# TRANSMEM: Transition-Aware Retrieval for Evolving Personal Memory

**Shigeng Chen**[1]* **Linhao Luo**[2]* **Changlong Shi**[3] **Zhangchi Qiu**[1] **Shirui Pan**[1]†
[1]Griffith University [2]Monash University
[3]The Hong Kong Polytechnic University
shigeng.chen@griffithuni.edu.au s.pan@griffith.edu.au

## Abstract

Long-term personalized agents are increasingly expected to answer queries over user histories in which preferences, goals, and constraints evolve across interleaved conversations. Existing retrieval-based and structured-memory systems can surface relevant facts, but often return isolated memory units that do not explain which earlier preference was refined, contradicted, or superseded by later evidence. We introduce TRANSMEM, a transition-aware memory retrieval framework that stores preference updates as explicit retrieval units alongside ordinary factual memories. During memory construction, TRANSMEM routes user interaction history into topic-structured memory graphs, detects shift events, and grounds them as transition records. At inference time, TRANSMEM first retrieves factual anchors and then adaptively expands over transition records, producing a compact context of factual anchors and accepted transition records. Experiments on PersonaMem and PersonaBench across two Qwen backbones show that TRANSMEM outperforms representative retrieval-based and structured-memory baselines. Controlled analyses further show that transition records and adaptive expansion contribute beyond factual anchoring, chronological expansion, flat transition retrieval, and fixed-hop traversal. These results suggest that preference-update records are useful retrieval units for long-term personalization. Code is available at https://github.com/Aries-chen/TransMem.

## 1 Introduction

Long-term memory is becoming central to personalized language agents. As LLM-based assistants interact with users over extended periods, they must track not only isolated facts but also evolving preferences, goals, routines, and constraints [19, 37, 17]. This is difficult because users are non-stationary: a preference for quick solo workouts may shift toward longer, social outdoor activities after joining a hiking group. These changes are rarely cleanly segmented; outdated preferences, transient states, and current intentions often coexist across interleaved conversational threads. Multi-turn fragility further compounds the problem, as even strong LLMs can over-rely on premature assumptions and fail to recover from early errors [10]. Recent benchmarks similarly show persistent weaknesses in multi-session reasoning, temporal updating, and dynamic user-profile tracking under long, topic-interleaved contexts [11, 16, 30, 25, 9].

Existing memory-augmented approaches have made substantial progress in storing, organizing, and retrieving personal information. Traditional retrieval-augmented generation (RAG) retrieves semantically similar chunks from past interactions [12], but such similarity-driven retrieval can surface stale or noisy evidence when user histories span long time periods, especially under knowledge

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

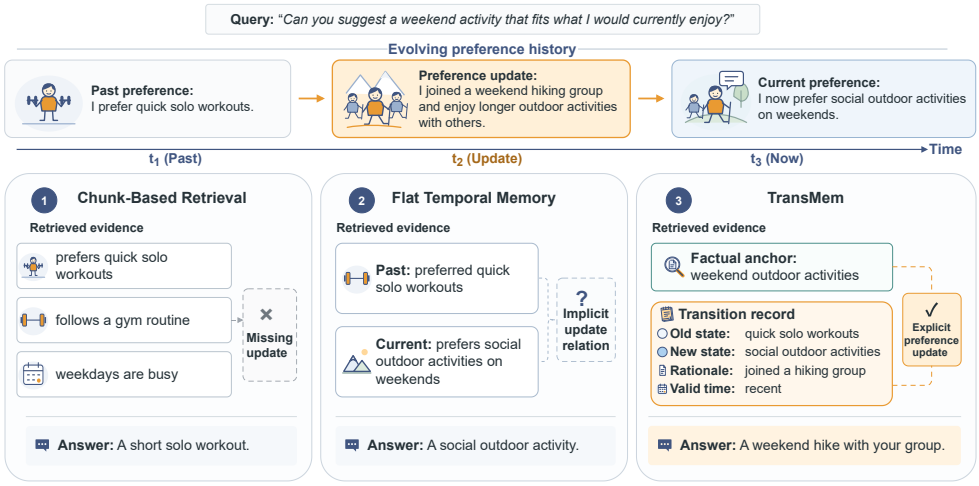


Figure 1: Evolving personal preferences: snippets can be stale and temporal facts leave updates implicit, while TRANSMEM retrieves factual anchors with explicit transitions.

volatility and temporal conflicts [20]. Recent memory systems introduce stronger organization and consolidation mechanisms, including sentence graph memory [31], temporal knowledge graphs [22], scalable memory stores [4], agentic note organization [32], staged memory architectures [5], and recollection-style retrieval [36]. These systems improve how memories are stored or selected, yet evolving personal preferences introduce a distinct retrieval challenge: the decisive evidence is often not a single memory item, but the update that explains how one user state superseded another.

A subtle difficulty is that the evidence needed to resolve a preference update is not always the evidence most similar to the query. A user may ask for a current recommendation without mentioning the historical event that changed the preference. Even when old and new facts are retrieved together, presenting them as independent snippets leaves the preference-update relation for the model to infer. The useful evidence is therefore often the update itself: what the user previously preferred, what replaced it, why it changed, and when the new state became valid.

Figure 1 contrasts recommendations based on stale snippets, independent temporal facts, and explicit preference-update evidence. This motivates *preference-transition retrieval*: retrieving the update relation itself rather than leaving the model to infer it from disconnected factual snippets. We introduce **TRANSMEM**, a transition-aware memory retrieval framework that stores preference updates as transition records alongside ordinary factual memories. During construction, TRANSMEM routes streaming interactions into topic-structured memory graphs and detects shift events as transition records. During retrieval, it first grounds the query in factual anchors and then adaptively expands over adjacent transition records when update evidence is needed. This design recovers compact preference-update trajectories while avoiding unrelated chronological neighbors and keeping online retrieval lightweight.

We evaluate TRANSMEM on complementary personalized-memory benchmarks, with PersonaMem [9] stressing long-context, topic-interleaved preference evolution and PersonaBench [25] testing broader personalized QA. Experimental results show that TRANSMEM consistently achieves the best overall score among representative baselines. Ablation and control experiments with factual-only retrieval, chronological expansion, flat transition retrieval, and fixed-hop expansion show that the gains are not explained by factual anchoring, temporal-neighbor context, direct transition-text retrieval, or fixed traversal depth alone.

Our contributions are as follows. First, we identify preference updates as a missing retrieval unit in long-term personalized memory and represent them as structured transition records. Second, we introduce TRANSMEM, which combines topic-structured memory construction, shift-event detection, and adaptive transition-aware retrieval. Third, we evaluate TRANSMEM on PersonaMem and PersonaBench across two Qwen backbones, with controlled analyses isolating the roles of transition records and adaptive expansion.

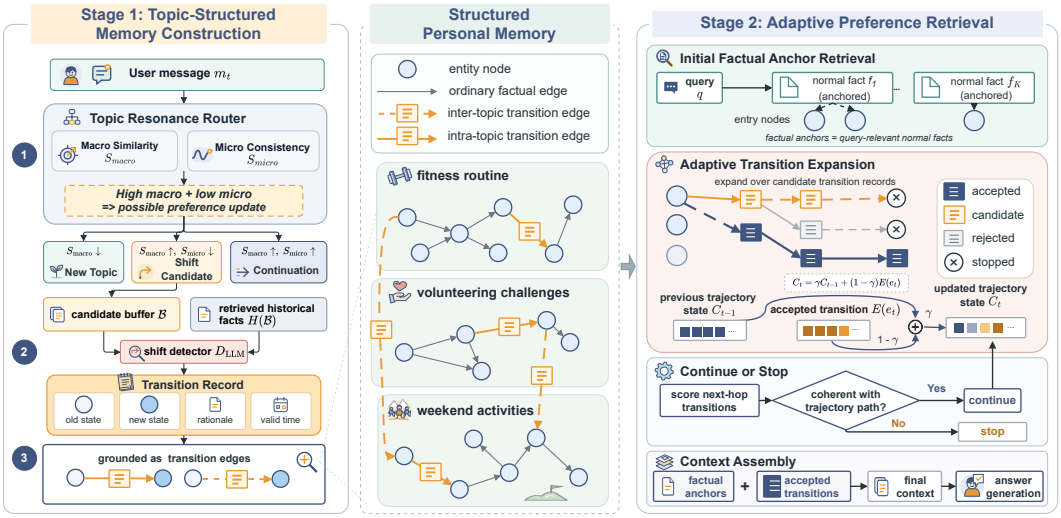


Figure 2: Overview of TRANSMEM. Stage 1 (memory construction): ❶ organize messages into topic graphs via the Topic Resonance Router; ❷ compare candidate messages with historical facts to extract transition records; ❸ ground transition records into structured personal memory shown in the center. Stage 2 (inference): retrieve factual anchors and adaptively expand over transition records to assemble the final context.

## 2 Methodology

**Preliminary.** We consider personalized question answering for a user $u$ with chronological message history $(m_1, \ldots, m_t)$. A memory-augmented system builds a memory $\mathcal{M}_u$ from this history and retrieves an evidence context $\mathcal{C}(q)$ for a query $q$. An LLM with parameters $\theta$ then generates an answer $y$ conditioned on the query and retrieved evidence:

$$\mathcal{C}(q) = \text{Retrieve}(q, \mathcal{M}_u), \qquad y \sim p_\theta(\cdot \mid q, \mathcal{C}(q)) .$$

Questions may concern historical facts, current preferences, or changes across time.

**Overview.** We propose TRANSMEM, a two-stage framework for retrieving evidence from long-term personal memory under evolving user preferences. The construction stage organizes each user's interaction history into topic-structured memory graphs and stores both ordinary factual memories and explicit preference-update records. The retrieval stage first selects query-relevant factual anchors and then adaptively expands over stored transition records when preference-update evidence is needed. Figure 2 illustrates the overall pipeline.

**Notation.** In TRANSMEM, the memory $\mathcal{M}_u$ is represented by the user memory graph $G_u$. We use *topic* to denote a fine-grained, user-specific topic graph (e.g., fitness routine), rather than a coarse domain label such as hobbies or work. We denote the $i$-th topic graph by $T_i$ with summary embedding $s_i$. A *normal fact* is an atomic, non-transition memory extracted from the user history. A *transition record* is a structured old-to-new preference update detected from historical evidence. A transition record stores the content of the update, while its grounded transition edge provides the graph structure used for traversal.

### 2.1 Topic-Structured Memory Construction

**Topic Resonance Router.** Each incoming message $m_t$ is routed through a two-stage decision over existing topic graphs. The first gate measures semantic alignment with each topic summary,

$$S_{\text{macro}}(m_t, T_i) = \cos(E(m_t), s_i) , \tag{1}$$

where $E(\cdot)$ is the text embedding operator. Let

$$i^* = \arg\max_i S_{\text{macro}}(m_t, T_i), \qquad M_t = S_{\text{macro}}(m_t, T_{i^*}). \tag{2}$$

The second gate measures entity-level consistency with the candidate topic, namely how strongly the entities in $m_t$ overlap with the entities already associated with $T_{i^*}$:

$$S_{\text{micro}}(m_t, T_{i^*}) = \frac{|K(m_t) \cap K(T_{i^*})|}{\max(1, |K(m_t)|)}, \tag{3}$$

where $K(m_t)$ extracts entities from the message and $K(T_i)$ denotes entities already associated with facts in topic graph $T_i$. The denominator is clipped by one so that messages without extracted entities do not produce undefined scores. We write $\mu_t = S_{\text{micro}}(m_t, T_{i^*})$.

The routing decision is

$$r_t = \begin{cases} \textsc{Birth}, & M_t < \tau_{\text{macro}}, \\ \textsc{Deepening}, & M_t \geq \tau_{\text{macro}} \wedge \mu_t \geq \tau_{\text{micro}}, \\ \textsc{Evolve}, & M_t \geq \tau_{\text{macro}} \wedge \mu_t < \tau_{\text{micro}}. \end{cases} \tag{4}$$

A low macro score creates a new topic graph. A high macro score and high micro score indicate that the message continues an existing topic. A high macro score but low micro score means that the message is topically related but introduces different entities or constraints; TRANSMEM therefore treats it as a candidate preference update rather than immediately committing it as a transition. Regardless of the routing outcome, each message is ingested into $G_u$ so that the memory graph remains a complete record.

