PIECING IT TOGETHER: HARMONIZING PARTICIPANT-GROUNDED MEMORY IN MULTI-PARTY DIALOGUES

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language model (LLM) agents increasingly participate in large-scale collaborative projects, memory mechanisms for *multi-party* dialogues have emerged as a new challenge. However, existing works primarily rely on semantic-based retrieval from a two-party tradition, struggling to piece together a holistic view across groups and time. To address this challenge, we propose PIT from a participant-grounded perspective, which reframes multi-party memory retrieval as completing a query-specific puzzle by assembling scattered pieces. Specifically, we first construct an event–participant bipartite graph to derive participant-grounded event representations, which are further consolidated into coherent memory pieces. A query-specific router then explores their underlying dependencies and harmonizes them together into a coherent memory view. Extensive experiments on EverMemBench demonstrate consistent improvements across different LLMs, outperforming the strongest baselines by 6.71%–19.91%. Further analyses show that participant grounding generally improves event representations and retrieval reliability. The code can be available at https://anonymous.4open.science/r/PIT-E7F1.

1 INTRODUCTION

As LLM agents increasingly participate in collaborative project workflows, such as software development, online community discussions, and educational activities, *multi-party dialogue* is becoming an important mode of human–agent interaction (Eckhaus et al., 2025; Zhang et al., 2025b). As illustrated in Figure 1, agents in these settings must manage information exchanged among diverse participants (e.g., product managers, frontend and backend engineers, and operations engineers) across broader conversational history. This interaction structure fundamentally poses significant challenges for current agent memory.

Existing agent memory systems typically organize and retrieve historical interactions based on semantic relevance (Park et al., 2023; Zhong et al., 2024; Packer et al., 2024; Chhikara et al., 2025). However, unlike conventional two-party conversations that largely unfold within a single conversational context, relevant information in multi-party dialogue can be highly scattered. Different participants may freely engage in discussions across multiple groups, while the same topic may recur at different points in time. As a result, reconstructing the complete context for a given query requires LLM agents to harmonize these fragmented pieces of information, with participants serving as key anchors for tracing relevant evidence.



Figure 1: Illustration of multi-party agent memory system.

The recent advent of EverMemBench (Hu et al., 2026), a benchmark dedicated to agent memory in multi-party dialogue, opens up the opportunity for this emerging setting. We propose PIT (**P**iecing **I**t **T**ogether), which uses participants as threads to assemble fragmented memories into coherent context as completing a query-specific puzzle. Specifically, we first construct an event-participant bipartite graph that explicitly connects each event to its involved partici-

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.

pants. This structure traces evidence scattered across different conversations through shared participants. By propagating high-order information over this graph, PIT derives participant-grounded representations that preserve event semantics while capturing event–participant interactions and event–event dependencies mediated by shared participants. Furthermore, we consolidate contextually related events, transforming scattered events into memory pieces. To selectively assemble these pieces for a given query, a query-specific router progressively explores their dependencies and harmonizes the relevant evidence into a coherent memory context for answer generation.

Experiments on EverMemBench demonstrate that PIT consistently outperforms strong memory baselines based on different LLMs by 6.71%–19.91%. Further analyses show that participant grounding improves event representation quality by bringing query-relevant events closer together, and enhances retrieval reliability by increasing multi-hop event coverage while reducing response premature stopping and response content drift.

Our main contributions are summarized as follows:

1. We propose PIT, the first agent memory framework specifically designed for multi-party dialogue, which reframes multi-party memory as a query-specific puzzle-completion process.
2. PIT identifies participants as key threads connecting fragmented events and organizes them into memory pieces, hereby guiding query-specific memory assembly.
3. Extensive experiments across different LLMs demonstrate consistent improvements, Further analyses show that participant grounding generally improves event representations and retrieval reliability, reducing premature stopping and content drift in responses.

2 RELATED WORK

In multi-party dialogues, information is distributed across participants and groups, yet research on organizing it remains limited. To the best of our knowledge, Hu et al. (2026) present the only publicly available and formally published benchmark for LLM agent memory in this setting. In this section, we discuss recent works from two related topics as follows.

Agent Memory for Two-Party Dialogue. Memory enables LLM agents to retain and reuse information from previous interactions. Existing approaches can be grouped into three types: (1),**Flat Memory.** Many methods store dialogues, observations, or feedback as independent textual entries retrieved by recency, importance, or semantic similarity (Park et al., 2023; Zhong et al., 2024; Chhikara et al., 2025). (2),**Hierarchical Memory.** Other methods organize growing histories into hierarchical or progressively consolidated representations, improving retrieval efficiency while preserving broader context (Packer et al., 2024; Tan et al., 2025; Sun & Zeng, 2025; Zhang et al., 2026). (3),**Graph-Structured Memory.** Graph-based methods connect people, facts, and events to support relational and multi-hop retrieval (Gutierrez et al., 2024; Anokhin et al., 2025; Rasmussen et al., 2025; Xu et al., 2025; Lu et al., 2026; Shu et al., 2026). Despite these advances, most systems are designed for two-party dialogue and do not explicitly model multiple participants, limiting their ability to distinguish participant-specific information and connect events across conversations and groups.

Multi-Party Dialogue Understanding. Before the era of LLMs, several studies investigated multi-party dialogue from the perspective of textual understanding, but these works address a fundamentally different problem from agent memory. Existing studies of multi-party conversations largely focus on two groups of tasks. The first concerns conversation structure understanding, including speaker identification, addressee recognition, and reply-to prediction (Meng et al., 2018; Le et al., 2019; Gu et al., 2021; 2023). These tasks capture who speaks to whom and how utterances are related within a dialogue. The second concerns response selection and generation, which aim to select or produce an appropriate next response from the current dialogue context (Ouchi & Tsuboi, 2016; Zhang et al., 2018; Hu et al., 2019; Wang et al., 2020; Gu et al., 2022; Li & Zhao, 2023).

Both lines focus on individual dialogues, without explicitly modeling event-participant relations or organizing events across dialogues. Consequently, they are unsuitable for agent memory tasks in multi-party settings, PIT fills this gap by linking events to their participants and assembling related events across dialogues for each query.

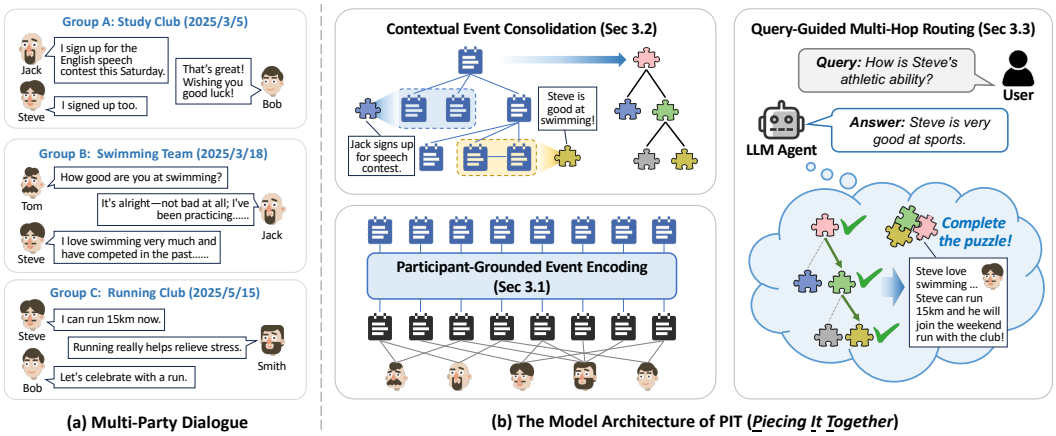

Figure 2: Overview of PIT. (a) Multi-party dialogues contain events across participants, groups and time. (b) PIT incorporates participant information into event representations, consolidates related events into pieces, and progressively assembles query-relevant pieces for answer generation.

Table 1: Extraction objects and their corresponding contents.

Extraction Object	Corresponding Content
Event	event_description, memory_tags, participants, point_in_time, index
Facts	subject, predicate, object, index

3 METHODOLOGY

In Figure 2, we present an overview of PIT. From multi-party dialogues, PIT extracts events, identifies facts, and enriches event representations with participant context (▷ Section 3.1). It then constructs a dynamic k-NN event graph and consolidates complementary events into pieces (▷ Section 3.2). Given a query, PIT follows useful connections from relevant events to assemble the required pieces, while pruning unrelated branches (▷ Section 3.3). The following sections detail these three stages.

