ONE TOKEN PER SPEAKER: AN OMNI-NATIVE, EXCHANGEABLE ENTITY INTERFACE FOR CONVERSATIONAL MEMORY ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speaker-specific memory requires linking utterances to the correct person. We investigate whether an omni model can derive and use speaker identity from its own audio features without a separate speaker encoder. A learned pooler over frozen Qwen2.5-Omni features produces one continuous token per enrolled speaker. With LoRA adaptation, speaker identification for session-memory retrieval reaches 97.3%, compared with 24.8% for mean pooling of the same features. Without target-domain adaptation, accuracy reaches 74.3% on VoxCeleb1 and 58.3% on AliMeeting. Deterministic ordering preserves predictions when unchanged enrollment entries are reordered.

Reliable identification does not, however, ensure effective use in dialogue. Under controlled candidate scoring, a continuous identity cue changes only one of 160 decisions relative to no identity. Providing the same resolved identity as text reduces episodes with wrong-person writes from 77.5% to 12.5%. The tested speaker-side and model-side adaptations do not close this gap. Over 30-turn meeting sequences, cosine matching over the same tokens roughly halves wrong-person writes relative to in-model use, although its errors persist longer.

Mapping cosine-matching results to existing candidate embeddings matches the gold-identity reference without additional parameters or training. This result is limited to the tested single-token candidate setting; the free-generation check ran but did not establish safety, reaching 25 of 40 valid decisions against a pre-registered floor of 32. Together, these findings demonstrate useful omni-native speaker representations while identifying their connection to downstream decisions as a remaining challenge.

1 INTRODUCTION

Recent omni models support real-time spoken interaction (Yu et al., 2025; Huang et al., 2026). In multi-party conversations, a model must identify who is speaking and retrieve information associated with that person. Conversational content alone is insufficient: two speakers may make the same request but have different conversational histories.

Speaker-aware language models combine linguistic content and speaker information for recognition, diarization, and personalized responses (Yin et al., 2026; Nam et al., 2026; Fu et al., 2025). These approaches often rely on a dedicated speaker encoder. For example, PAChat (Fu et al., 2025) projects speaker embeddings from an off-the-shelf diarization toolkit, together with Whisper features, into a frozen Llama. We ask whether an omni model's own audio features can support speaker-specific memory retrieval without an additional speaker model. Answering this question requires evaluating both the identity information carried by these features and the language model's ability to use it.

We introduce ENTITYTOKEN, which represents each enrolled speaker with a single continuous *entity token*. A pooler trained with speaker-paired supervision aggregates features from the frozen Qwen2.5-Omni audio encoder. In the memory task, each token is paired with a session-specific textual value, and query recordings are held out from enrollment. We adapt the language component with LoRA to retrieve the value associated with the query speaker, keeping the pretrained weights

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.

frozen. A fixed ordering of the token–value pairs ensures that re-enumerating the same enrolled entries leaves the model input unchanged.

We make three contributions:

- **One omni-native token supports speaker-specific memory retrieval.** With the same LoRA adaptation settings, learned pooling improves speaker identification accuracy from 24.8% with mean pooling of the same features to 97.3%, within 1.8 percentage points of the best dedicated speaker encoder (99.1%). Direct cosine lookup on the learned tokens achieves 99.0%, exceeding the LLM backend and showing that the backend offers no accuracy advantage in this task.
- **The approach supports cross-domain transfer and cross-model replication.** The method achieves zero-shot identification accuracies of 74.3% on VoxCeleb1 and 58.3% on room-disjoint AliMeeting, with a substantial decline on Mandarin meetings. Adapting only the LLM backend raises AliMeeting accuracy to 70.1%, recovering part of this loss. We also evaluate single-utterance enrollment in chronological meeting recordings and reproduce the approach on MiniCPM-o 4.5.
- **A parameter-free Omni-native speaker matching improves downstream use of identity.** Under candidate scoring, a continuous identity token is less effective than the same resolved identity supplied as text, whereas free generation benefits from the token. This limitation therefore depends on the decision rule. Computing the identity comparison outside the candidate-scoring path and mapping its result to candidate-token embeddings matches the true-identity reference in the tested setup, without additional trainable parameters or training.

2 Related work

Speaker-aware spoken language models. Prior work incorporates speaker information into language models (Yin et al., 2026; Nam et al., 2026; Fu et al., 2025), with evaluations focused on recognition, diarization, and response quality. We study identity as a key for session memory, evaluating correct entry selection, prediction consistency under reordering the same registry, and content leakage.

Continuous representations for language models. Flamingo's Perceiver Resampler (Alayrac et al., 2022) and BLIP-2's Q-Former (Li et al., 2023) use learned queries to aggregate visual features into fixed-size representations for language-model conditioning. Prefix-tuning (Li & Liang, 2021) and prompt tuning (Lester et al., 2021) instead learn task-specific continuous prompts while keeping the language model frozen. These prompts need not have a faithful discrete-token interpretation (Khashabi et al., 2022), and task-trained projectors can transfer them across models on related tasks (Su et al., 2022). Our pooler produces a representation for each speech utterance using speaker-discriminative supervision. We then freeze the pooler and adapt the language model to use these representations for speaker-specific memory retrieval. Section 4.2 examines their use in downstream decisions under candidate scoring.