**Shift Detection.** Messages routed as EVOLVE are accumulated in a candidate-update buffer $\mathcal{B}$. Detection is triggered only when the buffer reaches size $\kappa$, which reduces spurious updates caused by isolated or ambiguous messages. The router determines which messages become candidate updates, while the detector compares the resulting buffer against retrieved historical facts from the user memory graph:

$$\mathcal{H}(\mathcal{B}) = \underset{f \in \mathcal{F}_u}{\arg\,\text{topK}}\, s_{\text{hist}}(\mathcal{B}, f), \tag{5}$$

where $\mathcal{F}_u$ denotes the normal facts stored in $G_u$, and $s_{\text{hist}}$ scores their relevance to the buffered candidate messages.

An LLM-based shift detector then compares the buffered candidate messages with the retrieved historical facts and outputs structured shift events:

$$\mathcal{Z} = \mathcal{D}_{\text{LLM}}(\mathcal{B}, \mathcal{H}(\mathcal{B}))\,. \tag{6}$$

We use *shift event* for the detector output and *transition record* for the stored retrieval unit derived from it. Each detected update is represented as

$$z = \big(s^-, s^+, r, g, t_{\text{valid}}\big), \tag{7}$$

where $s^-$ is the old preference state, $s^+$ is the new preference state, $r$ is the event type, $g$ is the update rationale explaining why the change occurred, and $t_{\text{valid}}$ records when the new state becomes valid.

**Transition Linking and Edge Construction.** Let $V_u$ be the set of entity nodes in $G_u$, where entities include concepts such as activities and interests. Each topic graph $T_i$ is a subgraph of $G_u$. For each detected update $z \in \mathcal{Z}$, we ground its old and new states to existing nodes:

$$v^a = \underset{v \in V_u}{\arg\max} \cos(E(s^a), E(v))\,, \qquad a \in \{-, +\}. \tag{8}$$

Here, $E(v)$ denotes the text embedding of the node's entity or concept name. We retain the update only if both endpoint similarities are at least $\tau_g$. The existing nodes provide structural anchors, while the transition record preserves the complete states $s^-$ and $s^+$. Each retained record is stored with a directed transition edge:

$$e_z = \big(v^-, v^+, r, y_e, g, t_{\text{valid}}\big), \qquad y_e \in \{\text{intra}, \text{inter}\}, \tag{9}$$

where $v^-$ and $v^+$ are the grounded endpoints. The stored text follows the detector fields and is embedded as the retrieval representation of the transition record. The same shift detector is used for all transitions. After grounding $s^-$ and $s^+$ to graph nodes, we label the resulting transition according to endpoint assignment: it is an *intra-topic transition edge* if both endpoints belong to the same topic graph, and an *inter-topic transition edge* otherwise. Thus, the intra/inter distinction reflects where the old and new states are grounded, not a separate detector or supervision signal. The update rationale $g$ and valid time $t_{\text{valid}}$ are retained as part of the transition record and may be verbalized in the final retrieved context.

### 2.2 Adaptive Preference Retrieval

At inference time, TRANSMEM retrieves evidence in two stages: initial factual anchoring, followed by adaptive transition expansion. The final context contains factual anchors and accepted transition records, whose fields include the old and new states.

**Initial Factual Anchor Retrieval.** Given a query $q$, TRANSMEM first retrieves query-relevant normal facts:

$$\mathcal{A}(q) = \underset{f \in \mathcal{F}_{\text{normal}}}{\arg\,\text{topK}}\, s(q, f), \tag{10}$$

where $\mathcal{F}_{\text{normal}}$ contains all normal factual memories and $s(q, f)$ is the factual relevance score. The selected facts $\mathcal{A}(q)$ are *factual anchors*: they are included directly in the final context and also provide entry points into the transition graph. Each normal fact is stored as an atomic fact edge in $G_u$ with head and tail nodes; the endpoints of selected factual anchors provide the entry nodes for transition expansion.

**Adaptive Transition Expansion.** Let $p$ denote a partial transition path. For the empty path, candidate transition edges are adjacent to the endpoints of the factual anchors; after a transition is accepted, candidates are adjacent to the target endpoint of the current path. For retrieval scoring, each transition edge is represented by the text of its associated transition record. Let $C_p$ be the trajectory state of path $p$, with $C_\emptyset = E(q)$. A candidate next transition $e$ is scored by

$$\sigma(e \mid p) = \cos(C_p, E(e))\,, \tag{11}$$

where $E(e)$ embeds the text associated with transition edge $e$. A candidate is retained only if it remains coherent with the current trajectory:

$$e \in \Phi(p) \quad \Longleftrightarrow \quad \sigma(e \mid p) \geq \tau_c \;\wedge\; (p = \emptyset \;\vee\; \sigma(e \mid p) - \sigma_{\text{last}}(p) \geq -\delta_c)\,, \tag{12}$$

where $\tau_c$ is the continuation threshold, $\delta_c$ is the drop tolerance, and $\sigma_{\text{last}}(p)$ is the score of the most recently accepted transition in $p$. The first condition requires the next transition to remain relevant to the evolving trajectory. The second condition prevents abrupt coherence collapse within a single hop.

When a transition edge $e$ is accepted, its target endpoint determines the next structural neighborhood, while its associated transition record updates the trajectory state:

$$C_{p \oplus e} = \gamma C_p + (1 - \gamma) E(e), \tag{13}$$

where $p \oplus e$ denotes the path obtained by appending $e$. This interpolation keeps the trajectory state anchored to the original query while allowing accepted transition records to steer later hops, preventing expansion from drifting to locally adjacent but semantically unrelated transitions. The final retrieved context consists of factual anchors and accepted transition records, allowing the QA model to resolve stale and current preferences without retrieving unrelated chronological neighbors.

## 3 Experiments

### 3.1 Experimental Setup

**Datasets.** We evaluate on two complementary personalized-memory benchmarks. **PersonaMem** [9] directly stresses long-context preference evolution: each user is associated with a chronological message stream and questions spanning seven preference- and memory-related categories, and we use both its 32k-token and 128k-token context regimes. **PersonaBench** [25] provides a broader personalized-QA setting over synthetic private user data and evaluates answer quality with lexical F1. Together, the two benchmarks test whether transition-aware retrieval helps both in long, topic-interleaved preference histories and in a separate personalized-QA benchmark. Dataset and evaluation details are provided in Appendix A.

**Baselines.** We compare TRANSMEM against three baseline groups. *Zeroshot-only*: ZEROMEM provides no user history. *Retrieval-based*: NAIVERAG [12] and RF-MEM [36] select passages by query relevance. *Structured memory*: ZEP [22], A-MEM [32], MEM0 [4], and LIGHTMEM [5] maintain external user memory representations.

Table 1: PersonaMem results in the 32k context regime. Best results in bold; second-best underlined.

| | Method | Revisit Reasons | Track Evolution | Latest Prefs | Aligned Recs | New Scenario | Shared Facts | New Ideas | Overall |
|---|---|---|---|---|---|---|---|---|---|
| *Qwen3-8B* | ZEROMEM | 57.41 | 55.00 | 29.41 | 29.63 | 18.37 | 35.40 | 9.09 | 34.53 |
| | NAIVERAG | 68.52 | 54.00 | 58.82 | 38.89 | 51.02 | 54.87 | 18.18 | 47.37 |
| | RF-MEM | 53.70 | 56.00 | 35.29 | 61.11 | 42.86 | 57.52 | 19.32 | 47.79 |
| | ZEP | 68.52 | 55.00 | 52.94 | **66.67** | 61.22 | 54.87 | 22.73 | 52.42 |
| | A-MEM | **75.93** | 61.00 | **70.59** | 62.96 | 44.90 | 53.98 | 23.86 | 53.05 |
| | MEM0 | 66.67 | 51.00 | 64.71 | 61.11 | 48.98 | 51.33 | 10.23 | 46.74 |
| | LIGHTMEM | 59.26 | 50.00 | 64.71 | 61.11 | 63.27 | **63.72** | 20.45 | 52.00 |
| | TRANSMEM (ours) | **75.93** | **62.00** | 58.82 | 59.26 | **67.35** | 61.06 | **26.14** | **56.84** |
| *Qwen3.5-27B* | ZEROMEM | 66.67 | 57.00 | 23.53 | 42.59 | 26.53 | 36.28 | 7.95 | 38.11 |
| | NAIVERAG | 75.93 | 64.00 | 64.71 | 70.37 | 71.43 | 61.06 | 34.09 | 60.63 |
| | RF-MEM | **85.19** | 55.00 | 64.71 | 77.78 | 65.31 | 61.06 | 40.91 | 61.26 |
| | ZEP | 79.63 | 61.00 | 70.59 | 72.22 | 81.63 | 62.83 | 40.91 | 63.58 |
| | A-MEM | 68.52 | 61.00 | 70.59 | 75.93 | 61.22 | 71.68 | **50.00** | 64.42 |
| | MEM0 | 81.48 | **65.00** | 47.06 | 74.07 | 87.76 | **72.57** | 25.00 | 64.00 |
| | LIGHTMEM | 66.67 | 61.00 | 64.71 | 70.37 | 71.43 | 65.49 | 35.23 | 60.21 |
| | TRANSMEM (ours) | **85.19** | 62.00 | **82.35** | **79.63** | **89.80** | 71.68 | 44.32 | **69.26** |

Table 2: PersonaMem results in the 128k context regime. Best results in bold; second-best underlined.

| | Method | Revisit Reasons | Track Evolution | Latest Prefs | Aligned Recs | New Scenario | Shared Facts | New Ideas | Overall |
|---|---|---|---|---|---|---|---|---|---|
| *Qwen3-8B* | ZEROMEM | 44.30 | 50.91 | 34.83 | 30.65 | 38.60 | 26.62 | 17.86 | 32.67 |
| | NAIVERAG | 48.10 | 53.18 | 56.60 | 46.13 | 32.16 | 53.25 | 21.13 | 44.42 |
| | RF-MEM | 46.84 | 54.55 | 52.83 | 43.23 | 28.65 | 48.05 | 22.22 | 42.43 |
| | ZEP | 50.63 | 52.73 | 54.28 | 49.03 | 45.61 | 58.44 | 22.22 | 45.90 |
| | A-MEM | 51.27 | 53.64 | 52.54 | **53.55** | 29.82 | 51.95 | **26.14** | 45.26 |
| | MEM0 | 50.00 | 53.64 | 56.17 | 45.48 | 32.75 | 51.30 | 21.35 | 44.33 |
| | LIGHTMEM | 44.94 | 52.73 | 55.30 | 46.13 | 47.95 | **64.29** | 23.53 | 46.27 |
| | TRANSMEM (ours) | **51.90** | **55.00** | **56.75** | 51.29 | **53.22** | 62.99 | 21.13 | **48.03** |
| *Qwen3.5-27B* | ZEROMEM | 63.92 | 58.64 | 38.90 | 32.90 | 31.58 | 47.40 | 32.24 | 40.49 |
| | NAIVERAG | **72.78** | 59.55 | 63.43 | 62.26 | 47.37 | 79.22 | 41.61 | 58.77 |
| | RF-MEM | 72.15 | 60.91 | 69.81 | 58.06 | 50.29 | 78.57 | 38.56 | 59.83 |
| | ZEP | 69.62 | 50.45 | **70.54** | 63.23 | 51.46 | 73.38 | 43.36 | 60.30 |
| | A-MEM | 68.35 | 60.45 | 66.04 | 66.13 | 68.42 | 81.17 | 41.18 | 61.64 |
| | MEM0 | **72.78** | 59.55 | 64.88 | 62.26 | 68.42 | 81.82 | 40.52 | 60.85 |
| | LIGHTMEM | 67.72 | 57.73 | 62.41 | 55.16 | 57.31 | 79.22 | 41.39 | 57.61 |
| | TRANSMEM (ours) | 70.89 | **62.27** | 69.23 | **67.74** | **73.68** | **85.06** | **44.66** | **64.69** |

**Implementation Settings.** We use two open-source backbones, Qwen3-8B [33] and Qwen3.5-27B [21], with each used for both memory construction and final answer generation. Implementation details are reported in Appendix B.

### 3.2 Main Results

We first test whether retrieving preference-update records improves end-to-end personalized QA. Across the evaluated PersonaMem and PersonaBench settings, TRANSMEM achieves the highest overall score among representative retrieval-based and structured-memory baselines, with the largest improvement reaching 4.84 overall points. Controlled analyses with factual-only retrieval, chronological expansion, flat transition retrieval, and fixed-hop expansion further show that the improvements are not explained by factual anchoring, temporal-neighbor context, direct transition-text retrieval, or fixed traversal depth alone.