3.1 PARTICIPANT-GROUNDED EVENT ENCODING (▷ ALGORITHM 1)

Session-Based Event and Fact Extraction. Given an unbounded input stream $\mathcal{X} = (x_1, x_2, \ldots)$, we partition it into sessions $\mathbf{E} = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_N\}$ using a window of length L, where $\mathcal{E}_i = \{x_t \mid (i-1)L < t \leq iL\}$. From each session, we extract events characterized by a description, participants, tags, and occurrence time, together with fine-grained facts represented as schemaless $(\text{subject}, \text{predicate}, \text{object})$ triples. Events and facts are retained chronologically, even when contradictory, to preserve a revisitable historical record. The detailed extraction object and its corresponding content can be found in Table 1.

Event-Participant Bipartite Graph. We construct a bipartite graph containing event nodes and participant nodes, with an edge connecting each event to its involved participants. This graph explicitly records who is associated with each event and connects events across conversations through shared participants.

Participant-Grounded Event Representation. We first encode the textual content of event i using Qwen3-Embedding-0.6B (Zhang et al., 2025a), yielding its initial embedding $\mathbf{e}_i$. We then apply LightGCN (He et al., 2020) over the event–participant bipartite graph to incorporate participant-mediated information, producing the enhanced event embedding $\hat{\mathbf{e}}_i$.

3.2 CONTEXTUAL EVENT CONSOLIDATION (▷ ALGORITHM 2)

Event Graph Construction. Using the extracted events, we construct a dynamic event-level k-NN memory graph. The similarity between events is measured by the cosine similarity of their enhanced representations $\hat{\mathbf{e}}$. Moreover, for each incoming event, we select its top-k most similar historical events as neighbors. Each selected historical event is also updated: the new event is added directly if it has fewer than k neighbors; otherwise, it replaces the least similar neighbor if its similarity is higher. Thus, each event maintains at most k most similar neighbors as the graph evolves.

Context-aware Representation. To represent related memories as compact puzzle pieces, we organize the neighbourhood of each event into a set of contextual groups. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ denote the event graph. For each event $v \in \mathcal{V}$, its neighbourhood is defined as

$$N(v) = \{u_i \in \mathcal{V} \mid (v, u_i) \in \mathcal{A}\}. \tag{1}$$

We organize $N(v)$ into a collection of puzzle pieces

$$\Pi_v = \{G_1, G_2, \ldots, G_k\}, \tag{2}$$

where the pieces are mutually disjoint and jointly cover $N(v)$. Each piece captures a compact contextual direction associated with the central event v, and the number of pieces k is determined adaptively from the local neighbourhood structure.

Let h_v and h_i denote the representations of the central event v and its neighbour u_i, respectively. Inspired by Bai et al. (2026), we form each puzzle piece relative to its local context rather than relying on the neighbour representations alone. Specifically, we condition each neighbour on the central event through the following context-aware representation:

$$z_i^{(v)} = [h_i; |h_v - h_i|; h_v \odot h_i], \tag{3}$$

where $[\cdot;\cdot]$ denotes concatenation and $\odot$ denotes element-wise multiplication. This representation characterizes both the content of a neighbouring event and its relation to the central event.

Adaptive Clustering. We then apply a Ward-style agglomerative (Jr., 1963) procedure to form the puzzle pieces. Starting from the singleton partition

$$\Pi_v^{(0)} = \big\{\{u_1\}, \ldots, \{u_n\}\big\}, \tag{4}$$

we repeatedly combine the pair of candidate pieces with the lowest merge cost. For a candidate piece G, its prototype is defined as

$$c_G^{(v)} = \frac{1}{|G|} \sum_{u_i \in G} z_i^{(v)}, \tag{5}$$

and the merge cost between two candidate pieces G_a and G_b is

$$\Delta_v(G_a, G_b) = \frac{|G_a||G_b|}{|G_a| + |G_b|} \left\| c_{G_a}^{(v)} - c_{G_b}^{(v)} \right\|_2^2. \tag{6}$$

Consequently, neighbouring events that play similar contextual roles with respect to v are preferentially assembled into the same puzzle piece.

To determine when the piece formation process should stop, let $\{\Delta_t\}|_{t=1}^{T}$ denote the sequence of merge costs, where $T = n - 1$. We measure the logarithmic increase between consecutive merge costs as

$$J_t = \log(\Delta_{t+1} + \epsilon) - \log(\Delta_t + \epsilon), \qquad t = 1, \ldots, T-1, \tag{7}$$

where ϵ ensures numerical stability. Let $J = \{J_t\}|_{t=1}^{T-1}$. We define the median absolute deviation (MAD) of the logarithmic increases as

$$\mathrm{MAD}(J) = \mathrm{median}_t \left| J_t - \mathrm{median}(J) \right|. \tag{8}$$

MAD provides a robust estimate of the dispersion of the merge-cost increases and is less sensitive to extreme jumps than the standard deviation. Based on this statistic, we define the stopping threshold as

$$\theta_J = \mathrm{median}(J) + \alpha \cdot \mathrm{MAD}(J), \tag{9}$$

where α controls the granularity of the resulting puzzle pieces. A smaller α detects merge-cost jumps more readily and therefore retains more fine-grained pieces, whereas a larger α allows more neighbouring events to be assembled into the same piece. The process stops before the first merge associated with a significant increase in cost:

$$t^\star = \min\{t \in 1, \ldots, T-1 \mid J_t > \theta_J\}, \qquad \Pi_v^\star = \Pi_v^{(t^\star)}. \tag{10}$$

If no significant increase is detected, the procedure continues until all neighbouring events are assembled into a single piece. The resulting collection $\Pi_v^\star$ provides a compact set of contextual puzzle pieces centered on event v, which can later be selected and combined according to the query.

3.3 Query-Guided Multi-Hop Routing (▷ Algorithm 3)

Piece Selection and Expansion. The event graph and its preformed puzzle pieces provide a shared memory structure from which each query consolidates a different memory puzzle. Starting from a query-relevant seed event, the router progressively selects pieces that contribute to the query. At routing step ℓ, let $\mathcal{F}^{(\ell)}$ denote the current frontier. For each event $v \in \mathcal{F}^{(\ell)}$, we retrieve its puzzle-piece collection $\Pi_v^\star$ and exclude previously visited events, producing

$$\mathcal{C}^{(\ell)} = \bigcup_{v \in \mathcal{F}^{(\ell)}} \Pi_v^\star = \{P_1^{(\ell)}, P_2^{(\ell)}, \ldots, P_{k_\ell}^{(\ell)}\}. \tag{11}$$

Each candidate piece contains a coherent event context that may contribute additional evidence for the query. An LLM-based gate evaluates each candidate conditioned on the query and its current branch

$$\left(g_j^{(\ell)}, c_j^{(\ell)}\right) = \mathrm{Gate}_{\mathrm{LLM}}\left(q, P_j^{(\ell)}, B_j^{(\ell)}\right), \tag{12}$$

where $g_j^{(\ell)} \in \{0,1\}$ indicates whether the piece is accepted and expanded, $c_j^{(\ell)} \in [0,1]$ is its confidence, and $B_j^{(\ell)}$ contains the pieces retained along the branch. Accepted pieces are ranked by confidence:

$$\mathcal{R}^{(\ell)} = \mathrm{Sort}_{c\downarrow}\{\left(P_j^{(\ell)}, c_j^{(\ell)}\right) \mid g_j^{(\ell)} = 1\}. \tag{13}$$

Their unvisited events form the frontier for step $\ell + 1$. This confidence-prioritized breadth-first routing continues until no suitable candidate remains or the maximum depth is reached. The retained pieces form the query-specific puzzle $\mathcal{P}_q$.

Evidence Integration. After routing, we trace each retained event to its source fragment and recover the associated structured facts. Let $\mathrm{Facts}(e)$ denote the facts extracted from the source fragment of event e. For each retained piece P,

$$\mathrm{Facts}(P) = \bigcup_{e \in P} \mathrm{Facts}(e). \tag{14}$$

The final evidence is organized as

$$\mathcal{E}_q = \{(P, \mathrm{Facts}(P)) \mid P \in \mathcal{P}_q\}. \tag{15}$$

The query and its assembled evidence are then provided to the LLM:

$$\hat{y} = \mathrm{LLM}\left(q, \mathcal{E}_q\right). \tag{16}$$

The retained pieces preserve the event-level reasoning structure, while their source facts provide the fine-grained evidence needed for a grounded response.