Evaluation depends on whether speaker identities are anonymous or registered. Anonymous diarization allows speaker-label permutations during scoring (Nam et al., 2026), whereas fixed-registration systems such as HumanOmni-Speaker (Bai et al., 2026) require predictions to match registered identities. Our registry follows the latter setting, but derives its keys from frozen tokens rather than participant names. Renaming a participant therefore leaves the associated key unchanged.

Full-duplex and omni interaction systems. Real-time duplex systems (Yu et al., 2025; Huang et al., 2026) and omni interaction systems (Yan et al., 2026; Hu et al., 2026a) handle speaker identity separately from the model's cross-turn state. VoiceMem (Xie et al., 2026) develops streaming long-term memory while taking speaker identity as an input, rather than modeling and evaluating how that identity is obtained. This complements our focus on linking audio-derived identity to session memory. Full-duplex interaction motivates our setting, but duplex control is outside the scope of our evaluation.

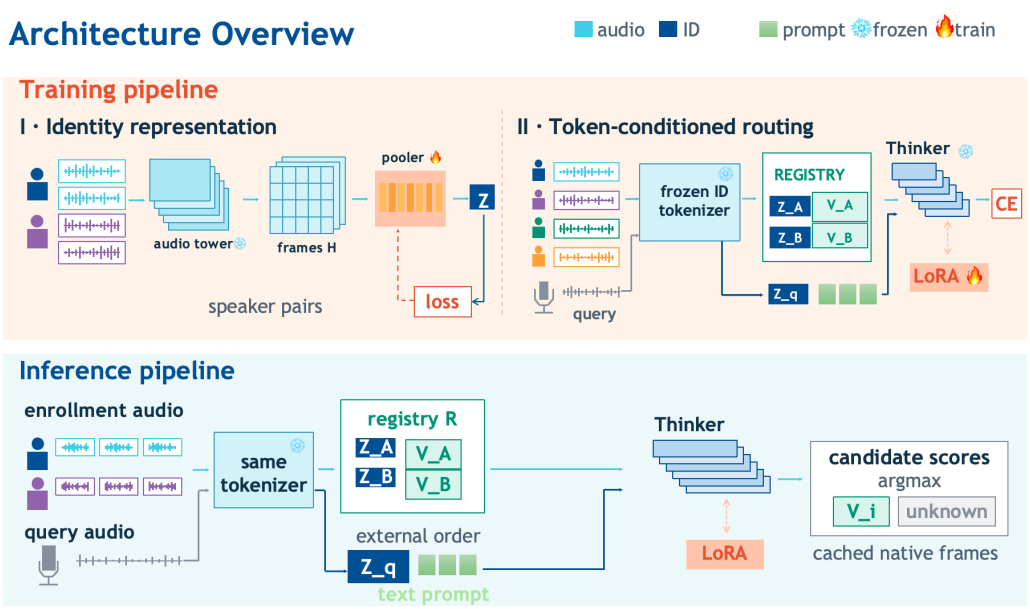

Figure 1: **Identity-token construction and Thinker input.** *(A)* A learned query cross-attends over frozen base model audio frames; the added attention pooler and residual MLP emit one L2-normalised continuous identity token ($K{=}1$) per utterance. The pooler is trained in Stage I and frozen thereafter. *(B)* Three enrollment utterances per speaker pass through the same tokenizer; their tokens are averaged and renormalised into that speaker's single registry entry Z_i, so the one-token-per-speaker interface is formed here rather than in (A). Entries are paired outside the model with session memory values V_i in the fixed-order registry of Section 3.2, ordered by a deterministic digest of the frozen token bytes. Query audio yields Z_q. Stage II trains only the q/v LoRA adapter; the pooler and the backbone stay frozen. Cyan denotes audio, navy identity tokens, green registry values, and coral text-prompt embeddings; snowflakes and flames mark frozen and trainable parameters. Equal-sized offset rectangles denote repeated layers, not changing tensor dimensions; speaker icons and waveforms are schematic. The output shown is the fixed candidate readout of Section 3.4 — an argmax over the five candidate token ids — not the length-normalised candidate-string score defined separately in Section 4.2, and not open-ended generation. Overlap diarization, on-line slot creation, vision, and full-duplex control are out of scope here.

Target-speaker ASR and diarization. Target-speaker extraction and speaker-attributed ASR (Wu et al., 2026; Zhang et al., 2026; Hu et al., 2026b) use speaker information for segment-level speech processing rather than maintaining speaker-specific memory across turns. Our method instead associates speaker tokens with session state. The intervention analysis in Section 4.3 shows that improved speaker retrieval does not necessarily improve memory selection by the language model. We do not incorporate these systems or claim improvements in ASR or diarization performance.

