PreCoMem: Predictive Cognitive Memory for Self-Evolving Long-Term Dialogue Agents

Jian Zhong[1*] **Zeyu Liu**[1*] **Pingchuan Cao**[1] **Rongduo Han**[1] **Shunye Tang**[1]
Guohuan Xie[3] **Mingyuan Qin**[4] **Ran Ji**[5] **Xiao Liang**[2] **Wei Wang**[1†] **Haining Zhang**[1†]

[1]Nankai University [2]vivo AI Lab [3]Tsinghua University
[4]Huazhong University of Science and Technology
[5]University of the Chinese Academy of Sciences
zhongjian@mail.nankai.edu.cn

Abstract

Long-horizon dialogue agents need to keep user memories stable enough for personalization while adapting when the user genuinely changes. Existing memory systems often append every observation or overwrite old entries by heuristic rules, which makes transient noise hard to distinguish from real belief drift. We introduce **PreCoMem**, a memory consolidation framework inspired by predictive coding that represents memory as confidence-weighted beliefs about the user's state. Its core signal, *Effective Surprisal*, combines semantic deviation and contradiction evidence, then downweights this signal when retrieval is diffuse and unreliable. According to effective surprisal, a two-threshold gate maps each turn to one of three deterministic updates: `MAINTAIN` reinforces supported beliefs, `PROFILE` stores ambiguous cues as low-confidence hypotheses, and `CORRECT` softly decays reliably contradicted facts. The `PROFILE` stage acts as an explicit waiting room, preventing premature commitment to noisy evidence. Experiments on LoCoMo, LongMemEval-S, and PersonaMem-v2 show consistent state-of-the-art accuracy; our new DiMoBench confirms that PreCoMem adapts to non-stationary belief drift without over-reacting to noise. The code is available at `https://github.com/ZhongWishing/PreCoMem`.

1 Introduction

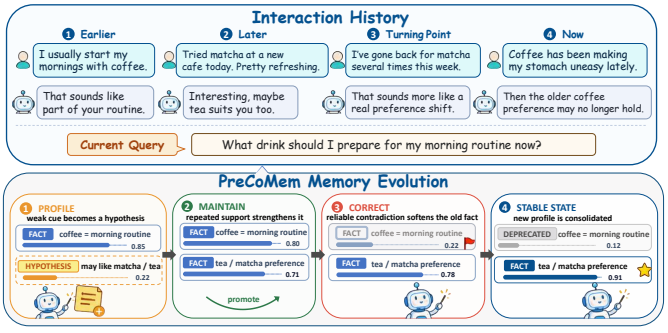

Figure 1: **Memory evolution case study (coffee → tea).** PreCoMem autonomously manages the lifecycle of facts and hypotheses, achieving a stable transition between conflicting belief states without manual intervention.

*Equal contribution.
†Corresponding authors.

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

Large Language Models (LLMs) are moving from stateless responders toward long-term partners that interact with the user over hundreds or thousands of turns [1–3]. Such partners need more than stored transcripts; they require a user model to sustain personalized interaction [4, 5]. To formalize the core units of this user model, we define a *belief* as a compact, actionable claim about the user's preferences, goals, or life context, paired with a confidence weight that can be reinforced or revised over time. This user model underpins both agent personalization and behavioral self-correction, thereby redefining the role of long-term memory: rather than merely serving as simple context extension [6, 7], long-term memory enables the agent to maintain consistent behavior across sessions.

The harder question is not *whether* information can be stored, but *how* it should be consolidated and revised. A practical memory consolidation mechanism must decide which observations to retain, how strongly to trust them, and when to maintain, revise, or correct stored items without explicit supervision. We refer to this as the **memory stability–plasticity problem**: the agent should preserve stable user traits while remaining plastic enough to adapt when those traits change [8]. Long-horizon interaction makes the problem hard for three coupled reasons. First, most turns carry little signal, while a small fraction may contain cues indicative of more persistent user states. Second, user preferences and life facts evolve over time [9], so the system must adapt under *non-stationarity and belief drift* [10, 11]. Third, corrective feedback is rarely explicit [12], and noisy retrieval over long contexts further blurs the picture [13–16]. Therefore, the brittle memory update mechanism causes two opposite failures: premature correction from noise, or persistent over-reliance on outdated memories, which we call *old-memory pull*, a phenomenon related to loss of plasticity and retrieval interference in continual adaptation systems [8, 17].

Recent memory systems have made impressive progress on retrieval and storage organization. Hierarchical and OS-style stores [6, 18, 19], structured indices with timestamps or topic shards [20–23], and agentic retrievers [24, 25] all sharpen *what* the system can recall. However, the question of *when* to update is still often handled by recency overwriting, fixed-threshold importance, or one-shot append rules [4, 26]. These triggers cannot reliably separate persistent drift from transient noise [17], so they collapse the stability–plasticity trade-off precisely where long-horizon dialogue stresses it most. A fuller literature discussion is provided in Appendix Section A.

We draw inspiration from *predictive coding*, which posits that learning is driven by mismatch between expectation and reality, i.e., prediction error [27, 28]. In information theory the same quantity is called *surprisal* [29, 30]. We use this view as a design principle for the update path rather than as a claim that dialogue memory is a fully specified probabilistic model. The resulting criterion is simple: events consistent with current memory should be reinforced, while surprising events may require learning or correction [31, 32]. Crucially, surprisal must be paired with a measure of *retrieval reliability*, so that update decisions stay conservative when recall itself is uncertain or noisy [13, 33].