4 Experiments

In this section, we evaluate PIT to answer the following research questions (RQs):

- **RQ1**: Does our method outperform existing memory systems on multi-party memory tasks?
- **RQ2**: How much does each proposed component contribute to the overall performance?
- **RQ3**: How does participant grounding affect the discovery of relevant event evidence?
- **RQ4**: How do key hyperparameter settings affect the performance of PIT?

Table 2: Performance comparison on the EverMem benchmark.

Method	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
GPT-4.1-mini										
Full Context	*83.57*	*2.41*	*7.00*	*63.43*	*25.06*	*42.54*	*39.20*	*35.50*	*38.27*	*37.44* (-11.52)
+ MemoBase	60.09	12.85	18.00	64.68	36.77	30.60	17.05	29.59	38.78	34.27 (-14.69)
+ Mem0	55.40	11.24	6.33	66.17	52.46	51.87	22.73	31.36	36.22	37.09 (-11.87)
+ A-Mem	79.34	12.85	12.67	75.12	65.81	41.04	31.82	27.81	37.76	43.31 (-5.65)
+ MemOS	71.36	18.88	15.67	69.90	51.99	45.15	28.98	**32.54**	**48.47**	42.55 (-6.41)
+ REMem	48.83	**26.91**	19.33	54.48	39.11	47.76	**34.09**	14.79	18.37	33.74 (-15.22)
+ PIT	**81.22**	22.89	**21.67**	**79.10**	**69.09**	**63.06**	29.55	30.77	43.37	**48.96**
Qwen-3.5-Flash										
Full Context	*53.57*	*2.02*	*10.83*	*77.12*	*77.33*	*78.57*	*30.49*	*34.29*	*43.90*	*45.34* (-8.32)
+ MemoBase	**84.98**	11.24	4.67	71.14	86.42	66.79	30.68	49.11	51.53	50.73 (-2.93)
+ Mem0	66.20	1.61	1.33	75.87	84.78	69.03	30.68	44.38	46.94	46.76 (-6.90)
+ A-Mem	78.40	10.04	3.00	71.64	86.18	71.64	**34.09**	47.93	46.94	49.98 (-3.68)
+ MemOS	69.01	10.84	5.67	78.11	84.78	**72.76**	22.73	**50.30**	56.12	50.03 (-3.36)
+ REMem	71.83	8.03	**13.67**	66.17	47.07	50.37	28.98	31.95	29.59	38.62 (-15.04)
+ PIT	81.69	**17.67**	8.67	**78.36**	**90.40**	72.01	27.27	**50.30**	**56.63**	**53.66**
GLM-4										
Full Context	*Context Length Exceeded*									
+ MemoBase	80.85	17.27	6.67	69.90	73.54	38.43	41.48	50.62	52.18	47.35 (-10.40)
+ Mem0	70.42	3.21	4.33	74.63	76.58	44.78	**42.61**	48.52	50.00	45.64 (-12.11)
+ A-Mem	79.81	13.65	7.33	75.62	76.81	42.91	35.23	49.70	53.06	47.63 (-10.12)
+ MemOS	80.85	15.26	9.00	69.40	78.92	44.85	36.36	50.62	53.06	48.16 (-9.59)
+ REMem	71.83	8.03	13.67	66.17	47.07	50.37	28.98	31.95	29.59	38.62 (-19.13)
+ PIT	**85.92**	**22.49**	**36.00**	**85.07**	**81.26**	**66.79**	20.45	**58.58**	**63.27**	**57.75**

4.1 EXPERIMENTAL SETUP

Datasets & Evaluation Metrics. We evaluated our method on the only currently officially published multi-party benchmark. EverMemBench (Hu et al., 2026) serves as the primary benchmark for agent memory, featuring multi-party and multi-group conversations with 2,400 QA pairs covering Fine-Grained Recall, Memory Awareness, and Profile Understanding. Moreover, we use Accuracy (ACC) as the primary evaluation metric across all task dimensions.

Baselines. We compare the model with several well-established baselines, including MemoBase (Team, 2025), Mem0 (Chhikara et al., 2025), A-Mem (Xu et al., 2025), MemOS (Li et al., 2025), REMem (Shu et al., 2026). Detailed introduction is listed in Appendix C.1.

4.2 MAIN RESULTS (RQ1)

In Table 2, we report the metrics for various sub-tasks and the overall average performance of PIT on EverMemBench, evaluated across three mainstream LLM settings. The results shows that PIT achieves the best overall average performance, outperforming the strongest competing memory systems, and demonstrates distinct preferences across specific sub-problems with varying focuses.

The gains are particularly evident in multi-event recall and memory awareness. Participant grounding helps distinguish topically similar events involving different participants, while contextual consolidation and query-guided multi-hop routing connect relevant evidence across conversations. This is especially beneficial for consistency, proactivity, and memory update tasks, which require tracking the correct participants and their changing states. PIT also substantially outperforms *Full Context*, demonstrating the effectiveness of assembling query-specific evidence rather than directly using the entire history. However, the gains are less consistent for style understanding. This task depends on recurring linguistic patterns in the original utterances, which may be weakened when dialogues

are compressed into event-centric representations. This suggests that PIT preserves factual and relational information more effectively than fine-grained stylistic cues.

4.3 ABLATION STUDIES (RQ2)

As shown in Table 3, we conduct ablations by removing Bipartite Graph, Event Consolidation, and Multi-Hop Router individually while keeping the remaining components unchanged across all three backbone models. Detailed settings are provided in Appendix D.

The ablations show the same contribution order across all three backbones: The larger effect of Event Consolidation suggests that organizing related events into compact memory pieces is central to effective retrieval. Participant grounding and multi-hop routing further help distinguish participant-specific information and connect relevant pieces for each query. Together, these results show that the three components provide complementary benefits to multi-party conversational memory. For detailed data on different types of question under GPT-4.1-mini, Qwen-3.5-Flash and GLM-4, please refer to Tables 7, 8 and 9.

Table 3: Average performance of different ablation settings.

	GPT	Qwen	GLM
PIT	**48.97**	**53.67**	**57.76**
w/o Bipartite Graph	47.11	52.35	54.32
w/o Event Consolidation	43.12	51.80	53.32
w/o Multi-Hop Router	47.32	53.34	55.75

4.4 QUALITY OF PARTICIPANT-GROUNDED EVENT REPRESENTATION (RQ3)

Does Participant Grounding Improve Event Representation? In this part, we compare gold-evidence event embeddings with and without participant grounding across 2,400 queries under three backbone models. Evidence concentration is measured by normalized dispersion R, with improvements summarized by its reduction ΔR, mean relative improvement I, and win rate WR. Detailed settings are provided in Appendix E.1.

Table 4 shows that participant grounding consistently reduces evidence dispersion across all three backbones. The improvement holds for more than 99.5% of queries, although its magnitude varies across models. This broad consistency indicates that the gain is not tied to a particular backbone: incorporating participant information generally brings the evidence required by the same query closer in the representation space.

Table 4: Overall performance of participant grounding on evidence cohesion.

	GPT	Qwen	GLM
R_{PIT}	0.6523	0.6559	0.6524
$R_{\text{w/o}}$	0.8689	0.9360	0.7255
ΔR	24.92%	29.92%	10.07%
I	24.53%	29.53%	10.29%
WR	99.83%	99.58%	99.54%

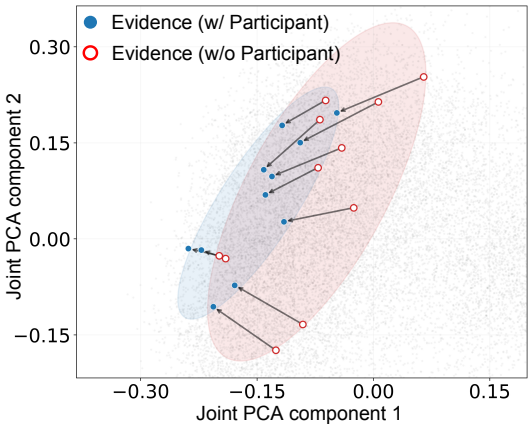

Figure 3: A case of gold-evidence cohesion with vs. without participant grounding.