3 The EntityToken Interface

3.1 Setup

For each enrollment utterance R_e from speaker e, the frozen audio encoder produces a sequence of frame-level features. A learned pooler aggregates these features into one continuous token, replacing mean pooling. We average the tokens from three enrollment utterances and renormalize the result to obtain the speaker's entity token $Z_e \in \mathbb{R}^d$ (Figure 1).

The *LLM backend* is Qwen2.5-Omni's Thinker, with frozen pretrained weights and a parameter-matched adapter. Given a registry $\mathcal{R} = \{(Z_i, V_i)\}_{i=1}^{n}$ and a query token Z_q, it retrieves the value associated with the query speaker or returns *unknown*. We evaluate this bounded memory-retrieval task, rather than open-ended generation or full-duplex interaction.

The injected form is an implementation choice, not a commitment. Entity tokens enter the LLM backend by splicing continuous vectors at marker positions, and we verified this is interchangeable with carrying the same tokens *in the vocabulary*: writing each session's minted tokens into reserved embedding rows and referencing them by id produces bit-exact identical logits (maximum difference 0.0) and identical accuracy (.973) in the seed-17 configuration where we ran the check, with untied input/output embeddings keeping the rows out of generation. A vocabulary-resident identity token is thus a drop-in equivalent; by the same bit-exactness, vocabulary residence cannot by itself supply the missing correspondence Section 4.2 measures.

3.2 Invariance to registry order

Before constructing the model input, we sort the token–value pairs using keys computed solely from the frozen token representations Z_i. These keys are deterministic and unique within the registry; they do not depend on speaker names or other textual identifiers. No cryptographic properties are required.

Any permutation of the same registry therefore produces an identical input sequence and, under deterministic inference, the same prediction (Proposition 1, Appendix A). This invariance follows from input construction rather than a property learned by the LLM backend.

The guarantee applies only to reordering fixed entries. It does not extend to changes in enrollment recordings, token values, or registry membership, including entity removal. Key collisions violate the uniqueness assumption and are treated as implementation failures.

3.3 Design motivation: locality and rate

Two theoretical results motivate the design under explicit assumptions (Appendix A). Theorem 1 shows that permutation equivariance, together with single-slot intervention locality, implies shared per-slot processing. Our experiments establish only invariance to reordering a fixed registry; they do not establish locality of the underlying model state.

By Fano's inequality, distinguishing among $N{=}1{,}172$ uniformly distributed identities with at most 5% error requires approximately 9.4 bits of identity information (Theorem 2). This is a necessary lower bound, not a sufficient coding budget or a guarantee that the token contains only identity information. We therefore evaluate both memory-retrieval accuracy and transcript decodability under a calibrated post-pooling additive white Gaussian noise (AWGN) channel at specified rate budgets (Section 4.3). These are operational measurements, not estimates of mutual information.

3.4 Training and evaluation setup

We select 11,720 utterances from 1,172 speakers, with three enrollment and seven query utterances per speaker. The training, development, and test sets contain 820, 176, and 176 speakers, respectively, with no speaker overlap.

The primary learned- and mean-pooling comparisons use the same audio-feature cache, session construction, LLM backend, adaptation budget, and test queries. The learned pooler additionally receives speaker-supervised training before language-model adaptation. Each session combines recorded speech with fixed text templates. Memory values are assigned independently of the transcripts, so transcript content alone does not determine the target value (Appendix X.3).

We use three random seeds (17, 29, and 43) and compute paired-bootstrap confidence intervals clustered by held-out speaker. Full setups, hyperparameters, computational costs, and convergence results are provided in Appendices X and X.10.

4 Results

4.1 Speaker identification with omni-native tokens

Speaker identification. Learned omni-native pooling improves identification accuracy over mean pooling by 72.5 percentage points, with all speaker-clustered CIs excluding zero. The same single-

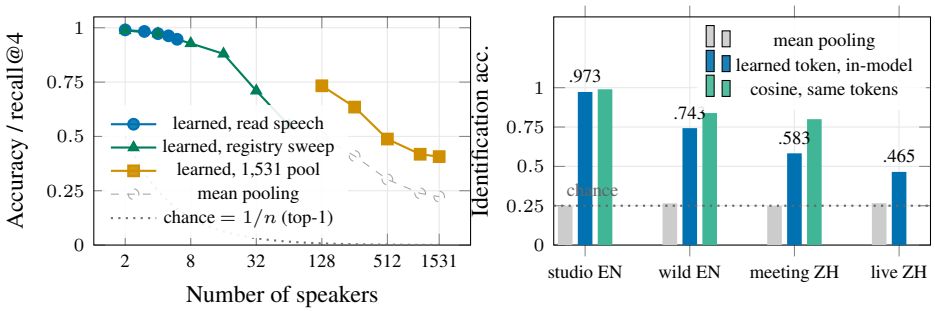

Figure 2: **Effects of registry size and domain shift.** *Left:* Speaker identification accuracy decreases from 99.0% to 94.7% as the number of enrolled speakers increases from two to six. A separate experiment extends the registry to 64 speakers (Table 16). The population-scale experiment instead measures retrieval recall@4, reaching 40.7% on the audited 1,531-speaker pool while remaining above mean pooling (Table 15). *Right:* Four-speaker identification with the frozen interface across LibriSpeech, VoxCeleb1, AliMeeting, and chronological meeting evaluation; the last setting uses single-utterance enrollment. Mean pooling of the same features remains near chance. Cosine matching on the learned tokens exceeds the LLM backend by 1.7, 10.2, and 20.7 percentage points on LibriSpeech, VoxCeleb1, and AliMeeting, respectively. (Appendix Q).

token representation ($K{=}1$) also improves standalone speaker retrieval by 46.9 percentage points. We additionally evaluate sensitivity to the order of enrolled entries. With deterministic ordering, prediction consistency under reordering is 100%, compared with 97.2% without it (Table 11; pooling controls in Appendix S).