Table 1 shows that TRANSMEM does not dominate every category, but it consistently improves the overall score under both backbones. Baselines remain competitive on locally answerable or stable-fact categories, while TRANSMEM achieves the highest *New Scenario* accuracy under both backbones and the highest *Track Evolution* accuracy with Qwen3-8B. This pattern is consistent with our view that the useful retrieval unit is often not an isolated fact but the transition that explains how one user state supersedes another.

Table 3: PersonaBench results (lexical F1, %). Best results in bold; second-best underlined.

| | Method | Basic Info | Pref Easy | Pref Hard | Social | Overall |
|---|---|---|---|---|---|---|
| *Qwen3-8B* | NAIVERAG | 13.96 | 21.29 | 24.48 | 15.65 | 17.88 |
| | RF-MEM | 18.63 | <u>23.80</u> | 18.60 | 15.88 | 18.58 |
| | A-MEM | 16.71 | 21.54 | 26.71 | 21.82 | 20.21 |
| | LIGHTMEM | <u>19.37</u> | 23.38 | <u>29.05</u> | **23.21** | <u>22.43</u> |
| | TRANSMEM (ours) | **23.80** | **25.14** | **30.16** | <u>22.00</u> | **24.67** |
| *Qwen3.5-27B* | NAIVERAG | 14.31 | 23.11 | 22.93 | 16.08 | 18.07 |
| | RF-MEM | 14.61 | 16.08 | 23.11 | 22.93 | 18.21 |
| | A-MEM | 27.34 | 23.63 | <u>24.53</u> | <u>35.85</u> | <u>28.38</u> |
| | LIGHTMEM | **33.26** | <u>31.61</u> | 23.29 | 20.45 | 28.34 |
| | TRANSMEM (ours) | <u>28.19</u> | **31.79** | **24.95** | **38.46** | **30.38** |

Table 2 reports the 128k regime, which more directly stresses the failure mode in Figure 1: relevant evidence is separated by longer spans of unrelated history, and old and new user states are more likely to coexist. Under this regime, TRANSMEM again achieves the best overall accuracy with both backbones. The category-level pattern is most informative on *Track Evolution* and *New Scenario*, where TRANSMEM is strongest under both backbones. Open-ended ideation categories such as *New Ideas* are less diagnostic of transition retrieval because success also depends on preference-consistent generation and creative extrapolation; we analyze one such case in Appendix C.7. We further note that with Qwen3.5-27B, FULLTEXT attains the strongest scores on *Revisit Reasons* and *Track Evolution*, where direct access to the complete history removes retrieval noise; this advantage scales linearly with context cost and is unavailable for Qwen3-8B in the 128k regime (Appendix C.1).

PersonaBench provides a complementary open-ended evaluation beyond PersonaMem's multiple-choice format. As shown in Table 3, TRANSMEM achieves the best overall F1 under both backbones, with category-level differences reflecting the open-ended nature of this benchmark. These results indicate that transition-aware memory remains useful beyond PersonaMem, although absolute F1 remains modest in this harder open-ended setting.

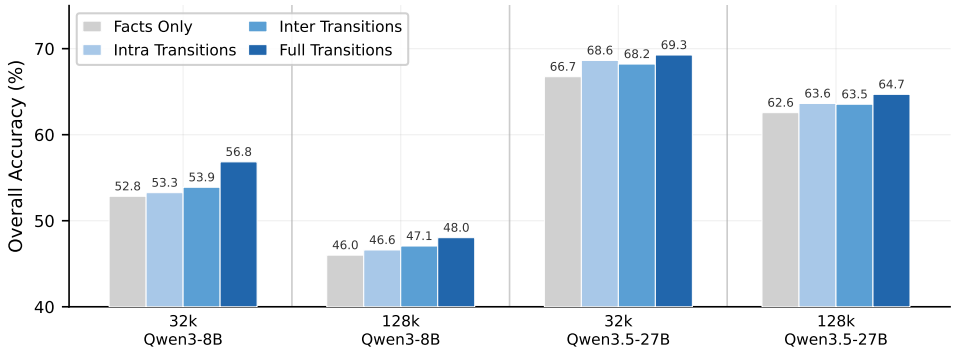


Figure 3: Ablation results comparing four retrieval variants across backbone and context settings. *Facts Only*: factual retrieval without transition edges; *Intra Transitions*: intra-topic transition edges only; *Inter Transitions*: inter-topic transition edges only; *Full Transitions*: TRANSMEM with both edge types.

### 3.3 Ablation Study

We next isolate whether explicit transition structure contributes beyond factual anchoring. Figure 3 compares four retrieval variants that progressively add transition-edge structure across both backbones and context regimes. Across all four evaluation settings, the *Facts Only* variant yields the lowest overall accuracy, indicating that time-aware factual anchors alone do not fully resolve revised or conflicting user states. Adding either intra-topic or inter-topic transition edges improves over the factual-only baseline, while the full model performs best in every setting. In three of four settings, inter-topic transition edges contribute slightly more than intra-topic edges alone (e.g., +1.05 vs.

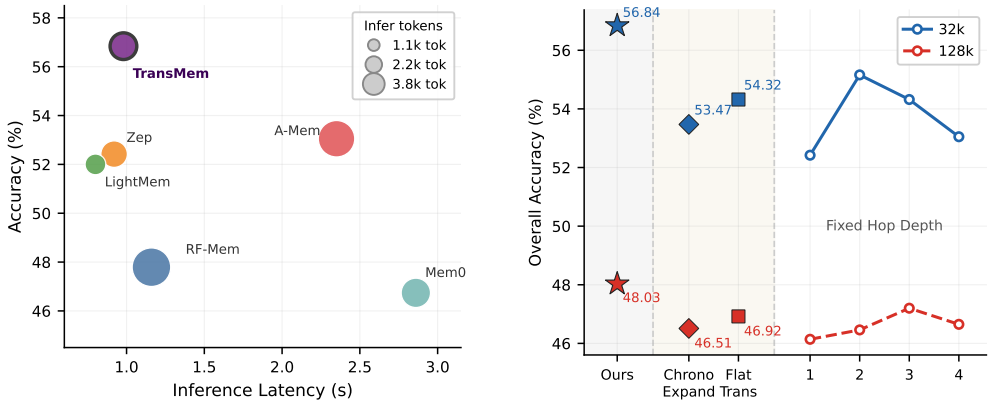


Figure 4: **Left**: Inference latency and token usage versus overall accuracy in the 32k-8B setting; bubble area is proportional to inference token count. **Right**: Comparison of TRANSMEM with chronological expansion (CHRONOEXPAND), flat transition retrieval (FLATTRANS), and fixed-hop transition expansion.

+0.42 pp on 32k-8B), though the 32k-27B setting reverses this order (+1.89 vs. +1.47 pp). Thus, transition records provide information beyond factual anchors, and intra- and inter-topic transitions contribute complementary signals. A per-category comparison is shown in Figure 5.

### 3.4 Accuracy–Efficiency Trade-off

We then ask whether the improvements come from retrieving more evidence or more informative evidence. We measure end-to-end inference time from query submission to final answer, including retrieval, context assembly, retrieval-time model calls, and QA generation. In our reproduced pipelines, MEM0 and A-MEM invoke an additional LLM call during retrieval, whereas TRANSMEM performs embedding-space expansion after memory construction. As shown in Figure 4 (left), TRANSMEM achieves the highest accuracy with 1.82k inference tokens and 0.98 s latency, compared with 3.4k tokens and 2.35 s for the next-best system by accuracy, A-MEM. This suggests that the improvements come from retrieving more targeted evidence rather than simply increasing the amount of context.

### 3.5 Analysis of Retrieval Expansion Strategies

Finally, we isolate the expansion mechanism using three controls from the same factual anchors. CHRONOEXPAND tests whether ordinary temporal neighbors are sufficient by adding chronological normal-fact neighbors from the same topic timeline. FLATTRANS tests whether transition records alone are sufficient by retrieving them as flat documents with query–transition similarity, without following graph endpoints. Fixed-hop variants test whether a manually chosen traversal depth can replace adaptive stopping. TRANSMEM differs from all three by following graph-grounded transition endpoints while updating a trajectory state during expansion. These controls correspond to the main retrieval choices in Figure 2: whether to use transition records, whether to follow grounded endpoints, and whether to stop adaptively. Further details are in Appendix C.3.

Figure 4 (right) compares these strategies for the Qwen3-8B backbone under both context regimes. CHRONOEXPAND (53.47% / 46.51%) improves over the worst fixed depths but trails TRANSMEM, showing that nearby chronological normal facts alone are not enough. FLATTRANS (54.32% / 46.92%) improves further, confirming that transition records carry useful signal, but still falls short of TRANSMEM (56.84% / 48.03%). This indicates that the gain comes not only from storing transition text, but also from using the memory graph to choose which transition records to consider next. Fixed-hop retrieval further shows that no single depth is uniformly appropriate: hop 2 is best in 32k while hop 3 is best in 128k, whereas adaptive expansion exceeds the best fixed alternative in both regimes (by 1.68 pp and 0.83 pp respectively). Together, these controls show that transition records

are a useful retrieval unit, but their strongest effect comes from using them in an adaptive graph expansion rather than retrieving them as isolated text.

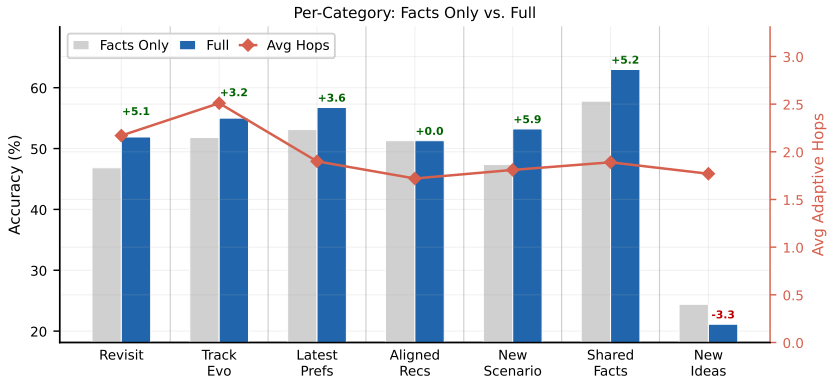


Figure 5: Per-category accuracy and adaptive retrieval depth on PersonaMem-128k (Qwen3-8B). Bars compare *Facts Only* with *Full Transitions*; the right axis shows average expansion depth.

**Category-wise Gains and Adaptive Depth.** Figure 5 examines how these gains vary across question categories in the 128k-8B setting. Full transition retrieval improves accuracy in five of seven categories, with the largest gains on *New Scenario* (+5.85 pp), *Shared Facts* (+5.20 pp), and *Revisit Reasons* (+5.06 pp). Expansion is deepest for *Track Evolution* (2.51 hops) and *Revisit Reasons* (2.17), consistent with retrieving longer trajectories for evolution-focused questions rather than imposing a fixed depth. *Aligned Recommendations* is unchanged, while *New Ideas* decreases by 3.27 pp, showing that additional transition context does not uniformly benefit open-ended generation. The 27B backbone improves across all seven categories; complete results and hop statistics are in Appendix C.4.

## 4 Related Work

**Agentic memory and memory management.** Long-term agent memory has been studied as a way to extend LLM-based agents beyond stateless interaction. Previous systems such as Generative Agents, MemoryBank, and MemGPT introduced persistent experience stores, reflection, hierarchical memory, and virtual context management for long-horizon interaction [19, 37, 17]. Recent systems improve memory writing and consolidation through temporal graphs, scalable memory stores, self-organizing notes, staged consolidation, and recollection-style retrieval [22, 4, 32, 5, 36]. Long-term dialogue and personalization systems further use event memory, persona modeling, segment-level memory construction, and persistent user profiles [13, 18, 29]. Other work studies modular, cognitive, or hierarchical memory organization for long-term agents and multi-agent systems [28, 34, 27, 24, 8, 14, 35]. A-Mem [32] organizes memories as linked notes and updates their contents and connections as new experiences arrive. TRANSMEM instead stores explicit old-to-new state transitions and conditions subsequent graph expansion on the transitions already accepted along the retrieval path.