Figure 3 provides a representative case. The hollow red and solid blue markers denote the same gold-evidence events before and after participant grounding, respectively, while each line traces the corresponding shift. The original event embeddings are relatively dispersed, whereas the grounded embeddings move toward a more compact region. This example illustrates how participant context aligns related evidence that may appear semantically scattered when represented only by event content.

Does Participant Grounding Improve Event Retrieval Performance? In this part, we compare the Top-K events retrieved with and without participant grounding across three backbone models. Retrieval is evaluated in terms of evidence coverage (*Any Hit@5*), ranking quality (*MRR*), source precision (*Source Precision@5*), success rate (*SR*) and its contradiction rate (*SR**). Detailed settings are provided in Appendix E.2.

Table 5: Participant-aware retrieval and evidence support evaluation across GPT-4.1-mini, Qwen-3.5-Flash, and GLM-4.

Model	Setting	Reference Message Hit Evaluation			Evidence Support Evaluation	
		AnyHit@5 ↑	*MRR* ↑	*Source Precision@5* ↑	*SR* ↑	*CR** ↑
GPT	w/ Participant	**70.54%**	**0.5899**	**39.70%**	**30.58%**	**0.42%**
	w/o Participant	69.21%	0.5854	38.98%	28.92%	0.29%
Qwen	w/ Participant	**69.75%**	0.5840	**39.05%**	24.42%	**0.50%**
	w/o Participant	69.21%	**0.5854**	38.98%	**25.79%**	0.37%
GLM	w/ Participant	**71.04%**	**0.6004**	39.69%	**28.04%**	**0.42%**
	w/o Participant	70.08%	0.5991	**39.73%**	28.00%	0.37%

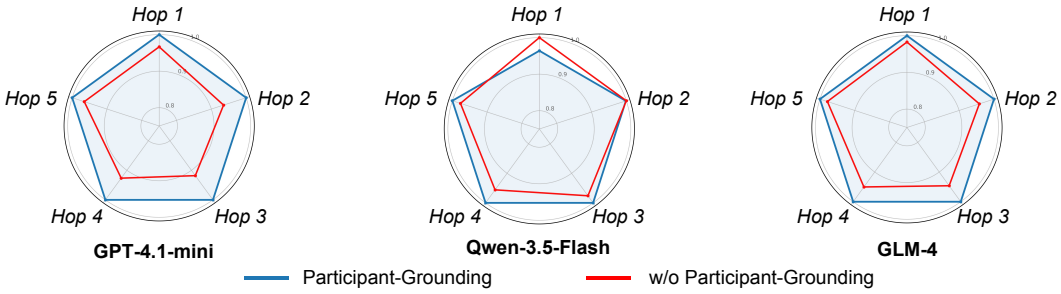

Figure 4: Gold-evidence coverage across routing hops with vs. w/o participant grounding across three backbone models.

As shown in Table 5, participant grounding consistently improves *Any Hit@5* and *CR** across all three backbones, indicating better relevant-evidence coverage and fewer cross-participant conflicts. Changes in *MRR* and *Source Precision@5* are small and model-dependent, suggesting that the main benefit lies in evidence coverage and consistency rather than ranking. Similarly, *SR* improves for GPT-4.1-mini, remains nearly unchanged for GLM-4, and decreases for Qwen-3.5-Flash, showing that retrieval gains do not translate uniformly into final-response quality.

4.5 Reliability of Participant-Grounded Multi-hop Retrieval (RQ3)

How Effectively Does the Router Identify Query-Specific Seed Events? In this part, we evaluate whether participant grounding helps the Puzzle Router reach relevant evidence within a limited traversal budget. Starting from the Top-K seed events, we measure the coverage of gold evidence after one to five hops, comparing PIT with its without participant counterpart across three backbones. Detailed settings are provided in Appendix E.3.

Figure 4 compares gold-evidence coverage across routing depths. For GPT-4.1-mini and GLM-4, the participant-grounded profile lies outside the participant-free profile at every hop. For Qwen-3.5-Flash, the two profiles are similar at the first two hops, while participant grounding provides clearer gains from the third hop onward. These results suggest that participant grounding helps the Router reach more distant relevant evidence within a bounded traversal budget. Detailed results are provided in Tables 10, 11, and 12.

How Does Participant Grounding Affect End-to-End Evidence Discovery? Figure 5 compares retrieval with and without participant grounding for a query about the cause of a pagination style misalignment. With participant grounding, the router follows the relevant chain from regression testing through Siyuan Qiu's request to Ming Gu's diagnosis, correctly recovering "CSS specificity."

Without participant grounding, the router either stops at a status-only summary with no causal evidence or drifts to a similar pagination bug handled by Chenguang Liu and Fang Wang, producing the incorrect answer "front-end state update issue." This case shows that participant information helps distinguish similar events and preserve the correct event chain during multi-hop retrieval.

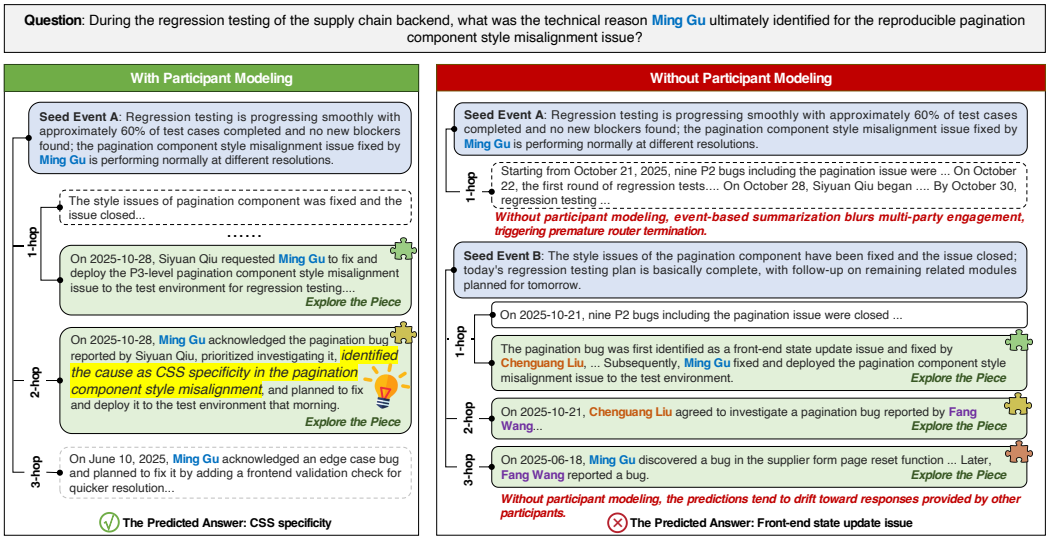

Figure 5: Participant modeling guides retrieval along Ming Gu's event chain to identify CSS specificity as the correct cause (left). Without it, retrieval stops at status-only evidence (top right) or drifts to another participant's pagination issue (bottom right). Colors mark seed, DIRECT, STOP, and gold-evidence events.

4.6 Sensitive Experiments (RQ4)

In this part, we examine the sensitivity of PIT to two design choices: the hop depths used for event expansion and the consolidation coefficient α in Piece Formation. Each setting is evaluated across three backbone models while the remaining components are fixed. Detailed settings are provided in Appendix F.

As shown in Table 6, combining one-hop and two-hop expansion consistently performs better than using either depth alone and remains close to the full configuration. This suggests that direct and more distant evidence are complementary for multi-event reasoning.

Regarding the consolidation coefficient α, an α value of 5 yielded the best overall performance across all tests. In contrast, $\alpha = 10$ consistently resulted in a significant performance drop; excessively high weights caused the consolidation of numerous events that should not have been merged, thereby distorting the Piece Formation process. Please refer to the Tables 13, 14 and 15 for detailed data from all experiments.

Table 6: Average performance of different sensitive settings.