Accuracy is similar across lexical-overlap quartiles and remains 92.4% at Dice overlap ≤ 0.1, while mean pooling stays near chance across groups. The separate seven-way random-content control scores 15.26%, consistent with chance (Appendix X.6).

On the same learned tokens, cosine matching significantly outperforms the LLM backend in every seed. An MLP matcher performs similarly to cosine, while a bilinear matcher is comparable to the backend (Appendix R). These results support the speaker-discriminative quality of the tokens, but do not establish an accuracy advantage for the LLM backend.

Generalization under domain shift. Under zero-shot domain shift, learned pooling remains above mean pooling, with every per-seed CI for the gain excluding zero (Figure 2; Appendix L). With the encoder and pooler frozen, backend-only mixture adaptation raises AliMeeting accuracy from 58.3% to 70.1%; gains on the official test and evaluation partitions are 10.4 and 7.7 points. On 69 chronologically replayed AliMeeting sessions with single-utterance enrollment (1,154 queries), learned tokens outperform chance and mean pooling in all nine seed-by-speaker-count configurations, reaching 46.5% with four speakers. The same enrollment change reduces LibriSpeech accuracy only from 97.3% to 92.7%, suggesting that enrollment size alone does not explain the meeting-domain gap (Appendix W).

Dedicated speaker encoders remain stronger under matched conditions (99.1% versus 97.3%), including after parameter matching. With in-domain backend training, they reach 96.3% on AliMeeting and 91.3% on VoxCeleb1 (Appendix Q). Cosine matching's advantage over the LLM backend increases from 1.7 points on LibriSpeech to 10.2 on VoxCeleb1 and 20.7 on AliMeeting (both OOD $p{<}10^{-15}$), consistent with the six prespecified cases (Appendix M).

Scaling and cross-model evaluation. Learned tokens outperform mean pooling across 2–64 enrolled speakers, with accuracy decreasing from 98.8% to 55.0% (Table 16). Population-scale retrieval and the withdrawn, unauditable 100% recall claim are documented in Appendices U and O. The preregistered tests detect no degradation on MMLU, VizWiz, or ASR within their sensitivity (Appendix T). An independent MiniCPM-o 4.5 implementation achieves 97.2% identification and 89.8% retrieval accuracy, although its language component remains within the Qwen family. The

Table 1: Identity-aware dialogue on real conversation. "WPW" is the fraction of episodes with a wrong-person write; dev is EasyCom, OOD is room-disjoint AliMeeting at a frozen operating point (N=40/arm, dependent-field setup; Appendix E).

	Dev WPW ↓	OOD policy acc. ↑	OOD WPW ↓
No identity	.825	.775	.975
Native token, in-model	.800	.750	.925
Native token, cosine routing	**.050**	**.900**	**.100**
Gold identity (ceiling)	.000	1.000	.000

Kimi-Audio configuration fails the acceptance criterion: accuracy is 19.0%, rising to 43.5% after token erasure (Appendix Q).

4.2 Identity-aware dialogue on recorded conversations

We evaluate *latest-statement recall* and *target service* (responding only to a designated speaker) on real recordings. Holding the model, audio, session context, and decoding budget fixed, we compare four identity conditions: none, native tokens, cosine matching on the same tokens, and gold identity. We measure action correctness and wrong-person-write (WPW) rate. Blinded machine labels confirmed by humans indicate low recall quality across conditions (0.14–0.21), with no clear differences (Appendix I).

The primary comparison uses the base LLM backend. Controls with either the routing adapter or an application-trained adapter show no significant WPW benefit from continuous tokens. Providing the resolved identity as text reduces WPW to 22.5% (p=.0039), indicating that the form of identity input matters in this setting (Appendix E.1).

Under candidate scoring, cosine matching on the native tokens yields the lowest WPW among non-oracle conditions (Table 1). Direct token insertion shows no significant improvement over no identity (p=1.0 on dev), despite verified insertion on every turn (432 dev, 160 OOD) and changed text outputs on approximately one third of turns. The WPW gap between direct insertion and cosine matching reaches 82.5 percentage points out of domain.

Under free generation, however, native tokens reduce WPW from 62.5% to 37.5%, with all ten discordant episodes favoring the token condition (p=.002; Appendix E.1). The limitation therefore depends on the decision rule, rather than indicating a general inability to use the tokens.

