAD/segmentation, providing a controlled test of speaker-attributed recognition without front-end segmentation variability. This setting isolates the backend capability of each system, namely whether it can jointly model lexical content, speaker attribution, and local temporal structure when reliable input boundaries are available.

The Sortformer and Parakeet rows should be read together as the strong cascade: Sortformer reports the SD/DER component, while Parakeet reports the ASR/cpWER component. Under this local protocol, G-STAR consistently reduces DER relative to Sortformer, showing stronger speaker-attribution behavior within short chunks. At the same time, it remains comparable to Parakeet on AMI and achieves lower cpWER on Fisher, MLC, and Candor, suggesting that the injected speaker cues do not compromise lexical recognition. Compared with strongend-to-end Speech-LLM baselines such as VibeVoice-ASR (Peng et al., 2026) and MOSS-Diarizen (MOSI.AI et al., 2026), G-STAR improves both metrics on all datasets. These results indicate that explicit cache-conditioned speaker cue fusion strengthens the LLM's ability to generate speaker-attributed transcripts, rather than relying only on the transcription backbone or on post-hoc diarization-style attribution.

5.2 Meeting-Level Evaluation

As shown in Table 2, G-STAR achieves the best meeting-level cpWER on Fisher, MLC, Candor, and AMI. The NVIDIA cascade rows report its SD and ASR components separately, while our controlled cascade uses the same main tracker/backbone design as G-STAR but performs late fusion between ASR and speaker timestamps. Compared with this controlled cascade, G-STAR obtains consistently lower cpWER, showing that the gain comes from tracking-conditioned generation rather than only from stronger components. The DER results reveal the expected trade-off: dedicated diarization systems can still be stronger for diarization purity, whereas G-STAR prioritizes speaker-attributed transcription with cache-consistent global labels.

We also conduct an extended evaluation on an internal out-of-domain test set to verify that the observed trend generalizes to practical meeting scenarios, including two-speaker and three-to-four-speaker configurations. Table 3 shows that our system, after further fine-tuning on a small amount of internal conversational data, consistently outperforms both the internal baseline and VibeVoice-ASR in terms of DER and cpWER.

5.3 Ablation Study

After establishing the main local and global results, we analyze why the proposed architecture helps. The ablation focuses on two modeling choices that directly control how speaker information enters the LLM: interleaved temporal fusion and hierarchical cross-entropy weighting.

Table 4 shows that the two components are complementary rather than redundant. Interleave fusion reduces both cpWER and DER, with a larger gain on cpWER, suggesting that periodically injecting speaker cues helps the LLM generate the correct lexical sequence together with structure-critical tokens such as speaker and timestamp markers. In contrast, hierarchical CE yields a clearer improvement on DER while leaving cpWER largely unchanged, indicating that reweighting timestamp and speaker-label tokens mainly sharpens temporal boundaries and speaker turn segmentation. The full model combines these effects, improv-

Table 2: Global (meeting-level) performance comparison, reported in cpWER/DER (%). The NVIDIA default cascade is shown as separate SD and ASR components; the controlled cascade uses the same main components as G-STAR but applies late fusion. DER is computed with collar 0. **Bold** indicates the best result.

Group	Component	Fisher	MLC	Candor	AMI
		cpWER/DER↓	cpWER/DER↓	cpWER/DER↓	cpWER/DER↓
NVIDIA cascade	Sortformer (SD) (Park et al., 2024)	– / **15.21**	– / 21.92	– / 18.03	– / **28.35**
	Parakeet (ASR) (Wang et al., 2025)	24.41 / –	31.03 / –	26.92 / –	35.70 / –
Speech-LLM	Vibevoice-ASR (Peng et al., 2026)	25.03 / 27.15	25.41 / 19.83	27.24 / 25.68	34.19 / 39.95
Controlled	Late-fusion cascade	21.01 / 23.41	23.18 / 21.38	17.62 / **17.67**	39.52 / 37.63
G-STAR	Ours	**16.44** / 16.85	**17.15** / **14.25**	**15.17** / 24.89	**30.85** / 32.23

Table 3: Global SA-ASR performance on the internal test set, reported in cpWER/DER (%). DER is computed with collar 0.5.