**Memory retrieval.** Retrieval-augmented generation retrieves chunks or passages that are semantically relevant to the query [12]. Temporal retrieval and time-aware memory organize evidence with timestamps, temporal knowledge graphs, or memory-augmented temporal reasoning modules [20, 22, 31, 26]. Graph-based retrieval further supports multi-hop evidence access through graph-augmented passage ranking, path retrieval, and graph walks [6, 2, 1]. These approaches improve access to relevant passages, facts, graph neighborhoods, or reasoning paths. In evolving personal memory, however, the missing retrieval unit is often the preference update itself: which earlier state was superseded, what new state replaced it, why the change occurred, and when the new state became valid. Recent memory systems also explore inference-time controllers for adaptive retrieval or reader feedback [23, 7], representing a different online-computation trade-off from the lightweight retrieval setting studied here. TRANSMEM instead moves preference-update interpretation to offline memory construction and retrieves stored transition records with embedding-based transition expansion at inference time.

## 5 Conclusion

We studied long-term personal memory under non-stationary user preferences. TRANSMEM stores preference updates as transition records and retrieves them through factual anchoring followed by adaptive transition expansion. Across PersonaMem and PersonaBench, TRANSMEM achieves the strongest overall scores under both Qwen backbones, and controlled analyses show that transition records and adaptive expansion add value beyond factual anchoring, chronological expansion, flat transition retrieval, and fixed-hop traversal. These results suggest that preference-update records are useful retrieval units for personal agents that must distinguish current user states from superseded ones as users change over time.

## 6 Limitations

TRANSMEM relies on backbone language models for offline shift detection and final answer generation, so detector errors, hallucinations, or calibration failures may propagate into retrieval and responses. Our evaluation focuses on text-based personalized-memory benchmarks and two open-source Qwen backbones; future work should test broader model families, multimodal histories, and longer deployment horizons. As long-term personal memory may contain sensitive information and persistent memories can become stale or be manipulated, practical systems should support user inspection, correction, deletion, and access control [15]. A promising deployment direction is to store sensitive memory graphs locally or in user-controlled storage, exposing only the minimal retrieved evidence needed for each interaction.

## Acknowledgment

This research was supported through an Australian Research Council (ARC) Future Fellowship (FT210100097) and grant DP240101547.

## References

[1] Martin Böckling, Heiko Paulheim, and Andreea Iana. Walk&retrieve: Simple yet effective zero-shot retrieval-augmented generation via knowledge graph walks, 2025. URL https://arxiv.org/abs/2505.16849.

[2] Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pages 30183–30191, 2026.

[3] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

[4] 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.

[5] Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

[6] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LWH8yn4HS2.

[7] Yikuan Hu, Jifeng Zhu, Lanrui Tang, and Chen Huang. RemindRAG: Low-cost LLM-guided knowledge graph traversal for efficient RAG. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JnKfAqLJb4.

[8] Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, Zeying Xie, and Xiaofang Zhou. Licomemory: Lightweight and cognitive agentic memory for efficient long-term reasoning. *arXiv preprint arXiv:2511.01448*, 2025.

[9] Bowen Jiang, Zhuoqun Hao, Young Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo Jose Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. In *Second Conference on Language Modeling*, 2025.

[10] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

[11] Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. Realtalk: A 21-day real-world dataset for long-term conversation. *arXiv preprint arXiv:2502.13270*, 2025.

[12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 9459–9474, 2020.

[13] Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5259–5276, 2025.

[14] Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, et al. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*, 2025.

[15] Zehao Lin, Chunyu Li, and Kai Chen. A survey on the security of long-term memory in llm agents: Toward mnemonic sovereignty, 2026. URL https://arxiv.org/abs/2604.16548.

[16] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[17] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

[18] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

[19] Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

[20] Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. It's high time: A survey of temporal question answering, 2025. URL https://arxiv.org/abs/2505.20243.

[21] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[22] 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.

[23] Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fugnQxbvMm.

[24] Haoran Sun and Shaoning Zeng. Hierarchical memory for high-efficiency long-term reasoning in llm agents. *arXiv preprint arXiv:2507.22925*, 2025.

[25] Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh RN, Tulika Manoj Awalgaonkar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, et al. Personabench: Evaluating ai models on understanding personal information through accessing (synthetic) private user data. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 878–893, 2025.

[26] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. Memotime: Memory-augmented temporal knowledge graph enhanced large language model reasoning. In *Proceedings of the ACM Web Conference 2026*, pages 4220–4231, 2026.

[27] Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, et al. O-mem: Omni memory system for personalized, long horizon, self-evolving agents. *arXiv preprint arXiv:2511.13593*, 2025.

[28] Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

[29] Rebecca Westhäußer, Wolfgang Minker, and Sebatian Zepf. Enabling personalized long-term interactions in llm-based agents through persistent memory and user profiles, 2025. URL https://arxiv.org/abs/2510.07925.

[30] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

[31] Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. SGMem: Sentence graph memory for long-term conversational agents, 2026. URL https://openreview.net/forum?id=r5BbGRXU77.

[32] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

[33] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[34] Zihe Ye, Jingyuan Huang, Weixin Chen, and Yongfeng Zhang. H-mem: Hybrid multi-dimensional memory management for long-context conversational agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7756–7775, 2026.

[35] Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng YAN. G-memory: Tracing hierarchical memory for multi-agent systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=mmIAp3cVS0.

[36] Yingyi Zhang, Junyi Li, Wenlin Zhang, Pengyue Jia, Xianneng Li, Yichao Wang, Derong Xu, Yi Wen, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Evoking user memory: Personalizing LLM via recollection-familiarity adaptive retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=f7p0F2X6XN.

[37] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

# A Dataset Details

This work focuses on evolving personal preferences rather than general conversational memory. While benchmarks such as LoCoMo [16], LongMemEval [30], and REALTALK [11] evaluate broad long-term dialogue memory, we adopt PersonaMem and PersonaBench because they directly expose dynamic personalization and preference-oriented QA settings.

## A.1 PersonaMem

PersonaMem [9] is a long-term personalized-memory benchmark constructed from multi-session dialogue histories in which user attributes, preferences, and contexts gradually evolve as a result of life events and changing situations. Each user case consists of a chronological message stream paired with evaluation questions spanning seven categories that target different aspects of memory utilization and personalized generation; we adopt the released 32k-token and 128k-token settings.

**User-case unit.** In PersonaMem, the same persona can appear with different historical chat contexts. Since these histories are not necessarily identical, we use `shared_context_id` as the user-case unit. This ensures that each evaluation case is paired with its own personalized memory rather than merging distinct histories from the same persona name.

**Data-quality filtering.** We observed a form of context-independent answerability in which the question stem alone identifies the correct option without consulting the user history. We use GPT-4o as a data-quality judge to detect and remove such items; the prompt is reproduced in Appendix B.6. The numbers reported throughout the paper are computed on the same quality-filtered evaluation set for all methods. Table 6 compares the category composition before and after filtering. The share of Track Evolution questions decreases from 23.60% to 21.05% at 32k and from 12.26% to 10.18% at 128k.

Table 4: PersonaMem evaluation-set statistics after filtering. Token counts use `tiktoken` with `cl100k_base`.

| Statistic | 32k | 128k |
|---|---|---|
| User cases | 37 | 55 |
| Questions | 475 | 2,161 |
| Avg. history tokens / user | 26,592.51 | 121,346.55 |
| Avg. messages / user | 174.65 | 761.91 |

Table 5: Evaluation aspects in PersonaMem.

| Aspect | Evaluates |
|---|---|
| Revisit Reasons | Recall the reason or event that caused a user preference to change. |
| Track Evolution | Recover how a user preference changes across time rather than only the latest state. |
| Latest Prefs | Recognize the most recent preference expressed in the user history. |
| Aligned Recs | Recommend items or activities aligned with the user's current preferences. |
| New Scenario | Apply user preferences to a new situation not directly seen in the history. |
| Shared Facts | Recall static events, activities, or interests the user previously shared. |
| New Ideas | Suggest new items or activities not already mentioned when the user asks for novelty. |

Table 6: PersonaMem query-type distribution before and after filtering. Each cell reports the count and its percentage within the corresponding split.

| | 32k | | 128k | |
|---|---|---|---|---|
| Aspect | Before | After | Before | After |
| Revisit Reasons | 99 (16.81%) | 54 (11.37%) | 237 (9.69%) | 158 (7.31%) |
| Track Evolution | 139 (23.60%) | 100 (21.05%) | 300 (12.26%) | 220 (10.18%) |
| Latest Prefs | 17 (2.89%) | 17 (3.58%) | 772 (31.56%) | 689 (31.88%) |
| Aligned Recs | 55 (9.34%) | 54 (11.37%) | 320 (13.08%) | 310 (14.35%) |
| New Scenario | 57 (9.68%) | 49 (10.32%) | 195 (7.97%) | 171 (7.91%) |
| Shared Facts | 129 (21.90%) | 113 (23.79%) | 155 (6.34%) | 154 (7.13%) |
| New Ideas | 93 (15.79%) | 88 (18.53%) | 467 (19.09%) | 459 (21.24%) |
| Total | 589 (100%) | 475 (100%) | 2,446 (100%) | 2,161 (100%) |

### A.2 PersonaBench

PersonaBench [25] evaluates personalized QA over simulated private user data. We use the official six-user setting from the release, comprising 263 QA items. Each user's private data consists of short multi-turn memory segments (87.83 per user on average) rather than a single raw chat transcript. It evaluates four aspects: *Basic information* that covers stable personal attributes and basic facts, *Preference easy* that targets direct preference evidence, *Preference hard* requiring harder preference reasoning beyond direct lookup, and *Social* which covers social attributes and relationships derived from private user data. We report lexical F1 following the official evaluation protocol. We reuse the construction and retrieval hyperparameters selected on PersonaMem-32k and apply a uniform 1,000-token retrieved-context budget to all methods.

### A.3 Evaluations

**PersonaMem** [9] evaluates QA on four-option questions. For each question type $c$ with $n_c$ questions answered at accuracy $a_c$ (in %), the global overall accuracy is the weighted average across all types:

$$\text{Overall} = \frac{\sum_c a_c \times n_c}{\sum_c n_c}.$$

End-to-end QA performance on **PersonaBench** [25] is measured with lexical F1 using the benchmark's official evaluation protocol. Note that Recall@$k$ is not used as the metric because the retrieval unit is not comparable across memory systems: TRANSMEM retrieves atomic transition records and normal facts, whereas chunk-based baselines retrieve much larger segments.

After applying the official answer preprocessing, let $G_i$ and $H_i$ denote the sets of unique gold and predicted tokens for example $i$. We compute

$$\text{Prec}_i = \begin{cases} \frac{|G_i \cap H_i|}{|H_i|}, & |H_i| > 0, \\ 0, & |H_i| = 0, \end{cases} \qquad \text{Rec}_i = \begin{cases} \frac{|G_i \cap H_i|}{|G_i|}, & |G_i| > 0, \\ 0, & |G_i| = 0. \end{cases} \tag{14}$$

The per-example F1 is

$$\text{F1}_i = \begin{cases} \frac{2\,\text{Prec}_i\,\text{Rec}_i}{\text{Prec}_i + \text{Rec}_i}, & \text{Prec}_i + \text{Rec}_i > 0, \\ 0, & \text{Prec}_i + \text{Rec}_i = 0. \end{cases} \tag{15}$$

## B Implementation Details

### B.1 Implementation Setup

**Models.** We use two backbones, Qwen3-8B [33] and Qwen3.5-27B [21]. Each backbone is used for memory graph construction, shift detection ($\mathcal{D}_{\text{LLM}}$), and inference-time QA.

**Retriever.** We use BGE-M3 [3] as the dense retriever for all embedding-based computations in both the construction and retrieval stages.

**Graph construction engine.** We use a modified Graphiti[3] implementation (Apache-2.0) as the underlying graph construction engine.

**Hardware.** All experiments are conducted on a single NVIDIA A100 GPU.

**Construction cost.** Table 7 reports the average per-user token cost for offline memory construction, measured on PersonaMem 32k with Qwen3.5-27B. TRANSMEM is substantially cheaper than A-MEM (722k vs. 950k tokens per user, a $1.31\times$ reduction) while achieving higher overall accuracy; it is more expensive than MEM0 but recovers a clear accuracy advantage through preference-trajectory retrieval.