Cosine matching resolves identity before inference and supplies the selected slot index as text, without names, transcripts, or an external speaker encoder. The in-model condition must perform the matching itself. Deterministic registry ordering preserves invariance to reordering fixed entries throughout the pipeline (Section 3.2).

Separating identity decisions from input format. We cross the identity-decision mechanism with the input format (Figure 3A). With external routing, text and continuous inputs yield identical episode-level outcomes (5.0% WPW). When the model decides, a text pointer to the cosine-selected speaker reduces WPW to 12.5%, whereas a continuous pointer at the same injection site yields 77.5%, compared with 80.0% without a pointer (p=1.0). The text–continuous gap is 65 percentage points, with all 26 discordant episodes favoring text (p=3.0×10^{-8}). At the frozen operating point, this gap widens to 77.5 points on unseen speakers in room-disjoint Mandarin far-field recordings: 87.5% versus 10.0% WPW, with 33 of 35 discordant episodes favoring text (p=3.7×10^{-8}).

To quantify token effects beyond the selected decision, we examine the recorded candidate log-probabilities. The analysis is specific to the following argmax decision rule.

Definition 1 (Candidate-scoring decision rule)**.** *Given context c (audio, any injected identity material, and text) and candidate strings $\{s_1, \ldots, s_m\}$, tokenize each s_k into $(t_k^1, \ldots, t_k^{L_k})$, score it teacher-forced under the frozen model f as* $\mathrm{score}(s_k \mid c) = \frac{1}{L_k} \sum_j \log p_f(t_k^j \mid c, t_k^{<j})$*, and decide* $\hat{k} = \arg\max_k \mathrm{score}(s_k \mid c)$.

The rule averages token log-probabilities, and identity information can affect the decision only through the candidate scores. Across 160 paired turns, injecting the token produces little change in the candidate distribution: mean KL divergence is 0.0021 (median 0.0005), 94% of values are below 0.01, and 159 of 160 argmax decisions remain unchanged. Among 130 turns pointing to a valid slot, both conditions select the named slot on exactly the same 90 turns. In contrast, the text pointer yields a mean KL of 0.0149 and selects the named slot on 129 of 130 turns, confirming that the measurement detects a change when one occurs. Thus, the observed token effect is limited to one argmax change in 160 turns, with mean KL about seven times smaller than for text; this is not evidence of an exactly zero effect.

Providing candidate slot names alongside the token yields a 70.0% error rate. The 10-percentage-point reduction does not meet our pre-registered criterion for a detectable improvement at this sample size, so this control does not establish output formatting as the explanation.

The continuous input channel is functional: candidate-derived vectors improve decisions, but native identity tokens perform similarly to random vectors despite 92.5% retrieval R@1 and a typical input norm (0.821, at the 46th percentile of the base model's embedding norms). This suggests a mismatch between identity tokens and the scored candidates, rather than insufficient speaker information or inappropriate input scale (Figure 3B; further controls in the supplementary material).

This comparison has a limited interpretation. The pooler was not trained to align its outputs with session-specific labels, whereas the text condition supplies the correct candidate string verbatim. Candidate scoring may therefore favor explicit label matching. Our experiments do not separate this advantage from difficulty grounding continuous identity representations.

Speaker matching without task-specific training. The frozen base model achieves 81.7% accuracy on a minimal three-speaker task using raw prototypes from clean speech (49/60 trials; 33.3% chance, $p{=}1.2{\times}10^{-12}$; Figure 3C). Performance remains high under candidate reordering and cross-chapter evaluation, supporting continuous speaker matching in this controlled setting.

The decline in far-field, multi-turn dialogue may reflect changes in both audio conditions and the decision rule, including constrained JSON and multi-token labels. Our intermediate test cannot reliably separate these effects: it includes only four speakers and four candidate sets, with 50.0% accuracy (exact $p{=}.0499$) and one cluster at chance under each of two groupings.

Speaker-side residual adaptation does not help. We train a gated low-rank residual on each enrollment prototype and the query, keeping all other components frozen. Training converges, but clean-setting accuracy drops from 81.7% to 63.3%, and conversational performance matches the random-vector baseline. This degradation resembles the joint-training result in Section 4.3, with only one-thousandth as many trainable parameters (Appendix C).

Rejecting low-confidence turns improves performance in conditions that successfully use identity information, whereas native-token and no-identity curves nearly coincide (Appendix J).

Error persistence across turns. We evaluate 30-turn AliMeeting windows from 14 meetings in three rooms, with speakers enrolled beforehand and settings fixed on EasyCom. Wrong-person writes account for 52.0% of writes with native tokens and 54.1% without identity, a statistically indistinguishable difference, versus 23.1% with cosine matching (CI [11.7, 38.0]%) and 0% with gold identity. However, cosine errors persist longer: 2.68 turns versus 1.08–1.40 for the in-model conditions, with meeting-paired CIs excluding zero. They also remain unoverwritten at window end more often (12.3% versus 2.4–3.6%). Persistence ends at the next write, usually another incorrect one, so shorter persistence does not demonstrate repair. Pilot design, exclusions, and meeting-clustered analyses are reported in Appendix H; three rooms do not establish room-level robustness.