	GPT	Qwen	GLM
PIT	48.97	53.67	57.76
Number of hops			
hop = 1	44.39 (-4.58)	49.77 (-3.90)	53.73 (-4.03)
hop = 2	46.58 (-2.39)	51.60 (-2.07)	52.35 (-5.41)
hop = 1, 2	48.17 (-0.80)	52.50 (-1.17)	57.18 (-0.58)
Consolidation Coefficient α			
$\alpha = 0$	47.32 (-1.65)	53.34 (-0.33)	55.75 (-2.01)
$\alpha = 1$	45.99 (-2.98)	52.70 (-0.97)	55.62 (-2.14)
$\alpha = 5$	49.10 (+0.13)	52.84 (-0.83)	57.47 (-0.29)
$\alpha = 10$	42.61 (-6.36)	48.57 (-5.10)	51.78 (-5.98)

5 Conclusion

We study agent memory in multi-party dialogues, where relevant information is fragmented across participants, groups, and time. We introduce PIT, a participant-grounded framework that treats retrieval as query-specific puzzle completion. It enriches event representations with participant context, consolidates related events into memory pieces, and connects them through query-guided multi-hop routing. On EverMemBench, PIT outperforms strong baselines across backbone LLMs by 6.71%-19.91%. Further analyses show that participant grounding improves representation quality and retrieval reliability, increasing multi-hop coverage while reducing premature stopping and content drift in responses. These results highlight the importance of modeling participant relations and reconstructing query-specific context beyond semantic similarity alone.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring mandatory disclosure. We used generative AI tools only to improve the readability of the manuscript through language polishing and paragraph restructuring. All AI-assisted revisions were carefully reviewed and further edited by the authors to ensure that they accurately preserved the intended technical meaning. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: learning knowledge graph world models with episodic memory for llm agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

Xu Bai, Bin Lu, Kun Zhang, Shengbo Chen, Xinbing Wang, Chenghu Zhou, and Meng Jin. Rethinking efficient graph coarsening via a non-selfishness principle. In *Forty-third International Conference on Machine Learning*, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Niv Eckhaus, Uri Berger, and Gabriel Stanovsky. Time to talk: Llm agents for asynchronous group communication in mafia games. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 11356–11368, 2025.

Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. Mpc-bert: A pre-trained language model for multi-party conversation understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 3682–3692, 2021.

Jia-Chen Gu, Chao-Hong Tan, Chongyang Tao, Zhen-Hua Ling, Huang Hu, Xiubo Geng, and Daxin Jiang. Hetermpc: A heterogeneous graph neural network for response generation in multi-party conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 5086–5097, 2022.

Jia-Chen Gu, Zhenhua Ling, Quan Liu, Cong Liu, and Guoping Hu. Gift: Graph-induced fine-tuning for multi-party conversation understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 11645–11658, 2023.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, pp. 59532–59569, 2024.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, 2020.

Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Yi Bai, Dannong Xu, Tianwei Lin, Xiaohong Li, Yunyun Han, Jian Pei, and Yafeng Deng. Evaluating long-horizon memory for multi-party collaborative dialogues. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 9082–9090, 2026.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. Gsn: A graph-structured network for multi-party dialogues, 2019. URL https://arxiv.org/abs/1905.13637.

Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and Rui Yan. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 1909–1919, 2019.

Yiyang Li and Hai Zhao. Em pre-training for multi-party dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 92–103, 2023.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. Memos: An operating system for memory-augmented generation (mag) in large language models, 2025. URL https://arxiv.org/abs/2505.22101.

Zhengxuan Lu, Dongfang Li, Yukun Shi, Beilun Wang, Longyue Wang, and Baotian Hu. Structured episodic event memory. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 6125–6141, 2026.

Zhao Meng, Lili Mou, and Zhi Jin. Towards neural speaker modeling in multi-party conversation: the task, dataset, and models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

Hiroki Ouchi and Yuta Tsuboi. Addressee and response selection for multi-party conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2133–2143, 2016.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Yiheng Shu, Padmaja Jonnalagedda, Xiang Gao, Bernal Jimenez Gutierrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. Remem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, pp. 55958–55987, 2026.

Haoran Sun and Shaoning Zeng. Hierarchical memory for high-efficiency long-term reasoning in llm agents, 2025. URL https://arxiv.org/abs/2507.22925.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 8416–8439, 2025.

Memodb Team. Memobase: User profile-based long-term memory for ai chatbot applications, 2025.

Weishi Wang, Steven C.H. Hoi, and Shafiq Joty. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6581–6591, 2020.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems*, pp. 17577–17604, 2025.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, and Yang Yang. Lightweight llm agent memory with small language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 12914–12929, 2026.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir Radev. Addressee and response selection in multi-party conversations with speaker interaction rnns. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. 2025a. URL https://doi.org/10.48550/arXiv.2506.05176.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, Lei Hou, and Juanzi Li. Simulating classroom education with llm-empowered agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 10364–10379, 2025b.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

A SUPPLEMENTARY PROMPT TEMPLATE

The Prompt Template for Event Extract

Extract non-redundant memory events from the provided group-dialogue messages.
Return a JSON object only, with one top-level key "events". Each event must contain:
- "event_description" (str): A concise, self-contained description of a useful event, state, decision, plan, request, question, preference, risk, or outcome.
- "participants" (list[str]): People explicitly involved in the event. Use [] if none are identifiable.
- "memory_tags" (list[str]): 1–5 short normalized tags describing the event type, status, or subject.
- "point_in_time" (str or null): The event time, if explicitly stated or resolvable from the message time.
- "message_index" (list[str]): message_index value(s) that directly support the event.
Rules:
1. Extract only information explicitly supported by the messages; do not infer unstated facts.
2. Include information useful for retrieval, profiling, task tracking, graph construction, or question answering.
3. Split distinct events and merge duplicate or near-duplicate events.
4. Ignore greetings, thanks, acknowledgements, emojis, filler, and repetitions without new information.
5. Resolve relative time expressions from the corresponding message time; otherwise use null.
6. Return valid JSON only, with no comments, explanations, or extra keys.
Output format:

```
{
  "events": [
    {
      "event_description": "...",
      "participants": ["...", "..."],
      "memory_tags": ["...", "..."],
      "point_in_time": null,
      "message_index": ["..."]
    }
  ]
}
```

Answer:

The Prompt Template for Fact Extract

Extract non-redundant atomic fact triples from messages and their associated events in a group dialogue.
Return a JSON object with one top-level key "facts". Each fact must contain exactly:
- "subject" (str): The entity or concept the fact is about. Do not default to the speaker.
- "predicate" (str): A concise, normalized semantic relation.
- "object" (str): A concise entity, value, status, requirement, target, or result—not a full sentence.
- "message_index" (list[str]): message_index value(s) that directly support the fact.
- "event_id" (list[str]): Stable event ID value(s) whose events semantically correspond to the fact. Use [] when the fact does not correspond to any supplied event.
Rules:
1. Extract all explicit, text-supported facts.
2. Capture underlying relationships, not speech acts such as SAID, ASKED, or SUGGESTED.
3. Use a person as the subject only when the fact concerns their role, responsibility, preference, availability, action, or ownership.
4. Each triple must represent one atomic fact. Split compound facts, merge duplicates, and combine their supporting event IDs.
5. Copy only event_id values supplied in the user prompt. Never invent an ID.
6. Add an event_id only when that event expresses or directly supports the same fact. Sharing a message alone does not prove correspondence.
7. Keep all fields concise and reusable for graph construction. Exclude trivial, irrelevant, or unsupported information.
8. Return valid JSON only, with no extra keys, comments, or explanations.
Output format:

```
{
    "facts":[
        {
            "subject":"...",
```

```
            "predicate":"...",
            "object":"...",
            "message_index":["..."],
            "event_id":["..."]
        }
    ]
}
```

Answer:

The Prompt Template for Event Set Summary

You are a precise event memory summarizer. Return only one valid JSON object.
Please summarize the given events, taking into account their underlying logic and chronological relationships. Do not include markdown, code fences, explanations, or extra text.
Answer:

The Prompt Template for Puzzle Router

Select query-relevant event sets from a summarized event tree.
For each request, you are given one query, one seed event, the current parent event set, and candidate child event sets.

Classify every candidate as exactly one of:
- DIRECT: useful evidence/context for answering the query and worth expanding.
- STOP: irrelevant, redundant, too weak, or not worth expanding.

Return valid JSON only. Return only candidates labeled DIRECT in results. Do not include STOP candidates. Any candidate omitted from results will be treated as STOP. Include a confidence number from 0 to 1 for every returned DIRECT candidate.
Answer:

The prompt template for Memory-Augmented Answer Prompt, Full-Context Answer Prompt and LLM-as-a-Judge can be found in EverMembench (Hu et al., 2026), we will not elaborate further here.