Table 7: Average per-user token cost for offline memory construction (PersonaMem 32k, Qwen3.5-27B).

| Method | Input | Output | Total |
|---|---|---|---|
| MEM0 | 434,977 | 121,250 | 556,227 |
| A-MEM | 867,453 | 82,513 | 949,966 |
| TRANSMEM (ours) | 620,976 | 101,201 | 722,177 |

### B.2 Memory Baselines

**RF-Mem [36].** RF-Mem models retrieval through a familiarity–recollection mechanism that first performs lightweight top-$k$ retrieval and then selectively expands evidence through a slower recollection path when uncertainty remains. It is a strong retrieval-oriented baseline because it adapts evidence collection at inference time without explicitly constructing or traversing preference-transition edges.

**Zep [22].** Zep builds agent memory on top of a temporal knowledge graph, converting conversations into entities, relationships, and temporally scoped facts via its Graphiti engine. Outdated facts are invalidated rather than deleted, so provenance is preserved. We reproduce Zep from its open-source release rather than its commercial API.

**A-Mem [32].** A-Mem introduces an agentic memory system inspired by the Zettelkasten method, converting each interaction into a structured note with contextual descriptions and keywords, then dynamically linking it to related historical memories. Unlike flat vector stores, A-Mem emphasizes note-level self-organization and memory updating, making it a meaningful comparison for preference-tracking tasks.

**Mem0 [4].** Mem0 is a scalable long-term memory architecture that dynamically extracts, consolidates, updates, and retrieves salient information from ongoing conversations. We use the graph-enhanced variant, which additionally represents relational structures among memory items, and reproduce it from the open-source release rather than the commercial API.

**LightMem [5].** LightMem is an efficiency-oriented memory framework using a three-stage pipeline inspired by the Atkinson–Shiffrin model: sensory memory filters raw interactions, short-term memory groups and summarizes by topic, and long-term memory performs offline consolidation. This design reduces online overhead and is a strong baseline for understanding whether structured compression alone suffices in long-context settings.

### B.3 Intra- and inter-topic transitions

Preference updates are not always expressed within a single stable topic boundary. Such updates may occur within a fine-grained topic graph, such as revising a fitness routine, or across related topic graphs, such as a change in social life altering what outdoor activities the user now prefers. We

[3] `https://github.com/getzep/graphiti`

therefore label grounded transition edges by the topic assignment of their endpoints, which helps separate local refinements from updates whose consequences cross fine-grained topic graphs.

**Intra-topic transitions.** An intra-topic transition captures a preference update whose old and new states are grounded within the same topic graph. For example, a user may initially prefer short morning workouts but later state that they now prefer longer evening strength sessions after changing their work schedule. Both states belong to a `fitness-routine` topic, but the transition record is still necessary because the later state revises how future fitness-related recommendations should be interpreted. Thus, intra-topic transitions primarily model local refinements, replacements, or reversals within a stable topical context.

**Inter-topic transitions.** Some preference updates are expressed in one topical context but change how memories in another topic should be interpreted. For example, a user may initially prefer quick solo workouts because of a busy schedule, but later prefer longer outdoor activities after joining a weekend hiking group. The old state may be grounded under a `fitness-routine` topic, whereas the new state may be grounded under a `social-outdoor-activities` topic. Both endpoints are produced by the same shift detector; the edge is labeled inter-topic only because the grounded endpoints belong to different topic graphs. Thus, inter-topic transitions model preference updates whose consequences cross fine-grained topic boundaries.

Figure 3 evaluates the contribution of these two edge types separately, showing that intra-topic and inter-topic transitions provide complementary retrieval signals.

### B.4 Transition Records

The detector outputs shift events, which are stored as transition records after grounding. Table 8 shows examples of the stored fields.

**Grounding threshold.** We set $\tau_g = 0.50$ and discard a detected update if either endpoint similarity falls below this threshold. This lightweight check requires both extracted states to have sufficiently similar anchors in the existing graph. It can reject extraction hallucinations that lack such anchors without an additional LLM call.

**Illustrative grounding example.** The message "I prefer quick solo workouts" introduces entity nodes such as `User` and `solo workouts`, with a factual edge `User` $\xrightarrow{\text{PREFERS}}$ `solo workouts` in a fitness topic graph. A later message expressing a preference for social outdoor activities is also ingested as factual memory. The shift detector produces a record with $s^- =$ "quick solo workouts" and $s^+ =$ "social outdoor activities". Equation 8 maps these states to the existing `solo workouts` and `social outdoor activities` nodes. If both similarities meet $\tau_g$, we store the record and its directed edge $v^- \to v^+$; the edge is inter-topic when the endpoints belong to different topic graphs. The state descriptions remain in the record rather than becoming new nodes.

Table 8: Examples of transition records produced by the shift detector.

| Type | Old state | New state | Update rationale | Valid time |
|---|---|---|---|---|
| Music preference shift | Engaging with the quirky and humorous side of music | Focusing on emotional and reflective aspects of music | The user moved from lighthearted storytelling to a more introspective music experience. | 2015-03-22 |
| Learning approach shift | Learning through formal, high-density sources | Preferring accessible, informal, discussion-friendly sources | Jargon-heavy material caused confusion, while informal mentoring worked better. | 2018-09-14 |

### B.5 Algorithm Workflow

Algorithms 1 and 2 restate the construction and retrieval stages of TRANSMEM in pseudocode, making the order of operations and all threshold dependencies explicit.

Algorithm 1 processes user history in a single pass. Every message is routed and ingested; messages routed as EVOLVE are added to a candidate-update buffer $\mathcal{B}$ that triggers detection once $\kappa$ candidates

**Algorithm 1** Topic-Structured Preference Graph Construction.

**Require:** user history $m_1, \ldots, m_T$; thresholds $\tau_{\text{macro}}$, $\tau_{\text{micro}}$, $\kappa$; grounding threshold $\tau_g = 0.50$; maturity bound $n_{\text{mat}}$; historical-evidence budget $K_{\text{hist}}$
**Ensure:** topic-structured user memory graph $G_u$

initialize topic set $\{T_i\}$ and candidate-update buffer $\mathcal{B}_u \leftarrow \emptyset$
**for** each incoming message $m_t$ **do**
  compute $S_{\text{macro}}(m_t, T_i)$ for all existing topic graphs $T_i$
  $M_t \leftarrow \max_i S_{\text{macro}}(m_t, T_i)$;  $i^* \leftarrow \arg\max_i S_{\text{macro}}(m_t, T_i)$
  **if** $M_t < \tau_{\text{macro}}$ **then**
    $r_t \leftarrow$ BIRTH;  create a new topic graph and ingest $m_t$
  **else**
    $\mu_t \leftarrow S_{\text{micro}}(m_t, T_{i^*})$;  $z_t \leftarrow \text{mature}(T_{i^*}; n_{\text{mat}})$
    **if** $\neg\, z_t$ **or** $\mu_t \geq \tau_{\text{micro}}$ **then**
      $r_t \leftarrow$ DEEPENING;  ingest $m_t$ into $T_{i^*}$
    **else**
      $r_t \leftarrow$ EVOLVE;  ingest $m_t$ into $T_{i^*}$
      $\mathcal{B}_u \leftarrow \mathcal{B}_u \cup \{m_t\}$
      **if** $|\mathcal{B}_u| \geq \kappa$ **then**
        $\mathcal{H}(\mathcal{B}_u) \leftarrow \arg\text{topK}_{f \in \mathcal{F}_u,\, K_{\text{hist}}}\, s_{\text{hist}}(\mathcal{B}_u, f)$  ▷ retrieve compact historical evidence from the user graph
        $\mathcal{Z} \leftarrow \mathcal{D}_{\text{LLM}}(\mathcal{B}_u, \mathcal{H}(\mathcal{B}_u))$
        **for** each detected update $z = (s^-, s^+, r, g, t_{\text{valid}}) \in \mathcal{Z}$ **do**
          map $s^-, s^+$ to $v^-, v^+ \in V_u$ using Equation 8
          **if** both endpoint similarities are at least $\tau_g$ **then**
            set $y_e = \text{intra}$ if endpoints share a topic, otherwise $y_e = \text{inter}$
            store $e_z = (v^-, v^+, r, y_e, g, t_{\text{valid}})$ and its record $z$
          **else**
            discard $z$
          **end if**
        **end for**
        $\mathcal{B}_u \leftarrow \emptyset$
      **end if**
    **end if**
  **end if**
  periodically refresh affected topic-summary embeddings $s_i$
**end for**
**return** $G_u$

accumulate. For each triggered buffer, TRANSMEM retrieves a compact historical-evidence set $\mathcal{H}(\mathcal{B})$ from the user memory graph before calling the shift detector. This historical-evidence retrieval is capped at $K_{\text{hist}} = 10$ normal facts, so the detector compares the candidate buffer against a small evidence set rather than the full user history. The three pseudocode labels correspond to the routing outcomes of Equation 4: BIRTH instantiates a new topic when the macro gate fails; DEEPENING continues an existing topic when both gates pass; EVOLVE flags an evolution candidate when only the macro gate passes. The shift detector prompt used by $\mathcal{D}_{\text{LLM}}$ is shown in Appendix B.6.

**Maturity bound.** We enable the EVOLVE branch only after the matched topic graph contains at least $n_{\text{mat}}$ ingested messages. This safeguard prevents sparse early topic graphs from producing unstable shift candidates; it does not introduce a separate detector or transition type. In the early phase of a topic, the entity set $K(T_{i^*})$ is small, so the entity-overlap score $\mu_t = S_{\text{micro}}(m_t, T_{i^*})$ is naturally low simply because the topic has accumulated few entities, not because the new message conflicts with the existing state. Without the maturity check, such low-overlap messages would immediately route to EVOLVE and produce spurious candidate updates against an essentially empty topic. Before the topic matures, all topically related messages are therefore routed to DEEPENING; once it matures, the macro–micro routing in Equation 4 applies as written. We adopt $n_{\text{mat}}$=15 and refresh topic summaries after every 10 topic-local deepening messages, counting each message as one update unit. These values are treated as supporting implementation settings rather than as primary

**Algorithm 2** Adaptive Preference Retrieval.

**Require:** query $q$; graph $G_u$; factual-anchor budget $K_A$; continuation threshold $\tau_c$; drop tolerance $\delta_c$; retention $\gamma$; beam size $B$; max hops $H$
**Ensure:** evidence context $\mathcal{C}(q)$

$\mathcal{A}(q) \leftarrow \arg\mathrm{topK}_{f \in \mathcal{F}_{\mathrm{normal}},\, K_A}\, s(q, f)$ ▷ retrieve factual anchors
$\mathcal{N}_0(q) \leftarrow$ transition-edge candidates adjacent to the endpoints of factual anchors in $\mathcal{A}(q)$
**for** each $e \in \mathcal{N}_0(q)$ **do**
    $\sigma_0(e) \leftarrow \cos(E(q), E(e))$
**end for**
$\widetilde{\mathcal{N}}_0(q) \leftarrow \{e \in \mathcal{N}_0(q) : \sigma_0(e) \geq \tau_c\}$
initialize $\mathcal{P}$ from the top-$B$ candidates in $\widetilde{\mathcal{N}}_0(q)$
**if** $\mathcal{P} = \emptyset$ **then**
    **return** $\mathrm{Assemble}(\mathcal{A}(q), \emptyset)$ ▷ no relevant transition found
**end if**
**for** each path $p \in \mathcal{P}$ **do**
    $C^{(p)} \leftarrow E(q)$;  $\sigma^{(p)}_{\mathrm{last}} \leftarrow \sigma_0(e^{(p)}_1)$
    $C^{(p)} \leftarrow \gamma C^{(p)} + (1-\gamma)E(e^{(p)}_1)$
**end for**
$\mathcal{P}_{\mathrm{done}} \leftarrow \emptyset$
**for** $t = 2, \ldots, H$ **do**
    $\mathcal{P}' \leftarrow \emptyset$
    **for** each active path $p \in \mathcal{P}$ **do**
        $\mathcal{N}(p) \leftarrow$ transition-edge candidates adjacent to the current endpoint of $p$
        **for** each candidate next-hop edge $e \in \mathcal{N}(p)$ **do**
            $\sigma(e \mid p) \leftarrow \cos\big(C^{(p)}, E(e)\big)$
            **if** $\sigma(e \mid p) \geq \tau_c \ \wedge\ \sigma(e \mid p) - \sigma^{(p)}_{\mathrm{last}} \geq -\delta_c$ **then**
                extend $p$ to $p'$ with $e$
                $C^{(p')} \leftarrow \gamma C^{(p)} + (1-\gamma)E(e)$
                $\sigma^{(p')}_{\mathrm{last}} \leftarrow \sigma(e \mid p)$
                add $p'$ to $\mathcal{P}'$
            **end if**
        **end for**
        **if** no candidate extension of $p$ is accepted **then**
            $\mathcal{P}_{\mathrm{done}} \leftarrow \mathcal{P}_{\mathrm{done}} \cup \{p\}$
        **end if**
    **end for**
    **if** $\mathcal{P}' = \emptyset$ **then**
        **break**
    **end if**
    $\mathcal{P} \leftarrow \mathrm{TopB}(\mathcal{P}')$
**end for**
$\mathcal{E}_{\mathrm{trans}}(q) \leftarrow$ unique transition records along paths in $\mathcal{P} \cup \mathcal{P}_{\mathrm{done}}$
**return** $\mathrm{Assemble}(\mathcal{A}(q), \mathcal{E}_{\mathrm{trans}}(q))$

experimental axes; sensitivity analyses for the main construction and retrieval hyperparameters are reported in Appendix C.5 and Appendix C.6.