These findings are consistent with a mismatch between speaker representations and candidate scoring. Cosine matching supplies an explicit identity decision, whereas direct token injection provides no comparable benefit under this setting. Joint training also fails to close the gap and can degrade the representation (Section 4.3).

The development split contains only two independent target speakers, making speaker-clustered intervals uninformative. Generalization therefore rests on the OOD split, with different speakers, corpus, and language; within-split comparisons use exact McNemar tests. Under free generation,

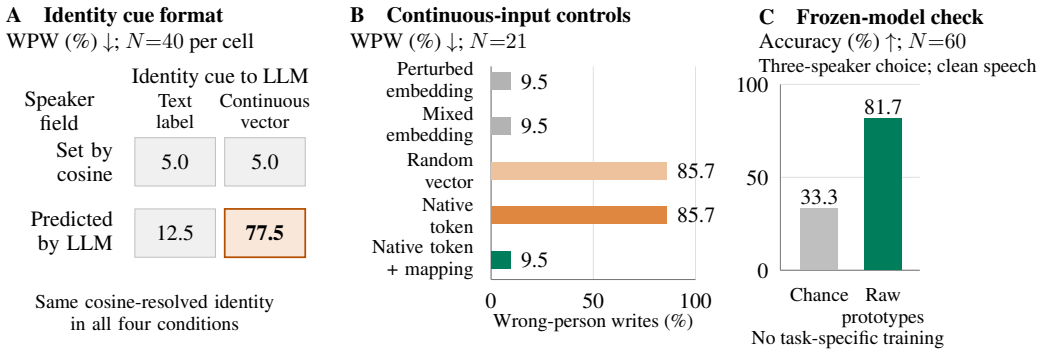

Figure 3: **Identity cues, continuous inputs, and speaker matching.** WPW measures episodes with a wrong-person write. *A:* all conditions use cosine-resolved identity. The speaker field is either set directly (top) or predicted by the LLM from a text or continuous cue (bottom). *B:* single-token candidates; the same 21 episodes selected for native-token injection on every turn. The first two inputs perturb the target candidate embedding or average it with an unrelated embedding; the random control is norm-matched. Mapping converts native-token cosine scores into a weighted sum of candidate embeddings, outside the LLM's scoring path. *C:* three-speaker identification without task-specific training (49/60 correct). Setups and metrics differ across panels; C does not separate domain effects from setup effects.

cosine matching produces more episodes with non-target writes than native tokens (60% versus 20%; Appendix E.1), so its advantage does not extend to every error type.

Mapping identity to candidate embeddings without training. We compute cosine similarities between query and enrolled tokens and use the resulting weights to combine frozen candidate embeddings. No parameters are added or trained. On the 21-episode subset with injection on every turn, WPW falls from 85.7% to 9.5%, matching the gold-identity hint (all 16 discordant episodes favor the mapping; exact McNemar $p{=}3.05{\times}10^{-5}$). Matching still occurs outside the LLM's scoring path. This result uses single-token candidates, unlike the multi-token slot names in Table 1. The free-generation check ran but did not establish safety: 25 of 40 decisions were valid, against a pre-registered floor of 32 (Appendix F).

4.3 Boundaries and Negative Results

Three pre-registered analyses identify limitations of the interface (Appendix G).

Rate and content leakage. Adding noise trades off identification accuracy and transcript recoverability. At the budget meeting the identification tolerance, a same-speaker, seven-candidate transcript test still achieves 82.3% accuracy, versus 15.3% for the random-content control. Rate control therefore does not isolate identity, and stored tokens should be treated as sensitive data.

Retrieval and downstream identification. Textual inversion improves standalone retrieval from 90.9% to 93.2% but reduces identification by the frozen LLM backend from 96.4% to 91.8%. Retrieval gains therefore need not translate into better model-based decisions.

Joint training. Unfreezing the pooler reduces accuracy to near chance at learning rates 10^{-4} and 10^{-5} (25.1% and 24.5%, versus 97.3%). At 10^{-6}, it still does not outperform staged training with a frozen pooler; a doubled-budget frozen control also fails to close the gap. Increasing the token count from $K{=}1$ to $K{=}4$ provides no improvement in the tested settings.

Representation quality and downstream use. These results support the speaker information carried by native tokens, but show no benefit from the tested in-model adaptations. Cosine matching increasingly outperforms the LLM backend under domain shift and reduces wrong-person writes under candidate scoring (Section 4.2).

The channel experiments, together with 92.5% retrieval R@1, suggest that correspondence between identity tokens and scored candidates is a remaining limitation. They do not, however, establish a

single cause: the advantage of verbatim text labels has not been isolated, nor have acoustic-domain effects been separated from changes in the interaction setup.