System	2 speakers	3–4 speakers	Avg.
	cpWER/DER↓	cpWER/DER↓	cpWER/DER↓
Vibevoice-ASR (Peng et al., 2026)	11.10 / 14.83	54.48 / 38.33	47.64 / 34.62
Cluster-based Pipeline	23.56 / 14.20	41.14 / 30.76	38.37 / 28.15
G-STAR	**10.42 / 4.86**	**38.85 / 28.59**	**34.37 / 24.88**

ing token-level attribution through interleave fusion and diarization-sensitive structure prediction through hierarchical CE.

We further analyze the effect of the interleave ratio, which controls the relative frequency with which speaker cues are injected into the LLM token stream during fusion. As shown in Figure 3, an overly dense injection can interfere with lexical modeling by increasing the proportion of non-lexical conditioning tokens, while an overly sparse injection may provide insufficient speaker-tracking guidance. The results suggest that a moderate interleave ratio achieves the best balance between speaker attribution and transcription accuracy, confirming that the benefit of interleave fusion depends not only on whether speaker cues are used, but also on how frequently they are integrated.

6 Conclusion

We presented G-STAR, an end-to-end framework for timestamped speaker-attributed ASR under chunk-wise long-form inference. G-STAR uses cache-conditioned Sortformer-style tracking to provide slot-aligned speaker cues to the LLM, enabling

Table 4: Ablation study of model configuration under local setting.

Hierarchical CE loss	Interleave Temporal Fusion	AMI	Fisher	Candor
		cpWER/DER↓	cpWER/DER↓	cpWER/DER↓
×	✓	26.33 / 21.06	10.88 / 10.24	14.97 / 20.21
✓	×	28.63 / 21.28	14.23 / 9.02	18.30 / 18.10
✓	✓	**24.86 / 19.00**	**10.29 / 8.18**	**14.54 / 17.56**

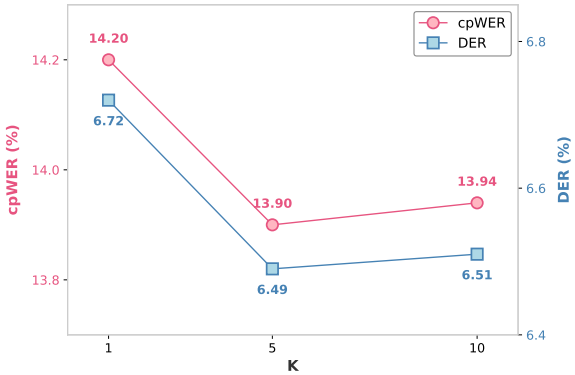

Figure 3: Effect of the interleave ratio K on MLC SA-ASR performance in the local setting. Smaller K injects speaker cues more frequently, while larger K provides sparser speaker conditioning.

lexical tokens, timestamps, and global speaker IDs to be generated under a shared conditioning context. Experiments across local and meeting-level settings show improved speaker-attributed transcription, and ablations confirm the roles of interleaved cue fusion and hierarchical CE. These results suggest that tracking-conditioned generation is a practical alternative to late-fusion cascades.

Beyond the empirical gains, the main implication of G-STAR is that speaker attribution can be treated as a conditioning signal during generation rather than as a post-processing decision made after ASR. The cache preserves a compact notion of who has appeared, the interleaved fusion mechanism exposes this information to the language model at temporally meaningful points, and the global SOT interface converts the resulting context into a single readable transcript. This combination is especially useful for long-form conversations. Overall, the framework offers a unified path for jointly improving transcription, timestamping, and speaker consistency while remaining compatible with modular training and chunk-wise deployment.