Algorithm 2 implements the factual anchoring followed by adaptive transition expansion described in Section 2.2. After retrieving factual anchors, all transition-edge candidates adjacent to the anchor entities are scored directly against the query embedding; the top-$B$ candidates seed the beam. If no candidate scores above the continuation threshold $\tau_c$, retrieval terminates at the factual anchors. Otherwise, each accepted path extends under a shared continuation rule $(\tau_c, \delta_c)$ and trajectory state updated by the retention factor $\gamma$; beam size $B$ and hop cap $H$ act as second-line safeguards (see Appendix C.6). At each hop, TopB retains up to $B$ paths with the highest latest-step scores $\sigma^{(p)}_{\mathrm{last}}$.

### B.6 Prompt Templates

We include the prompt templates used for transition detection, transition-record quality analysis, final QA on PersonaMem and PersonaBench, and PersonaMem data-quality filtering. Prompts are shown verbatim; line breaks are preserved for readability.

**Shift detector prompt.** $\mathcal{D}_{\text{LLM}}$ (Equation 6) is prompted as follows.

**Shift Detector (Memory Detective)**

You are a specialized Memory Detective. Your goal is to analyze a timeline of user activities and identify **"Shift Events"** — distinct points where the user's preferences, habits, or state **changed** from one to another.

**Input Context (Chronological Facts):**
{sorted_facts}

**Instructions:**

1. Read the timeline to detect contradictions or explicit updates over time.
2. Ignore simple refinements. Detect actual state changes.
3. For each shift, extract: event_type, old_state, new_state, trigger, valid_at (YYYY-MM-DD).
4. If NO significant shifts found, return an empty list.

**Output Format (strict JSON):**

```
{"shifts": [{"event_type": "...", "old_state": "...",
  "new_state": "...", "trigger": "...", "valid_at": "YYYY-MM-DD"}]}
```

**LLM-as-judge prompt for transition-record quality analysis.** The judge prompt used in Appendix C.5 is shown below.

**Transition-Record Judge (GPT-4o)**

**System:** You are evaluating shift records in a personalized lifelong memory system. A shift record claims a user's preference or behavior has CHANGED over time.
Rate VALID if ALL of the following hold:

1. The old state and new state describe meaningfully DIFFERENT preferences or behaviors.
2. The trigger (if present) plausibly explains the change.
3. The new state is specific enough to support future personalized retrieval.

Rate INVALID if ANY of the following hold: old/new states are essentially the same (trivial rewording); change is cessation only with no new preference; trigger does not plausibly cause the change; states are incoherent or self-contradictory; both states are too vague to be useful for retrieval.
Reply with exactly: Line 1: VALID or INVALID. Line 2: one-sentence reason ($\leq$30 words).

**Template:** Shift type: {kind} | Fact: {fact} | Trigger: {trigger} | Source: {src_name} | Target: {tgt_name} | VALID or INVALID?

**Final QA prompt for PersonaMem.** At inference time, the QA backbone is asked to choose one of the four candidate options using the retrieved memory context.

**Final QA Prompt (PersonaMem)**

**Relevant historical information:**
{context_text}

**Question:**
{question}

**Candidate options:**
(a) {option_a}
(b) {option_b}
(c) {option_c}
(d) {option_d}

Find the most appropriate model response and only give your final answer as `<final_answer>(a)</final_answer>`, `<final_answer>(b)</final_answer>`, `<final_answer>(c)</final_answer>`, or `<final_answer>(d)</final_answer>`. Do not provide any explanation.

**Final QA prompt for PersonaBench.** PersonaBench answers are short spans rather than multiple-choice options, so the prompt asks for a minimal answer drawn from the retrieved history.

**Final QA Prompt (PersonaBench)**

**Relevant historical information:**
{context_text}

**Question:** {question}

Answer using only a short answer span copied or normalized from the history when possible. Do not explain. Do not use a full sentence unless the answer itself is naturally a sentence. Do not add extra details, hedging, or commentary. Output only the final answer.

**PersonaMem quality prompt.** We use GPT-4o as a data-quality judge to detect PersonaMem items where the question stem alone reveals the correct option without consulting the user history (Appendix A).

**PersonaMem Quality Prompt**

You are an expert Data Quality Auditor. Your task is to detect "Data Leakage" in multiple-choice questions.

**Definitions:**
**LEAKED (True):** The Question Stem explicitly contains key information (e.g., specific user preferences, past events, names) that directly reveals the Correct Answer. The user does NOT need to recall any past memory to answer this.
**CLEAN (False):** The Question Stem provides context but does NOT give away the answer. The user MUST recall external memory/history to be sure. *CRITICAL:* If the answer can be guessed purely by random chance (25%) or general common sense, it is NOT a leak. Mark it as CLEAN. Only mark LEAKED if the text itself provides the evidence.

**Question:** {question}

**Output Format (strict JSON):**
```
{"reasoning": "Step-by-step analysis. Does the stem mention X?",
 "is_leaked": true or false}
```

# C Additional Experimental Results

## C.1 Full-Context Baseline

FULLTEXT is a prompt-only method that prepends the user's complete conversational history before each query, giving the model direct read access to all past interactions without retrieval or memory management. Table 9 reports its results across all four backbone × context configurations.

Table 9: FULLTEXT results across all configurations. The 128k + Qwen3-8B entry is unavailable because the full history exceeds the model's context limit.

| Context | Model | Revisit Reasons | Track Evo | Latest Prefs | Aligned Recs | New Scenario | Shared Facts | New Ideas | Overall |
|---|---|---|---|---|---|---|---|---|---|
| 32k | Qwen3-8B | 62.96 | 64.00 | 47.06 | 42.59 | 53.06 | 46.02 | 10.23 | 45.47 |
| | Qwen3.5-27B | 90.74 | 77.00 | 58.82 | 61.11 | 77.55 | 69.03 | 27.27 | 65.05 |
| 128k | Qwen3-8B | *— (exceeds backbone context limit)* | | | | | | | |
| | Qwen3.5-27B | 89.24 | 77.73 | 60.81 | 52.58 | 76.61 | 71.43 | 33.33 | 59.60 |

FULLTEXT with Qwen3.5-27B achieves the strongest results on *Revisit Reasons* and *Track Evolution*, where direct access to the complete record eliminates retrieval noise. However, every query requires forwarding the full history through the model; token consumption scales linearly with history length, and the 128k + Qwen3-8B configuration is infeasible because the model's 32k context limit is exceeded. As user histories grow longer, this hard limit becomes increasingly restrictive.

## C.2 By-Task Results for TRANSMEM

Tables 10 and 11 report per-task accuracy of TRANSMEM across both backbones. Topic abbreviations: *Food* (foodRecommendation), *Home* (homeDecoration), *Medical* (medicalConsultation), *Travel* (travelPlanning), *Legal* (legalConsultation), *Financial* (financialConsultation), *Dating* (datingConsultation), *Music* (musicRecommendation), *Book* (bookRecommendation), *Study* (studyConsultation), *Movie* (movieRecommendation), *Family* (familyRelations), *Therapy* (therapy), *Shopping* (onlineShopping), *Sports* (sportsRecommendation). The 128k benchmark adds *Shopping* and *Sports* not present in the 32k partition. Gains are not concentrated in a single domain; the magnitude of improvement varies

across topics, consistent with differences in preference-update density and temporal disambiguation difficulty.

Table 10: Per-task accuracy of TRANSMEM in the 32k context regime. $n$ = evaluation queries per topic.

| Topic | $n$ | Qwen3-8B | Qwen3.5-27B |
|---|---|---|---|
| Food | 2 | 100.00 | 100.00 |
| Home | 2 | 100.00 | 100.00 |
| Medical | 14 | 92.86 | 92.86 |
| Legal | 28 | 64.29 | 75.00 |
| Travel | 60 | 60.00 | 70.00 |
| Study | 29 | 58.62 | 62.07 |
| Dating | 67 | 58.21 | 70.15 |
| Movie | 84 | 57.14 | 71.43 |
| Book | 58 | 56.90 | 67.24 |
| Music | 45 | 55.56 | 60.00 |
| Therapy | 2 | 50.00 | 50.00 |
| Financial | 60 | 45.00 | 73.33 |
| Family | 24 | 37.50 | 54.17 |
| *Overall* | 475 | *56.84* | *69.26* |

Table 11: Per-task accuracy of TRANSMEM in the 128k context regime. $n$ = evaluation queries per topic.

| Topic | $n$ | Qwen3-8B | Qwen3.5-27B |
|---|---|---|---|
| Home | 111 | 58.56 | 63.96 |
| Study | 131 | 58.02 | 74.81 |
| Family | 117 | 55.56 | 70.94 |
| Travel | 126 | 53.97 | 65.87 |
| Music | 38 | 52.63 | 73.68 |
| Medical | 160 | 51.25 | 73.12 |
| Shopping | 164 | 51.22 | 56.10 |
| Movie | 75 | 49.33 | 70.67 |
| Book | 125 | 47.20 | 72.00 |
| Dating | 170 | 46.47 | 63.53 |
| Financial | 175 | 45.71 | 66.29 |
| Legal | 256 | 45.70 | 60.94 |
| Food | 198 | 41.41 | 60.61 |
| Sports | 114 | 41.23 | 56.14 |
| Therapy | 201 | 38.31 | 59.20 |
| *Overall* | 2161 | *48.03* | *64.69* |

### C.3 Various Retrieval Strategies

We include two inference-only controls to separate the effect of transition-aware retrieval from simpler context-expansion alternatives. Both controls reuse the same constructed memory and final QA prompt as TRANSMEM, but replace adaptive transition expansion with a simpler retrieval strategy.

The three expansion-control variants compare as follows:

- CHRONOEXPAND: factual anchors plus chronological normal-fact neighbors.
- FLATTRANS: factual anchors plus flat retrieved transition records.
- TRANSMEM: factual anchors plus graph-grounded adaptive transition expansion.

**ChronoExpand.** CHRONOEXPAND starts from the same factual anchors as TRANSMEM. For each anchor fact, it locates the anchor in the same topic's fact timeline (sorted by `valid_at`/`created_at`), retrieves the $h$ nearest normal facts on each side, and merges the result into the final context under the same token budget. It does not use transition records and does not follow transition endpoints. This tests whether nearby temporal context alone can explain the gains of TRANSMEM.

For example, if an anchor fact states that the user recently joined a hiking group, CHRONOEXPAND may also retrieve the immediately preceding and following normal facts in the same outdoor-activity timeline. It does not know whether those neighboring facts describe a preference update or merely adjacent context.

We select $h$ on a 20-user diagnostic subset (257 queries) using the same procedure as inference hyperparameter search (Appendix C.6). Table 12 reports overall accuracy for $h \in \{1, 2, 3\}$.

Table 12: CHRONOEXPAND overall accuracy on the diagnostic subset (20 users, 257 queries) for different chronological window sizes.

| Window $h$ | Overall Acc. (%) |
|---|---|
| 1 | 55.64 |
| **2** | **56.03** |
| 3 | 53.78 |

$h$=2 yields the highest accuracy and is used for all CHRONOEXPAND comparisons in Section 3.5. Wider windows can add temporally adjacent but answer-irrelevant normal facts, diluting the answer-bearing anchor for certain query types.