B ALGORITHM

Algorithm 1 Participant-Grounded Event Encoding

Require: Dialogue stream $\mathcal{X}$, session length L, neighbor cap k
Ensure: Events $\mathcal{E}$, fact triples $\mathcal{T}$, enhanced event representations $\widehat{\mathbf{H}}_E$, and event graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$

$\{\mathcal{S}_1, \ldots, \mathcal{S}_N\} \leftarrow \textsc{SegmentDialogue}(\mathcal{X}, L)$
$(\mathcal{E}, \mathcal{T}) \leftarrow \textsc{ExtractClues}(\{\mathcal{S}_i\}_{i=1}^{N})$ ▷ events and provenance-linked facts
for $e_i \in \mathcal{E}$ **do**
  $\mathbf{h}_i \leftarrow \textsc{EncodeEvent}(e_i)$
end for
$\mathbf{H}_E \leftarrow \textsc{Stack}(\{\mathbf{h}_i : e_i \in \mathcal{E}\})$
$\mathcal{G}_{EP} \leftarrow \textsc{BuildEventParticipantGraph}(\mathcal{E})$
$\widehat{\mathbf{H}}_E \leftarrow \textsc{LightGCN}(\mathcal{G}_{EP}, \mathbf{H}_E)$ ▷ participant-grounded event representations
$\mathcal{V} \leftarrow \mathcal{E}$; $\mathcal{A} \leftarrow \varnothing$
for each event e_i in chronological order **do**
  $\mathcal{N}(e_i) \leftarrow \textsc{TopK}(\{(e_j, \cos(\widehat{\mathbf{h}}_i, \widehat{\mathbf{h}}_j)) : j < i\}, k)$
  for $e_j \in \mathcal{N}(e_i)$ **do**
    $\mathcal{N}(e_j) \leftarrow \textsc{UpdateTopK}(\mathcal{N}(e_j), e_i, k)$
  end for
end for
$\mathcal{A} \leftarrow \{(e_i, e_j) : e_j \in \mathcal{N}(e_i)\}$
return $(\mathcal{E}, \mathcal{T}, \widehat{\mathbf{H}}_E, \mathcal{G})$

Algorithm 2 Contextual Event Consolidation

Require: Event graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, enhanced event representations $\widehat{\mathbf{H}}_E$, coefficient α
Ensure: Contextual puzzle pieces $\{\Pi_v^\star : v \in \mathcal{V}\}$

Fix a numerical constant $\epsilon > 0$
for $v \in \mathcal{V}$ **do**
  $\mathcal{N}(v) \leftarrow \{u_i \in \mathcal{V} : (v, u_i) \in \mathcal{A}\}$
  $n \leftarrow |\mathcal{N}(v)|$
  if $n = 0$ **then**
    $\Pi_v^\star \leftarrow \varnothing$
  else if $n \le 2$ **then**
    $\Pi_v^\star \leftarrow \{\mathcal{N}(v)\}$
  else
    for $u_i \in \mathcal{N}(v)$ **do**
      $\mathbf{z}_i^{(v)} \leftarrow [\widehat{\mathbf{h}}_i; |\widehat{\mathbf{h}}_v - \widehat{\mathbf{h}}_i|; \widehat{\mathbf{h}}_v \odot \widehat{\mathbf{h}}_i]$
    end for
    $(\Pi_v^{(0)}, \ldots, \Pi_v^{(n-1)}; \Delta_1, \ldots, \Delta_{n-1}) \leftarrow \textsc{WardMergeSequence}(\{\mathbf{z}_i^{(v)}\}_{i=1}^{n})$ ▷ using the Ward cost Δ_v in Eq. (6)
    for $t = 1$ to $n - 2$ **do**
      $J_t \leftarrow \log(\Delta_{t+1} + \epsilon) - \log(\Delta_t + \epsilon)$
    end for
    $\theta_J \leftarrow \mathrm{median}(\mathbf{J}) + \alpha\, \mathrm{MAD}(\mathbf{J})$
    if $\exists t$ such that $J_t > \theta_J$ **then**
      $t^\star \leftarrow \min\{t : J_t > \theta_J\}$
      $\Pi_v^\star \leftarrow \Pi_v^{(t^\star)}$
    else
      $\Pi_v^\star \leftarrow \Pi_v^{(n-1)}$
    end if
  end if
end for
return $\{\Pi_v^\star : v \in \mathcal{V}\}$

Algorithm 3 Query-Guided Multi-Hop Routing

Require: Query q, events $\mathcal{E}$, fact mapping $\mathrm{Facts}(\cdot)$, puzzle pieces $\{\Pi_v^\star\}$, maximum hop H
Ensure: Reconstructed evidence $\mathcal{E}_q$ and answer $\widehat{y}$

$\mathcal{F}^{(0)} \leftarrow \textsc{RetrieveSeedEvents}(q, \mathcal{E})$
$\mathcal{V}_{\text{seen}} \leftarrow \mathcal{F}^{(0)}$; $\mathcal{P}_q \leftarrow \varnothing$
for $\ell = 0$ to $H-1$ **do**
  $\mathcal{C}^{(\ell)} \leftarrow \bigcup_{v \in \mathcal{F}^{(\ell)}} \textsc{RemoveVisited}(\Pi_v^\star, \mathcal{V}_{\text{seen}})$
  if $\mathcal{C}^{(\ell)} = \varnothing$ **then**
    break
  end if
  for $P_j^{(\ell)} \in \mathcal{C}^{(\ell)}$ **do**
    $(g_j^{(\ell)}, c_j^{(\ell)}) \leftarrow \textsc{GateLLM}(q, P_j^{(\ell)}, \mathcal{B}_j^{(\ell)})$ ▷ $\mathcal{B}_j^{(\ell)}$ contains pieces retained on its branch
  end for
  $\mathcal{R}^{(\ell)} \leftarrow \textsc{SortByConfidence}(\{(P_j^{(\ell)}, c_j^{(\ell)}) : g_j^{(\ell)} = 1\})$
  if $\mathcal{R}^{(\ell)} = \varnothing$ **then**
    break
  end if
  $\mathcal{P}_q \leftarrow \mathcal{P}_q \cup \{P : (P, c) \in \mathcal{R}^{(\ell)}\}$
  $\mathcal{F}^{(\ell+1)} \leftarrow \bigcup_{(P,c) \in \mathcal{R}^{(\ell)}} P \setminus \mathcal{V}_{\text{seen}}$
  $\mathcal{V}_{\text{seen}} \leftarrow \mathcal{V}_{\text{seen}} \cup \mathcal{F}^{(\ell+1)}$
end for
$\mathcal{E}_q \leftarrow \{(P, \bigcup_{e \in P} \mathrm{Facts}(e)) : P \in \mathcal{P}_q\}$
$\widehat{y} \leftarrow \mathrm{LLM}(q, \mathcal{E}_q)$
return $(\mathcal{E}_q, \widehat{y})$

C Experiments Settings

C.1 Baselines Settings

We evaluate five representative memory systems, including MemoBase, Mem0, A-Mem, MemOS, and REMem. Following their default configurations, MemoBase provides up to 3K tokens of profile-based memories, A-Mem, Mem0, REMem retrieve the top-10 memories, MemOS retrieves the top-20 memories. All-MiniLM-L6-v2 is used for the text embedding models in all baselines. For comparison, the *Full Context* baseline directly provides the complete dialogue history (1M tokens) without any memory module.

C.2 Implementation Details

All algorithms are run on a Linux server running Ubuntu 20.04 LTS. Additionally, All methods are conducted on a GPU (NVIDIA GeForce RTX 4090, 24GB memory) using CUDA 12.6.

D Experiment Settings of Ablation Studies

w/o Bipartite Graph. We remove participant-aware augmentation by discarding the event–participant bipartite graph and LightGCN propagation. Each event is represented solely by its original textual embedding $\mathbf{e}_i$, which replaces the enhanced embedding $\hat{\mathbf{e}}_i$ in all downstream operations.

w/o Event Consolidation. We remove the LLM-based gating mechanism and randomly select neighboring events during BFS traversal, while keeping the remaining retrieval procedure unchanged.