5 Discussion and limitations

Our results support the use of omni-native speaker tokens for bounded session-memory retrieval on held-out speakers. Deterministic ordering guarantees consistency when unchanged enrollment entries are reordered, not when recordings or enrolled speakers change. Memory values remain text in context, so storing speaker-specific content in latent state is not tested. The tokens retain transcript information and are not identity-only; privacy protection is not established. The in-model condition accepts 94–100% of never-enrolled speakers (Appendix K), showing that accurate identification among enrolled speakers does not ensure reliable rejection of unfamiliar ones.

All experiments are offline and bounded, primarily using forced choice with separate free-generation tests. They do not establish open-ended response quality, memory formation from free dialogue, full-duplex latency or interruption handling, visual identity, or overlapping-speaker separation. Most results use one omni model, and the second implementation shares its LLM family; generalization across backbone families is therefore not established.

In the three-speaker identification experiment, reversing the prototype listing order preserves the predicted speaker in 70.0% of cases without the residual and 71.7% with the trained residual. Deterministic ordering (Section 3.2) prevents this input variation but does not make the model itself order-invariant. The dialogue comparisons also retain reference audio in every condition, and the separating ablation yielded no interpretable comparison (Appendix C); the tokens' additional benefit over that audio remains unresolved.

6 Conclusion

We show that a frozen omni audio encoder, combined with a learned pooler, supports speaker identification and bounded session-memory retrieval using one continuous token per speaker. The method performs well on 176 held-out speakers, and deterministic ordering preserves predictions when unchanged enrollment entries are reordered. Independent implementations succeed on two of the three tested models. Cosine matching over these tokens supports identity-dependent decisions in the evaluated dialogue tasks.

Direct token use remains less reliable in the tested dialogue setting, and neither speaker-side residual training nor model-side adaptation closes the observed gap. Mapping cosine-matching results to candidate embeddings matches the gold-identity reference without additional parameters or training, but only under the tested candidate-scoring setup; the free-generation check ran but did not establish safety, reaching 25 of 40 valid decisions against a pre-registered floor of 32. These findings distinguish learning useful speaker representations from making them reliably usable in dialogue. The next step is to train and evaluate identity tokens embedded directly in streaming conversation, beyond explicit registry lookup.

Reproducibility statement

All primary experiments use three fixed random seeds (17, 29, 43). The corpus is split speaker-disjointly into 820 train / 176 dev / 176 test speakers (1,172 total; three enrollment and seven query utterances per speaker); the split is fixed and will be released as a manifest file. Confidence intervals are paired bootstrap intervals clustered by held-out speaker.

The six result files behind the headline numbers are frozen, and we list the checksum of each so a reader can confirm they match what we report:

Result file	checksum
Reference LLM backend `aggregate.json`	`a6b8d1cf54733f40feea0893053a621e b8c5d4e1cff9ce6698a7699ae403233b`
Full rate `summary.json`	`76e4747e0a4673554ef2741d2c7150dc 5f5483f016136e303f6bafe13d5b982c`
Content check `summary.json`	`26360c762db0ab3be28ecaba1c595563 a76b7867115e81e0a2bdb338796184ad`
TI-to-LLM backend `summary.json`	`cc1b14cf0cca67780d8c081effe9da1c 36a2e393227e22e735a39f9306538b2d`
TI amortized-init `metrics.json`	`20b63990fbc7d16c52f485719d7c39ba 8cd199f22eaa4fd211904cc64a0a88cc`
TI random-init `metrics.json`	`8d36249516ccac61b8b26d6f55a84ae6 68cf7611bd34644c97c91563d5a37d9c`

The field `minimum_bits_within_2pp_on_every_seed` in the rate payload reflects an implementation decision within the tested grid ($\{4096, 8192, 16384\}$ bits); it is not evidence of a universal or information-theoretic minimum. Accepted wording: "operational tolerance point under this setting."

Code and split manifests will be released at camera-ready. Hyperparameter details (LoRA rank, learning rate, steps, registry size) appear in Appendix X.

ETHICS STATEMENT

Biometric-adjacent data. Reference utterances link a voice recording to a named session identity. Although we evaluate on a public speaker corpus and our entity tokens are not human-interpretable, voice recordings are biometric-adjacent personal data. Deployment of a system based on this interface would require informed consent from all enrolled speakers, a stated retention and deletion policy for stored tokens, and compliance with applicable data protection regulations (e.g. GDPR, CCPA, and sector-specific rules).

Residual content risk at the operational rate. Our rate experiment is designed to detect and report the content information that remains in the entity token after rate-controlling the channel. At the operational rate of 8,192 bits, a held-out transcript check achieves accuracy **0.8231**. This is not a performance target; it is a *measured risk*. It means that at the rate point where routing accuracy is preserved (.9677), a seven-candidate test still picks the right transcript far above its 15.26% random-content control. That is a discriminability measure, not an estimate of what fraction of the transcript is present; we do not have the latter. Any deployment must therefore treat stored entity tokens as sensitive data equivalent in risk to partial transcripts, not as anonymous speaker identifiers. Rate tightening reduces this leakage (check drops to 0.5747 at 4,096 bits) but incurs a routing cost; the tradeoff is explicitly quantified and must be decided in context.