**FlatTrans.** FLATTRANS starts from the same factual anchors as TRANSMEM. It then collects all transition records associated with the user's active topic groups—both intra-topic transition edges and inter-topic transition edges—and ranks them as flat documents by cosine similarity between the query embedding and each transition record embedding. The top-$K_T$ transition records are merged with the factual anchors under the same final context budget and QA prompt as TRANSMEM. Unlike TRANSMEM, FLATTRANS does not follow transition endpoints, does not maintain a trajectory state, and does not adaptively expand across hops. This tests whether transition text is sufficient once factual anchors are fixed, independently of graph-grounded endpoint traversal.

For example, if the query asks for a current cultural recommendation, FLATTRANS may retrieve transition records mentioning cultural participation, but it does not require those records to be reachable from the factual anchors or to form a coherent endpoint-grounded update path.

On Qwen3-8B, FLATTRANS reaches 54.32% (32k) and 46.92% (128k), which are the values reported alongside CHRONOEXPAND and TRANSMEM in Section 3.5.

### C.4 Adaptive Retrieval: Per-Category Gain and Hop Statistics

This section reports a finer-grained breakdown of where adaptive trajectory expansion helps. For both Qwen3-8B and Qwen3.5-27B in the 128k setting, we compare the *Facts Only* condition (factual anchor retrieval only, no expansion) with the full TRANSMEM model and report the average adaptive hop depth per query category.

**Qwen3-8B (128k).** Table 13 provides the values underlying the main-text comparison in Figure 5. Overall accuracy increases from 46.00% to 48.03%, with an average expansion depth of 1.92 hops.

Table 13: Per-category *Facts Only* vs. *Full* TRANSMEM with average adaptive hop depth (128k, Qwen3-8B).

| Category | Facts Only | Full | Δ | Avg Hops |
|---|---|---|---|---|
| Revisit | 46.84 | 51.90 | **+5.06** | 2.17 |
| Track Evo | 51.82 | 55.00 | **+3.18** | 2.51 |
| Latest Prefs | 53.12 | 56.75 | **+3.63** | 1.90 |
| Aligned Recs | 51.29 | 51.29 | 0.00 | 1.72 |
| New Scenario | 47.37 | 53.22 | **+5.85** | 1.81 |
| Shared Facts | 57.79 | 62.99 | **+5.20** | 1.89 |
| New Ideas | 24.40 | 21.13 | −3.27 | 1.77 |
| *Overall* | *46.00* | *48.03* | *+2.03* | *1.92* |

**Qwen3.5-27B (128k).** On the stronger backbone, adaptive expansion yields positive deltas across all seven categories (Figure 6 and Table 14). Gains are largest for *New Ideas* (+3.48 pp), *Shared Facts* (+3.24 pp), and *Track Evolution* (+2.27 pp). As at 8B, *Track Evolution* and *Revisit* trigger the deepest average expansions (2.24 and 1.92 hops), although the largest accuracy gain occurs on *New Ideas*. The 27B per-category pattern is broadly consistent with the 8B pattern for trajectory-sensitive categories, and the regression on *New Ideas* observed at 8B does not appear at 27B, suggesting that the stronger backbone handles open-ended trajectory context more selectively.

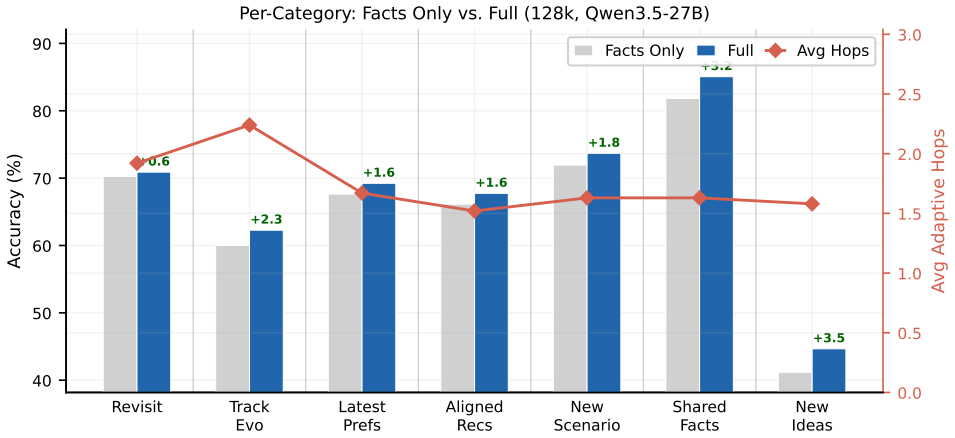


Figure 6: Per-category comparison of *Facts Only* versus *Full* TRANSMEM in the 128k setting (Qwen3.5-27B). Bars show accuracy; the right axis shows the average adaptive hop depth of the full model.

Table 14: Per-category *Facts Only* vs. *Full* TRANSMEM with average adaptive hop depth (128k, Qwen3.5-27B).

| Category | Facts Only | Full | Δ | Avg Hops |
|---|---|---|---|---|
| Revisit | 70.25 | 70.89 | **+0.64** | 1.92 |
| Track Evo | 60.00 | 62.27 | **+2.27** | 2.24 |
| Latest Prefs | 67.63 | 69.23 | **+1.60** | 1.67 |
| Aligned Recs | 66.13 | 67.74 | **+1.61** | 1.52 |
| New Scenario | 71.93 | 73.68 | **+1.75** | 1.63 |
| Shared Facts | 81.82 | 85.06 | **+3.24** | 1.63 |
| New Ideas | 41.18 | 44.66 | **+3.48** | 1.58 |
| *Overall* | *62.56* | *64.69* | *+2.13* | *1.74* |

### C.5 Construction-Stage Hyperparameter Selection

This section reports three empirical studies on the construction stage: a validation of the topic router and LLM shift detector, a sensitivity analysis of the primary construction hyperparameters, and a quality analysis of the stored transition records. All studies use the same 20-user diagnostic subset (257 queries) from PersonaMem 32k. We first fix the construction parameters and then select the retrieval parameters on the same diagnostic subset.

**Topic router and LLM shift detector.** TRANSMEM maintains a per-user memory graph organized into fine-grained topic subgraphs, isolating preference facts from semantically distinct life areas. The topic router provides the provisional context for candidate shifts, while the shift detector compares buffered activity against a compact chronological evidence set that may include relevant facts outside the current topic. To evaluate the contribution of each component, we compare four configurations that add topic structure and transition edges incrementally.

The results in Table 15 indicate that each component provides a positive incremental gain in this study and the combined system performs best. Adding transition edges to a single-topic graph

Table 15: Validation of topic routing and transition edges (20-user diagnostic subset, 257 queries). Overall accuracy (%).

| Configuration | Topic Router | Trans. Edges | Acc (%) |
|---|---|---|---|
| Single topic, no transitions, no router | ✗ | ✗ | 51.75 |
| Single topic + transition edges (intra only) | ✗ | ✓ | 54.86 |
| Multi-topic router, no transitions | ✓ | ✗ | 55.64 |
| Full TRANSMEM (multi-topic + transitions) | ✓ | ✓ | **57.59** |

provides $+3.11$ pp, suggesting that the LLM detector can identify useful preference-update signals even without per-topic isolation. Adding the topic router without transition edges provides $+3.89$ pp over the flat baseline, consistent with per-topic isolation reducing inter-topic interference in factual retrieval. The full system, combining both components, achieves the highest accuracy ($+5.84$ pp over the flat baseline).

**Construction hyperparameter sensitivity.** The construction stage is governed by three hyperparameters: the macro routing threshold $\tau_{\text{macro}}$, the micro entity-consistency threshold $\tau_{\text{micro}}$, and the shift-detection buffer size $\kappa$. Due to the substantial offline cost of rebuilding topic graphs, this study uses the fixed 20-user diagnostic subset.

Table 16 reports the key configurations. A higher $\tau_{\text{macro}}$ raises the bar for assigning a message to an existing topic, increasing BIRTH (topic creation) events and producing a more fragmented topic graph; this risks isolating semantically related facts into separate topics. A lower $\tau_{\text{micro}}$ suppresses evolution candidates and reduces the number of transition edges, which in turn limits the recall of preference-update signals available to the retriever. With both routing thresholds fixed, buffer sizes $\kappa$=5 and $\kappa$=20 yield the same accuracy. We adopt $\tau_{\text{macro}}$=0.65, $\tau_{\text{micro}}$=0.25, and $\kappa$=5.

Table 16: Construction hyperparameter sensitivity (20-user diagnostic subset).

| $\tau_{\text{macro}}$ | $\tau_{\text{micro}}$ | $\kappa$ | Acc (%) |
|---|---|---|---|
| **0.65** | **0.25** | **5** | **56.81** |
| 0.70 | 0.25 | 5 | 54.47 |
| 0.65 | 0.20 | 10 | 56.42 |
| 0.65 | 0.25 | 20 | 56.81 |

**Transition-Record Quality Analysis.** To assess the semantic quality of transition records produced by the shift detector, we sample stored records from the frozen PersonaMem 32k memory graphs and evaluate them with GPT-4o as an external judge. This analysis focuses on whether stored records are meaningful and usable for retrieval, rather than estimating exhaustive recall over all possible preference updates in the raw history. The judge prompt is shown in Appendix B.6.

*Setup.* We randomly sample 100 intra-topic and 100 inter-topic transition edges (200 total) with a fixed random seed and present each edge to GPT-4o as an independent judge. An edge is rated VALID only if (1) old and new preference states are meaningfully different, (2) the trigger plausibly explains the change, and (3) the new state is specific enough to support future personalized retrieval. An edge is INVALID if the change is trivial rewording, cessation-only without a replacement preference, based on an implausible trigger, incoherent, or too vague.

*Results.* Table 17 reports the outcome.

Table 17: LLM-judged quality of stored transition records.

| Record type | Sampled | Valid rate |
|---|---|---|
| Intra-topic | 100 | 98.0% |
| Inter-topic | 100 | 99.0% |
| Total | 200 | 98.5% |

The high valid rate indicates that most stored transition records encode a meaningful old-to-new preference update with a plausible update rationale. The three INVALID cases share the same failure mode: the detector promotes a cessation event into a transition record without a replacement preference, so the target entity does not encode a useful new preference state.

*Positive example.* User `aa95cf`, intra-topic (`craft supplies` → `crafting event`, 2015-03-23): *"Crafting Preference Shift: Disliking crafting and finding it tedious → Appreciating crafting as a therapeutic and creative activity (Trigger: Joining a DIY crafting group and creating personalized gifts)."* Verdict: VALID. The before/after states are clearly distinct, the trigger is concrete, and the new state is specific enough to guide future recommendations.

*Negative example.* User `373af9`, intra-topic (`first aid techniques` → `anxiety`, 2015-01-21): *"Health Knowledge Shift: Interested in learning first aid → Discontinued pursuit of medical knowledge due to stress and anxiety."* Verdict: INVALID. The edge describes cessation without encoding a new preference direction; the target entity `anxiety` is not actionable for personalized retrieval.

### C.6 Inference-Stage Hyperparameter Selection

We study the main inference-stage hyperparameters of TRANSMEM on a randomly selected sample of 20 users from PersonaMem 32k, comprising 257 evaluation queries. To keep the search tractable, the retrieval configuration is tuned sequentially: we first vary the factual-anchor budget (Equation 10), then the trajectory-retention factor $\gamma$ (Equation 13), and finally the continuation/stopping rule $(\tau_c, \delta_c)$ (Equation 12). The resulting configuration is used for all main full-evaluation experiments. For the 128k setting, the trajectory-related configuration chosen from this study is carried over, and only the factual-anchor budget is retuned to account for longer histories.

**Factual-anchor budget (top-$k$).** Figure 7 shows the accuracy curves for varying $k$ (Equation 10), with separate panels for Qwen3-8B ($k \in \{10, \ldots, 80\}$) and Qwen3.5-27B ($k \in \{30, \ldots, 110\}$) since the optimal range differs by backbone. For Qwen3-8B, accuracy improves up to $k$=50 (56.81%), with $k$=60 matching this peak before declining; we adopt $k$=50 as the smaller budget that matches the peak. For Qwen3.5-27B, the curve peaks at $k$=90 (72.37%), and we adopt this value directly.