Table 7: Ablation study results under GPT-4.1-mini

GPT-4.1-mini	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
PIT	81.22	22.89	21.67	79.10	69.09	63.06	29.55	30.77	43.37	48.97
w/o Bipartite Graph	82.63	18.88	20.67	75.12	69.09	57.09	28.68	30.91	40.94	47.11 (-1.86)
w/o Event Consolidation	48.83	12.85	13.67	78.61	65.34	61.57	33.52	38.46	35.20	43.12 (-5.85)
w/o Multi-Hop Router	84.51	20.08	25.00	75.36	70.26	60.30	25.68	23.73	40.92	47.32 (-1.65)

Table 8: Ablation study results under Qwen-3.5-Flash.

Method	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
PIT	81.69	17.67	8.67	78.36	90.40	72.01	27.27	50.30	56.63	53.67
w/o Bipartite Graph	85.38	12.56	7.08	76.78	89.47	75.93	30.14	46.38	47.47	52.35 (-1.32)
w/o Event Consolidation	71.69	12.88	9.00	74.08	91.57	70.90	31.25	50.25	54.59	51.80 (-1.87)
w/o Multi-Hop Router	80.98	18.88	11.00	82.09	90.87	70.24	26.55	51.48	47.93	53.34 (-0.33)

w/o Multi-Hop Router. We remove the proposed contextual consolidation mechanism and directly retrieve the top-k most similar events at each hop. We evaluate this variant under both one-hop and three-hop expansion to examine the effect of structural compression during multi-hop retrieval.

Table 9: Ablation study results under GLM-4.

Method	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
PIT	85.92	22.49	36.00	85.07	81.26	66.79	20.45	58.58	63.27	57.76
w/o Bipartite Graph	86.93	20.18	34.00	83.44	80.45	60.83	15.67	56.00	51.38	54.32 (-3.44)
w/o Event Consolidation	70.46	14.56	35.33	77.89	74.56	60.83	25.63	60.34	60.26	53.32 (-4.44)
w/o Multi-Hop Router	83.56	21.53	35.33	81.07	82.04	63.45	18.45	53.83	62.46	55.75 (-2.01)

E Supplementary Material for Quality of Participant-Grounded Event Representation Analysis

E.1 Experimental Setting and Results for Evaluating Event Representation

Compared Representations. We compare event representations under two conditions. The participant-grounded condition uses the event embeddings produced by participant-event LightGCN propagation, whereas the participant-free condition uses the event-only embeddings from the separately generated baseline pipeline.

Normalized Compactness. All embeddings are first ℓ_2-normalized. Let G_q be the m_q aligned gold-evidence events for query q. Their dispersion under condition $c \in \{\text{without}, \text{with}\}$ is

$$D_q^{(c)} = \frac{2}{m_q(m_q-1)} \sum_{i<j,\, i,j \in G_q} \left(1 - \cos\left(\mathbf{e}_i^{(c)}, \mathbf{e}_j^{(c)}\right)\right).$$

To control for differences in the global scale of the two embedding spaces, we sample 200,000 random aligned event pairs per topic using seed 42. The same pairs are used in both conditions. If $B_t^{(c)}$ is the mean cosine distance of these background pairs for topic t, the normalized compactness is

$$R_q^{(c)} = \frac{D_q^{(c)}}{B_{t(q)}^{(c)}}.$$

Lower values indicate more compact gold evidence relative to the corresponding topic-level background. Thus, R measures the compactness of gold evidence while controlling for the overall scale of each embedding space.

Aggregate Measures. We report the mean $R_q^{(c)}$ over all queries. This gives the overall normalized compactness under each condition, with lower values indicating better concentration. We additionally compute the query-level relative improvement

$$I_q = \frac{R_q^{(\text{without})} - R_q^{(\text{with})}}{R_q^{(\text{without})}},$$

and report $I = |Q|^{-1} \sum_{q \in Q} I_q$. A positive I_q favors the participant-aware representation, and its magnitude measures the proportional reduction in normalized dispersion. Finally, the win rate is

$$Win\ Rate = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}\left[R_q^{(\text{with})} < R_q^{(\text{without})}\right].$$

It measures the fraction of queries improved by participant information and therefore complements I, which measures the average improvement magnitude.

Joint-PCA Visualization and Scope. For the qualitative case study, the figure compares the initial event embeddings with the participant-grounded LightGCN embeddings from the participant-aware pipeline. Both sets are ℓ_2-normalized and concatenated with their background and gold-evidence events before fitting a single PCA model. Both sets are therefore projected onto the same two axes, and each arrow connects the same event across the two representations. The two-dimensional projection is used only for qualitative illustration; all reported distances and aggregate measures are computed in the original high-dimensional spaces. The aggregate experiment compares outputs from a separately generated event-only pipeline and the participant-event LightGCN pipeline. It is therefore a system-level comparison rather than a same-initial-embedding, propagation-only causal ablation.

E.2 Experimental Setting and Results for Retrieval Performance Analysis

Reference Message Hit Evaluation. This experiment measures whether the source messages of a retrieved seed event overlap with the annotated reference messages. For query q, let R_q denote the reference message set and $M(s_{q,i})$ the source message set of the i-th retrieved event. A source hit is defined as

$$h_{q,i} = \mathbb{I}\left(M(s_{q,i}) \cap R_q \neq \varnothing\right).$$

We report three metrics:

Any Hit@5: the proportion of queries with at least one source hit among the top-5 events,

$$\textit{Any Hit@5} = \frac{1}{N} \sum_{q=1}^{N} \mathbb{I}\left(\sum_{i=1}^{5} h_{q,i} > 0\right).$$

Mean Reciprocal Rank (MRR): the average reciprocal rank of the first source hit, with a value of zero when no hit occurs in the top five,

$$\textit{MRR} = \frac{1}{N} \sum_{q=1}^{N} \frac{1}{r_q}.$$

Source Precision@5: the proportion of retrieved events whose source messages overlap with the reference messages,

$$\textit{Source Precision@5} = \frac{\sum_{q=1}^{N} \sum_{i=1}^{5} h_{q,i}}{5N}.$$

These metrics capture source coverage, ranking quality, and evidence density, respectively.

Table 10: Gold-evidence coverage across routing hops with vs. w/o participant grounding using GPT-4.1-mini.

Max. Hops	PIT	w/o Participant	Δ
1	**22.74%**	21.98%	+0.76 pp
2	**40.78%**	38.13%	+2.65 pp
3	**58.96%**	54.13%	+4.83 pp
4	**77.71%**	72.00%	+5.71 pp
5	**89.55%**	86.44%	+3.10 pp

Table 11: Gold-evidence coverage across routing hops with vs. w/o participant grounding using Qwen-3.5-Flash.

Max. Hops	PIT	w/o Participant	Δ
1	21.19%	**21.98%**	−0.80 pp
2	38.08%	**38.13%**	−0.05 pp
3	**55.46%**	54.13%	+1.33 pp
4	**75.32%**	72.00%	+3.32 pp
5	**88.50%**	86.44%	+2.06 pp

Answer Evidence Support Evaluation. For each query, we jointly evaluate the top-5 seed events and their deduplicated source messages. The evaluator receives the query, gold answer, event descriptions, and source messages, including message IDs, speakers, timestamps, and original dialogue. It must rely exclusively on the provided evidence.

Each evidence package is assigned one of two grades: (1) $Support$: The evidence supports the complete gold answer; (2) $Contradiction$: The evidence is irrelevant or provides no meaningful support. A separate contradiction flag records whether any key evidence explicitly conflicts with the gold answer. The metrics of Support Rate and Contradiction Rate can be formulated as:

$$\textit{SR/CR} = \frac{1}{N}\sum_{q=1}^{N}\mathbb{I}(Support/Contradiction).$$

For the *Any Hit@5*, *MRR*, *Source Precision@5*, and *SR* indicators, higher values are better, whereas for *CR*, lower values are better; to standardize the directionality, we set $\textit{CR}^{*}$ = 1-*CR*.

Table 12: Gold-evidence coverage across routing hops with vs. w/o participant grounding using GLM-4.

Max. Hops	PIT	w/o Participant	Δ
1	**22.42%**	22.03%	+0.39 pp
2	**39.99%**	38.31%	+1.68 pp
3	**57.04%**	53.97%	+3.07 pp
4	**74.74%**	71.02%	+3.72 pp
5	**87.38%**	85.52%	+1.86 pp

E.3 Experimental Setting and Results for Identifying Query-Specific Seed Events Analysis

We evaluate whether the event-level memory graph enables retrieved seed events to reach the evidence events required for answering a query. This experiment isolates the structural quality of the event graph from the query-conditioned decisions of the Puzzle Router.