Limitations

Although G-STAR is designed around a cache-conditioned speaker tracker and is compatible with chunk-wise inference, this work does not fully evaluate a strictly streaming deployment. Future work will extend the system to real-time streaming settings and analyze latency, memory usage, and cache update stability under online constraints. Another limitation is data scale: the current experiments cover several public and internal meeting-style datasets, but larger and more diverse training data may further improve robustness across acoustic conditions, speaking styles, languages, and speaker configurations. Scaling both the Speech-LLM training data and the speaker-tracking supervision is therefore an important direction for improving generalization.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62401377) and the Yangtze River Delta Science and Technology Innovation Community Joint Research Project (Grant No. 2024CSJGG1100). This work was also supported by the Joint Fund Project of Shanghai Commercial Aircraft Systems Engineering Science and Innovation Center (No. B2025JS36236), and the Key Research and Development Program of Jiangsu Province, China (No. BE2022059 and No. BF2025029).

References

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, and 1 others. 2005. The AMI meeting corpus: A pre-announcement. In *Proc. International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 28–39.

Özgür Çetin and Elizabeth Shriberg. 2006. Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition. In *Proc. Interspeech*. Paper 1915-Mon2A2O.6.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The Fisher corpus: A resource for the next generations of speech-to-text. In *Proc. International Conference on Language Resources and Evaluation (LREC)*, pages 69–71.

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Steven Y. Guo, and Irwin King. 2025. Recent advances in speech language models: A survey. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 13943–13970.

Laurent El Shafey, Hagen Soltau, and Izhak Shafran. 2019. Joint speech recognition and speaker diarization via sequence transduction. In *Proc. Interspeech*, pages 396–400.

Yihui Fu, Luyao Cheng, Shubo Lv, Yukai Jv, Yuxiang Kong, Zhuo Chen, Yanxin Hu, Lei Xie, Jian Wu, Hui Bu, Xin Xu, Jun Du, and Jingdong Chen. 2021. AISHELL-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario. In *Proc. Interspeech*, pages 3665–3669.

Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and 1 others. 2025. Audio Flamingo 3: Advancing audio intelligence with fully open large audio language models. *arXiv preprint arXiv:2507.08128*.

Mingyue Huo, Yiwen Shao, and Yuheng Zhang. 2026. TagSpeech: End-to-end multi-speaker ASR and diarization with fine-grained temporal grounding. *arXiv preprint arXiv:2601.06896*.

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka. 2020. Serialized output training for end-to-end overlapped speech recognition. In *Proc. Interspeech*, pages 2797–2801.

Naoyuki Kanda, Jian Wu, Yu Wu, Xiong Xiao, Zhong Meng, Xiaofei Wang, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Takuya Yoshioka. 2022. Streaming speaker-attributed ASR with token-level speaker embeddings. In *Proc. Interspeech*, pages 521–525.

Liang Lu, Naoyuki Kanda, Jinyu Li, and Yifan Gong. 2021. Streaming end-to-end multi-talker speech recognition. *IEEE Signal Processing Letters*, 28:803–807.

Ivan Medennikov, Taejin Park, Weiqing Wang, He Huang, Kunal Dhawan, Jinhan Wang, Jagadeesh Balam, and Boris Ginsburg. 2025. Streaming Sortformer: Speaker cache-based online speaker diarization with arrival-time ordering. In *Proc. Interspeech*, pages 5238–5242.

MOSI.AI, Donghua Yu, Zhengyuan Lin, Chen Yang, Yiyang Zhang, Hanfu Chen, Jingqi Chen, Ke Chen, Liwei Fan, Yi Jiang, Jie Zhu, Muchen Li, Wenxuan Wang, Yang Wang, Zhe Xu, Yitian Gong, Yuqian Zhang, Wenbo Zhang, Songlin Wang, and 5 others. 2026. MOSS transcribe diarize technical report. *arXiv preprint arXiv:2601.01554*.

Bingshen Mu, Pengcheng Guo, Zhaokai Sun, Shuai Wang, Hexin Liu, Mingchen Shao, Lei Xie, Eng Siong Chng, Longshuai Xiao, Qiangze Feng, and 1 others. 2025. Summary on the multilingual conversational speech language model challenge: Datasets, tasks, baselines, and methods. *arXiv preprint arXiv:2509.13785*.