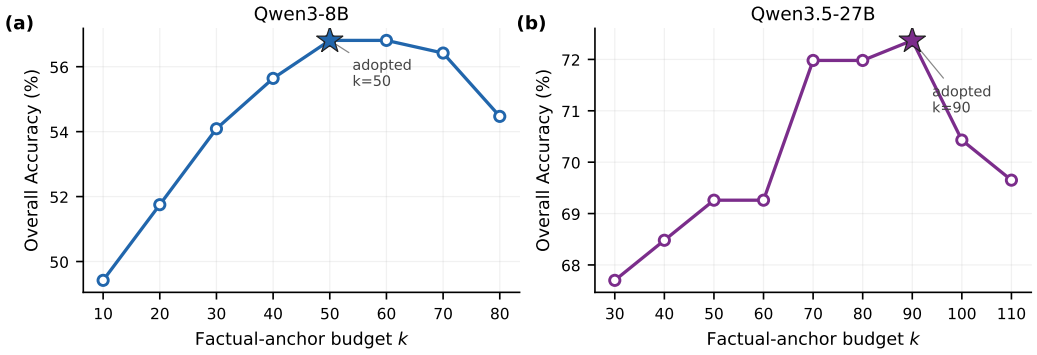


Figure 7: Sensitivity to the factual-anchor budget $k$. **(a)** Qwen3-8B; **(b)** Qwen3.5-27B. The adopted value is marked with a star.

**Trajectory-retention factor ($\gamma$).** The trajectory update in Equation 13 can be unrolled as

$$C_t = \gamma^t E(q) + (1-\gamma) \sum_{j=1}^{t} \gamma^{t-j} E(e_j).$$

Thus, the score of a next-hop candidate is evaluated against a mixture of the original query and the accepted transition path, with exponentially decaying weight on earlier accepted transitions. At $\gamma = 1$, the trajectory state remains equal to the query embedding, so transition scoring is query-conditioned while graph traversal remains path-dependent. When $\gamma = 0$, the state is overwritten by the most recent transition, making expansion more vulnerable to local drift. Intermediate values preserve query relevance while allowing accepted transition records to steer later hops, which is

consistent with the sensitivity results in Figure 8. The results show accuracy as $\gamma$ varies across $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. At $\gamma$=0, the trajectory state is overwritten by the most recently accepted edge, discarding all accumulated path information. Intermediate values outperform both extremes: the 8B curve peaks at $\gamma$=0.80 (57.20%) and the 27B curve at $\gamma$=0.60 (73.54%). This is consistent with the benefit of balancing accumulated preference context against newly accepted transition records at each hop. We adopt $\gamma$=0.80 for Qwen3-8B and $\gamma$=0.60 for Qwen3.5-27B.

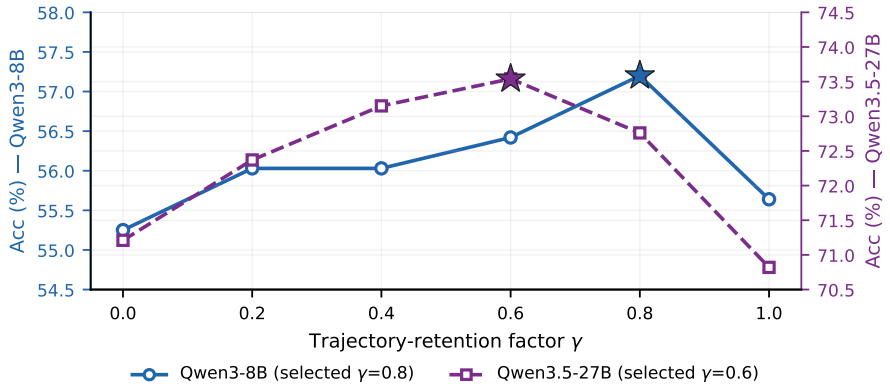


Figure 8: Sensitivity to the trajectory-retention factor $\gamma$. Two $y$-axes are used because the two backbones differ by roughly 17 pp on this subset. The adopted value for each backbone is marked with a star.

**Continuation and drop tolerance ($\tau_c$, $\delta_c$).** We evaluate four configurations spanning two levels of continuation threshold $\tau_c$ and drop tolerance $\delta_c$ (Equation 12). A lower $\tau_c$ is easier for a candidate edge to satisfy, allowing the retriever to continue to additional hops. A lower $\delta_c$ enforces stricter tolerance against abrupt score drops, guarding against trajectory drift. Table 18 reports the results. The configuration $\tau_c$=0.40, $\delta_c$=0.10 performs best across both backbones: the permissive continuation threshold encourages multi-hop traversal while the strict drop tolerance prevents the trajectory from drifting toward unrelated edges.

Table 18: Continuation/stopping sensitivity (Equation 12). Values are overall accuracy (%). Best configuration in bold.

| Backbone | ($\tau_c$, $\delta_c$) | Acc (%) |
|---|---|---|
| Qwen3-8B | (**0.40**, **0.10**) | **57.59** |
| | (0.50, 0.10) | 56.81 |
| | (0.50, 0.20) | 54.86 |
| | (0.40, 0.20) | 52.14 |
| Qwen3.5-27B | (**0.40**, **0.10**) | **72.37** |
| | (0.50, 0.20) | 71.98 |
| | (0.50, 0.10) | 71.60 |
| | (0.40, 0.20) | 69.26 |

**Beam size and hop cap.** Adaptive continuation in Algorithm 2 is controlled by $(\tau_c, \delta_c)$, while beam size $B$=3 and hop cap $H$=6 bound search width and depth.

**Final retrieval configuration and 128k transfer.** The adopted 32k configuration is $k$=50, $\gamma$=0.80, $\tau_c$=0.40, $\delta_c$=0.10 for Qwen3-8B, and $k$=90, $\gamma$=0.60, $\tau_c$=0.40, $\delta_c$=0.10 for Qwen3.5-27B. For the 128k setting, the trajectory configuration is carried over and only the factual-anchor budget is retuned, yielding $k$=90 for Qwen3-8B and $k$=110 for Qwen3.5-27B.

### C.7 Case Analysis

The categories where TRANSMEM is not consistently strongest tend to involve either locally answerable latest-state lookup or open-ended ideation. In the former case, a strong note- or profile-based memory baseline can succeed by directly retrieving the most recent surface fact, while transition expansion may add little if the question does not require resolving a supersession relation. In the latter case, performance depends not only on memory recovery but also on the backbone's ability to creatively extrapolate from retrieved evidence. These cases suggest that transition records are most beneficial when the answer depends on determining which prior state has been replaced, rather than when the task is direct lookup or unconstrained generation.

We illustrate when adaptive transition expansion helps and when it can introduce noise through four PersonaMem cases.

**Case 1: Aligned Recommendation** (`856b`)

**Query.** Recommend a local weekend event that is both cultural and interactive.
**Options.** (a) interactive boutique art festival; (b) refusal; (c) literary festival; (d) food/drink tasting.
**Outcome.** Gold `(a)`; factual-anchor-only `(c)`; TRANSMEM `(a)`.
**Representative factual anchors.**

- Conducted an insightful interview with a local author.
- Is involved in organizing community events.
- Personalized recommendations aim to create engaging interaction.

**Accepted transition records.**

- *Passive literary engagement → active literary-community participation* (intra-topic; rationale: national literature festival).
- *Reading for personal enjoyment → engaging with literature through interviews and creative-process exploration* (intra-topic; rationale: interview with a local author).
- *Passive literature consumption → active participation in literary communities and discussions* (inter-topic; rationale: book-signing events at indie bookstores).

*Interpretation.* The factual anchors make the literary option plausible but remain isolated: they do not specify whether the user's current preference is passive literary attendance or participatory cultural interaction. The accepted transition records shift the evidence toward active community-facing cultural participation, which is better aligned with the interaction-heavy art festival. These records provide complementary evidence for the preference update.

**Case 2: Preference-Arc Recovery** (`7d41`)

**Query.** Inspired by a travel memoir, the user restarted emotionally rich travel journaling. Which option best describes the attitude toward travel?
**Options.** (a) *disliking→liking→liking*; (b) *loving→disliking→liking*; (c) *no prior interest→liking→liking*; (d) *liking→disliking→liking*. *Note: the original answer choices are long; we show the preference-trajectory keywords.*
**Outcome.** Gold `(d)`; factual-anchor-only `(b)`; TRANSMEM `(d)`.
**Representative factual anchors.**

- Started a travel journal to capture emotions and stories from adventures.
- Experiences anxiety when thinking about organizing a trip.
- Travel no longer brings the same excitement.

**Accepted transition record.**

- *Enjoying travel planning and exploring new places → experiencing anxiety and difficulty with travel planning* (intra-topic; rationale: rising pressure from cooking and surprise-trip planning).

*Interpretation.* The anchors expose current journal interest and earlier anxiety, while the transition explicitly links earlier enjoyment of travel planning to later anxiety. Together, they support the *liking→disliking→liking* arc in `(d)`. The factual-anchor-only model instead selects `(b)`, which overstates the initial preference as *loving* rather than *liking*.

**Case 3: Expansion Noise on Fact Recall** (da4a)

**Query.** What are some fun ways to bring family together during the weekends?
**Options.** (a) family movie night; (b) garden project; (c) cooking challenge; (d) board-game tournament.
**Outcome.** Gold (d); factual-anchor-only (d); adaptive TRANSMEM (8B) (a).
**Representative factual anchor.**

- Games are chosen as one of the activities to bring the family together.

**Accepted transition record (most salient).**

- *Awkward campfire gatherings* → *family movie nights* (intra-topic; rationale: movie nights felt more comfortable and enriching).

*Interpretation.* The factual anchors support the gold board-game option. The accepted transition adds a competing "family movie nights" signal, and the 8B QA model selects that option after expansion. Qwen3.5-27B answers correctly from the same context, suggesting that the backbone affects how the retrieved evidence is used.

**Case 4: Over-Grounding on Open-Ended Idea Generation** (0787)

**Query.** What new creative writing pursuits could I explore to further engage with other readers and reflect on literary works?
**Options.** (a) blog posts / literature podcast; (b) literary salons at home; (c) online writers' workshops; (d) poetry reading circle.
**Outcome.** Gold (a); adaptive TRANSMEM (8B) (b).
**Representative factual anchors.**

- Enjoys reading literary fiction.
- Values reflecting on books with others.

**Accepted transition records.**

- *Reading for personal enjoyment* → *engaging with literature through interviews and creative-process exploration* (intra-topic; rationale: insightful interview with a local author).
- *Passive consumption of literature* → *active participation in literary communities and discussions* (inter-topic; rationale: book-signing events at indie bookstores).

*Interpretation.* The transition records are coherent and locally faithful to the user's literary trajectory. For an open-ended idea-generation query, however, they over-ground the model in familiar in-person literary-community formats, biasing the 8B backbone toward a conservative preference-consistent continuation (literary salons) rather than the more creative outward-facing formats in the gold answer. This case supports the main-text observation that *New Ideas* does not consistently benefit from deeper retrieval. Partly off-topic retrieved context may also contribute to this error.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the main contributions and scope of the work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: This paper includes a Limitations section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical results, theorems, or formal proofs. The equations in Section 2 define the proposed retrieval and scoring procedure rather than theoretical guarantees.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper provides implementation settings, prompts, hyperparameters, and evaluation details in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code and data are available at `https://github.com/Aries-chen/TransMem`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We cover all the details in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: We report full aggregate results across benchmarks, backbones, context regimes, and controlled analyses, but do not include error bars or formal statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware is described in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available or benchmark-provided data and does not involve collecting new human-subject data. We discuss limitations and privacy risks of long-term personal memory in the Limitations section.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The work may improve long-term personalized assistants, and the Limitations section discusses risks around sensitive, stale, or manipulated personal memories. We also discuss user inspection, correction, deletion, access control, and user-controlled storage.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release a new high-risk model, scraped dataset, or other asset requiring special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We cite the benchmarks, models, and baseline implementations used in this work. Appendix B credits the modified Graphiti engine (Apache-2.0); the repository retains third-party notices and the separate PersonaMem data license (MIT).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The repository documents installation, configuration, data provenance, and licenses for the TRANSMEM implementation and processed PersonaMem subsets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve new crowdsourcing experiments or new human-subject data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve new crowdsourcing experiments or new human-subject data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer:[Yes]

    Justification: The proposed method uses backbone LLMs for offline construction and final answer generation, as described in Sections 2 and 3.1.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.