No voice cloning or impersonation. The interface uses the entity token exclusively to route session memory to the queried entity in a forced-choice bounded task. No component of this system synthesizes, reconstructs, or approximates the reference speaker's voice. The entity token is an opaque vector index, not a generative voice representation. We did not attempt inversion, and we make no non-invertibility claim: recovering signal from frozen embeddings is an active attack literature (Morris et al., 2023), and stored tokens should be protected accordingly rather than assumed unrecoverable.

Use-case scope. The system is designed for personal-assistant and meeting-summary scenarios where a device owner enrolls known participants. It is not designed and should not be used for covert speaker identification, surveillance, or any application in which individuals are enrolled without knowledge.

USE OF LARGE LANGUAGE MODELS

LLM-based assistants were used substantially in this work, under author direction and with author verification: (i) *research ideation and execution* — experiment design, training and evaluation code,

and the execution of pre-registered runs were developed with LLM coding assistants; every quantitative claim traces to frozen, content-addressed artifacts with audited provenance, and acceptance thresholds were fixed before results were observed; the formal statements and proofs in the appendix were likewise drafted with LLM assistance and checked by the authors — all three rest on elementary or textbook arguments (permutation invariance of key-based ordering; Fano's inequality with the chain rule) that admit direct independent verification; (ii) *drafting and polishing* — manuscript sections were drafted and edited with LLM assistance and subsequently reviewed by the authors; (iii) *retrieval and discovery* — related work was located with LLM-assisted search, and all citations were verified programmatically against bibliographic databases. LLMs were *not* used to generate data: the session-memory corpus is a deterministic template composition over a fixed vocabulary (Appendix X), no LLM is invoked anywhere in the data-construction pipeline, and all speech audio comes from existing public corpora — no audio was generated. The authors take full responsibility for the content of this paper.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, 2022.

Detao Bai, Zhiheng Ma, and Xihan Wei. HumanOmni-Speaker: Identifying Who said What and When. *arXiv preprint arXiv:2603.21664*, 2026.

Dongjie Fu, Xize Cheng, Linjun Li, Xiaoda Yang, Lujia Yang, and Tao Jin. PACHAT: Persona-Aware Speech Assistant for Multi-party Dialogue. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://aclanthology.org/2025.emnlp-main.1492/.

Yibo Hu, Yu Qian, Mao Gu, Yingfan Tao, Yuhao Chen, Yongdong Luo, Zhuoqun Liu, Meiguang Jin, and Junfeng Ma. TLive-Omni: An Omni-Modal Understanding Model for E-Commerce Live Streaming. 2026a.

Yifan Hu, Peiji Yang, Zhisheng Wang, Yicheng Zhong, and Rui Liu. TellWhisper: Tell Whisper Who Speaks When. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026b. URL https://aclanthology.org/2026.acl-long.861/.

Muye Huang, Lingling Zhang, Xingyu Yu, Lei Shi, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jun Liu. DuplexOmni: Real-Time Listening, Seeing, Thinking, and Speaking for Full-Duplex Interaction. 2026.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3631–3643. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.266.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12448–12460. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.765.

KiHyun Nam, Jungwoo Heo, Siu Bae, Ha-Jin Yu, and Joon Son Chung. SpeakerLLM: A Speaker-Specialized Audio-LLM for Speaker Understanding and Verification Reasoning. 2026.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3949–3969. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.290.

Hao Wu, RongQi Han, Zhen Wang, Wei Liang, and Wei Xu. Diarization-Guided Qwen-ASR Adaptation for Multilingual Two-Speaker Conversational Speech. 2026.

Zhifei Xie, Jiaqi Lang, Ze An, Yifan Zhao, Dongchao Yang, Kai Li, Ziyang Ma, Mingbao Lin, Chunyan Miao, and Shuicheng Yan. VoiceMem: Streaming Dual-Brain Memory for Real-Time Interaction, 2026.

Ruiqi Yan, Wenxi Chen, Zhanxun Liu, Ziyang Ma, Haopeng Lin, Hanlin Wen, Hanke Xie, Jun Wu, Yuzhe Liang, Yuxiang Zhao, Pengchao Feng, Jiale Qian, Hao Meng, Yuhang Dai, Shunshun Yin, Ming Tao, Lei Xie, Kai Yu, Xinsheng Wang, and Xie Chen. SoulX-Duplug: Plug-and-Play Streaming State Prediction Module for Realtime Full-Duplex Speech Conversation. 2026.

Han Yin, Yafeng Chen, Chong Deng, Luyao Cheng, Hui Wang, Chao-Hong Tan, Qian Chen, Wen Wang, and Xiangang Li. SpeakerLM: End-to-End Versatile Speaker Diarization and Recognition with Multimodal Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i40.40745.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. SALMONN-omni: A Standalone Speech LLM without Codec Injection for Full-duplex Conversation. 2025.

Yiru Zhang, Hang Su, Lichun Fan, Ying Zeng, Chang Liu, Yifeng Wang, Yuquan Liang, Tao Li, Lian Li, Wenhao Yang, Jian Luan, Cong Zou, and Heng Qu. Xiaomi-CocktailASR-1 Technical Report. 2026.