For each model and topic, we use the event-level k-NN graph $G = (V, A)$ defined in Section 3.1.2, where each node represents an extracted event and $k = 5$. For structural evaluation, we convert it into an undirected, unweighted graph $\bar{G} = (V, \bar{A})$, where an edge $\{u, v\}$ is added if either $(u, v) \in A$ or $(v, u) \in A$. Graph distance is measured by the shortest-path length in $\bar{G}$.

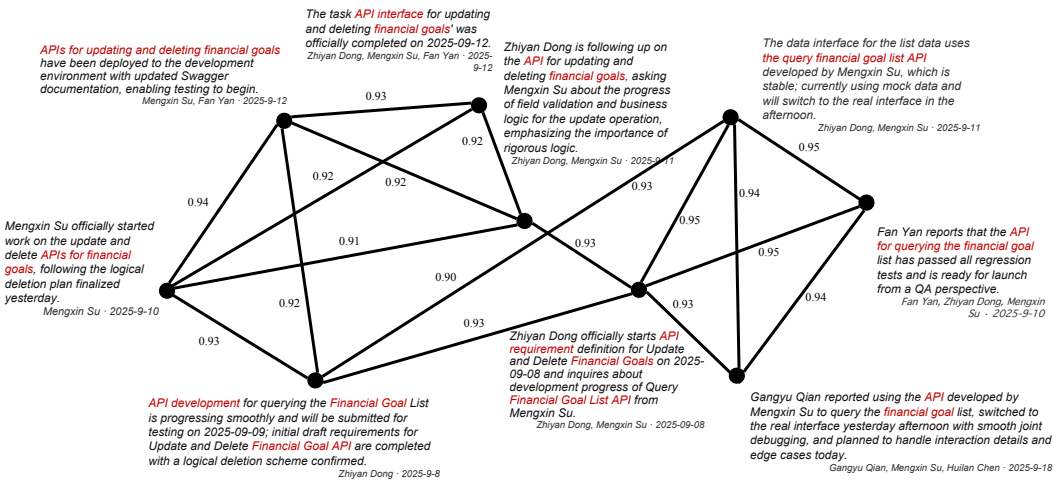

Figure 6: **A representative local subgraph centered on financial-goal APIs.** The events form a coherent trajectory from requirement definition and development to testing, deployment, and frontend integration. Nodes present event descriptions, participants, and timestamps, while edge labels denote cosine similarity between reciprocal Top-5 neighbors.

For query q, let R_q denote its annotated reference-message set and $M(v)$ the source messages of event v. The gold-evidence event set is defined as

$$\mathcal{E}_q^{\text{gold}} = \{v \in V \mid M(v) \cap R_q \neq \varnothing\}.$$

Let $\mathcal{S}_q = \{s_{q,1}, \ldots, s_{q,5}\}$ denote the Top-5 retrieved seed events. The distance from an evidence event e to the seed set is

$$d_{\bar{G}}(e, \mathcal{S}_q) = \min_{s \in \mathcal{S}_q} d_{\bar{G}}(s, e).$$

We report the cumulative evidence-event coverage within a maximum traversal depth h:

$$C(h) = \frac{\sum_q \sum_{e \in \mathcal{E}_q^{\text{gold}}} \mathbb{I}[d_{\bar{G}}(e, \mathcal{S}_q) \leq h]}{\sum_q |\mathcal{E}_q^{\text{gold}}|}.$$

We evaluate $h \in \{1, 2, 3, 4, 5\}$. We additionally report $C(\infty)$, the proportion of gold-evidence events that are reachable from the Top-5 seeds without a hop limit. Unreachable evidence events remain in the denominator.

To provide an intuitive view of the local structure formed by the event representations, we present a representative subgraph extracted from the constructed k-NN graph in Figure 6. This case captures the evolution of financial-goal APIs across multiple participants and dates, covering requirement definition and logical-deletion design, API development, progress tracking, regression testing, deployment, and frontend integration. Despite being distributed across different conversations and project stages, these events form a compact neighborhood with high reciprocal similarities, demonstrating that the k-NN graph connects complementary events that collectively describe the progression of the same task.

F Experiment Settings Parameter Sensitive Analysis

***Number of hops*.** We compare one-hop, two-hop, and combined one- and two-hop expansion to evaluate how expansion depth affects memory retrieval.

***Consolidation Coefficient* α.** We vary α among 0, 1, 5 and 10, where a larger α indicates stronger aggregation of neighboring event, with the no-formatting situation ($\alpha = 0$) as a reference, to examine the sensitivity of PIT to this consolidation coefficient.

Table 13: Parameter sensitivity results using GPT-4.1-mini.

GPT-4.1-mini	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
PIT	81.22	22.89	21.67	79.10	69.09	63.06	29.55	30.77	43.37	48.97
Number of hops										
hop = 1	77.93	16.06	21.00	69.90	59.02	56.34	31.25	30.77	37.24	44.39 (-4.58)
hop = 2	80.28	18.07	13.67	75.62	67.45	54.85	27.84	35.50	45.92	46.58 (-2.39)
hop = 1,2	77.46	20.08	20.00	76.37	69.56	64.55	30.11	31.50	43.94	48.17 (-0.80)
Coefficient α										
$\alpha = 0$	84.51	20.08	25.00	75.36	70.26	60.30	25.68	23.73	40.92	47.32 (-1.65)
$\alpha = 1$	85.45	21.18	24.74	77.36	65.84	63.46	23.64	21.36	30.92	45.99 (-2.98)
$\alpha = 5$	82.62	21.88	20.64	80.16	70.46	65.24	28.45	30.14	42.34	49.10 (+0.13)
$\alpha = 10$	71.42	16.94	15.38	75.14	63.42	56.74	23.46	20.48	40.54	42.61 (-6.36)

Table 14: Parameter sensitivity results using Qwen-3.5-Flash.

Qwen-3.5-Flash	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
PIT	81.69	17.67	8.67	78.36	90.40	72.01	27.27	50.30	56.63	53.67
Number of hops										
hop = 1	78.26	13.63	7.46	64.35	80.36	70.42	26.78	51.35	55.36	49.77 (-3.90)
hop = 2	80.42	15.26	6.28	75.80	84.64	70.36	26.24	49.48	55.90	51.60 (-2.07)
hop = 1,2	80.96	17.60	8.06	74.86	86.48	71.38	26.94	50.46	55.80	52.50 (-1.17)
Coefficient α										
$\alpha = 0$	80.98	18.88	11.00	82.09	90.87	70.24	26.55	51.48	47.93	53.34 (-0.33)
$\alpha = 1$	81.64	19.54	10.66	83.48	89.62	72.46	23.56	49.64	43.68	52.70 (-0.97)
$\alpha = 5$	79.48	16.58	9.42	84.75	89.53	73.82	25.21	50.14	46.59	52.84 (-0.83)
$\alpha = 10$	72.45	10.74	4.90	84.94	84.34	64.52	23.42	46.96	44.84	48.57 (-5.10)

Table 15: Parameter sensitivity results using GLM-4.

GLM-4	Fine-Grained Recall			Memory Awareness			Profile Understanding			Average
	Single	Multi	Temp	Const	Proact	Update	Style	Skill	Role	
Piece	85.92	22.49	36.00	85.07	81.26	66.79	20.45	58.58	63.27	57.76
Number of hops										
hop = 1	80.25	17.96	27.92	78.76	74.68	60.94	21.43	57.36	64.28	53.73 (-4.03)
hop = 2	79.84	17.30	28.49	70.15	75.98	61.52	19.40	55.96	62.48	52.35 (-5.41)
hop = 1,2	84.52	21.63	34.94	82.50	81.54	62.49	22.48	59.26	65.29	57.18 (-0.58)
Coefficient α										
$\alpha = 0$	83.56	21.53	35.33	81.07	82.04	63.45	18.45	53.83	62.46	55.75 (-2.01)
$\alpha = 1$	82.94	23.50	36.82	81.94	83.21	61.42	17.35	52.48	60.92	55.62 (-2.14)
$\alpha = 5$	84.35	20.42	35.10	84.05	80.46	66.92	21.38	59.20	65.36	57.47 (-0.29)
$\alpha = 10$	75.42	20.40	31.84	78.24	79.40	52.48	18.14	50.26	59.80	51.78 (-5.98)