Taejin Park, Ivan Medennikov, Kunal Dhawan, Weiqing Wang, He Huang, Nithin Rao Koluguri, Krishna C. Puvvada, Jagadeesh Balam, and Boris Ginsburg. 2024. Sortformer: Seamless integration of speaker diarization and ASR by bridging timestamps and tokens. *arXiv preprint arXiv:2409.06656*.

Jing Peng, Yucheng Wang, Bohan Li, Yiwei Guo, Hankun Wang, Yangui Fang, Yu Xi, Haoyu Li, Xu Li, Ke Zhang, Shuai Wang, and Kai Yu. 2025. A survey on speech large language models for understanding. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–32.

Zhiliang Peng, Jianwei Yu, Yaoyao Chang, Zilong Wang, Li Dong, Yingbo Hao, Yujie Tu, Chenyu Yang, Wenhui Wang, Songchen Xu, and 1 others. 2026. VIBEVOICE-ASR technical report. *arXiv preprint arXiv:2601.18184*.

Desh Raj, Daniel Povey, and Sanjeev Khudanpur. 2023. SURT 2.0: Advances in transducer-based multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3800–3813.

Desh Raj, Matthew Wiesner, Matthew Maciejewski, Paola Garcia, Daniel Povey, and Sanjeev Khudanpur. 2024. On speaker attribution with SURT. In *Proc. Speaker and Language Recognition Workshop (Odyssey)*, pages 91–98.

Andrew Reece, Gus Cooney, Peter Bull, Christine Chung, Bryn Dawson, Casey Fitzpatrick, Tamara Glazer, Dean Knox, Alex Liebscher, and Sebastian Marin. 2023. The CANDOR corpus: Insights from a large multimodal dataset of naturalistic conversation. *Science Advances*, 9(13):eadf3197.

Mohan Shi, Xiong Xiao, Ruchao Fan, Shaoshi Ling, and Jinyu Li. 2025. Train short, infer long: Speech-LLM enables zero-shot streamable joint ASR and diarization on long audio. *arXiv preprint arXiv:2511.16046*.

Elizabeth Shriberg, Andreas Stolcke, and Don Baron. 2001. Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In *Proc. Eurospeech*, pages 1359–1362.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. SALMONN: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Weiqing Wang, Taejin Park, Ivan Medennikov, Jinhan Wang, Kunal Dhawan, He Huang, Nithin Rao Koluguri, Jagadeesh Balam, and Boris Ginsburg. 2025. Speaker targeting via self-speaker adaptation for multi-talker ASR. In *Proc. Interspeech*, pages 5498–5502.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, and 1 others. 2025a. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*.

Kai-Tuo Xu, Feng-Long Xie, Xu Tang, and Yao Hu. 2025b. FireRedASR: Open-source industrial-grade mandarin speech recognition models from encoder-decoder to LLM integration. *arXiv preprint arXiv:2501.14350*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Han Yin, Yafeng Chen, Chong Deng, Luyao Cheng, Hui Wang, Chao-Hong Tan, Qian Chen, Wen Wang, and Xiangang Li. 2026. SpeakerLM: End-to-end versatile speaker diarization and recognition with multimodal large language models. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 34467–34475.

Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, Xiong Xiao, and Fil Alleva. 2018. Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks. In *Proc. Interspeech*, pages 3038–3042.

Fan Yu, Shiliang Zhang, Yihui Fu, Lei Xie, Siqi Zheng, Zhihao Du, Weilong Huang, Pengcheng Guo, Zhijie Yan, Bin Ma, and 1 others. 2022. M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6167–6171.

Binbin Zhang, Chengdong Liang, Shuai Wang, Xuelong Geng, Zhao Guo, Haoyu Li, Hao Yin, Xipeng Yang, Pengshen Zhang, Changwei Ma, and Lei Xie. 2025. WEST: LLM-based speech toolkit for speech understanding, generation, and interaction. *arXiv preprint arXiv:2509.19902*.