Building on this view, we propose **PreCoMem (Predictive Cognitive Memory)**, a self-evolving long-term memory framework for conversational LLM agents. PreCoMem operates in two active phases over a shared memory store. In *memory leveraging*, it retrieves confidence-weighted beliefs to condition response generation. In *memory evolution*, it compares new observations with recalled evidence and computes *Effective Surprisal* S_{eff}, a reliability-adjusted mismatch score. A two-threshold gate then routes each update to `MAINTAIN`, `PROFILE`, or `CORRECT`, corresponding to consistent, ambiguous, and reliably contradictory evidence. Figure 1 illustrates this loop on a coffee-to-tea preference shift, where a low-confidence hypothesis is gradually promoted to a fact while the contradicted old fact is softly decayed.

In summary, we (i) propose **PreCoMem**, a memory consolidation framework that casts long-term memory as a dynamic prediction of the user's state; (ii) introduce a *reliability-weighted surprisal signal* that gates each update through `MAINTAIN`, `PROFILE`, or `CORRECT`; and (iii) evaluate on LoCoMo, LongMemEval-S, PersonaMem-v2, and DiMoBench/DiMoBench-XL, controlled write-path stress tests with gradual drift and decoy retrieval. PreCoMem outperforms the strongest contemporary baselines while keeping effective memory stable across 1050 turns.

2 PreCoMem

2.1 Framework overview

As illustrated in Figure 2, PreCoMem operates as a closed loop of two active phases around a shared memory store. **Memory Leveraging** retrieves confidence-weighted beliefs and uses them to condition response generation. **Memory Evolution** then revises the memory state in light of

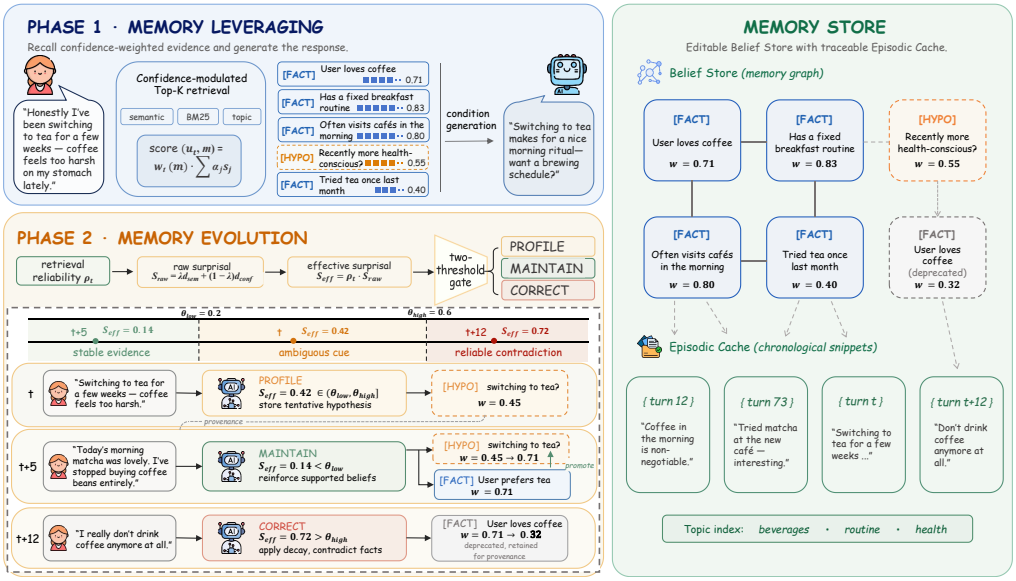

Figure 2: **The PreCoMem framework.** Two active phases operate over a shared memory store. **Phase 1, Memory Leveraging**: a confidence-modulated retrieval score combines semantic, lexical, and topic signals to select Top-K beliefs $\mathcal{C}_t$ for response generation. **Phase 2, Memory Evolution**: the effective surprisal S_{eff} and a two-threshold gate route each turn to `MAINTAIN`, `PROFILE`, or `CORRECT`, illustrated by representative low/medium/high-S_{eff} turns. **Memory Store**: an editable Belief Store $\mathcal{B}$ of `FACT`/`HYPO` nodes with confidence weights, linked by provenance to an immutable Episodic Cache $\mathcal{E}$ under a shared topic index; belief items are updated independently, and the drawn links are provenance pointers and topic groupings (Appendix Section C.11).

the new interaction: it computes *Effective Surprisal* S_{eff} against recalled evidence, and routes the update to one of three deterministic operators—`MAINTAIN`, `PROFILE`, or `CORRECT`. This read–write loop differs from prior memory systems in two ways. First, every belief item carries a continuous confidence weight $w_t(m) \in [0, 1]$ that is reinforced or softly decayed by the operators, instead of relying on discrete commit decisions such as ADD/IGNORE/UPDATE in O-Mem [21] or binary expiry intervals in EverMemOS [19]. Second, `PROFILE` buffers ambiguous observations as low-confidence hypotheses, letting weak evidence accumulate across turns before it is committed or discarded.

In implementation, each operator pairs a closed-form update rule with a fixed-prompt LLM sub-agent. The sub-agent only identifies which items are relevant and emits one bounded scalar, such as EVIDENCE_STRENGTH, SEVERITY, or INITIAL_CONFIDENCE. The deterministic layer then plugs these scalars into Equations (7), (8) and (10). Thus the LLM proposes local magnitudes, while the gate, confidence bounds, and closed-form w updates control when and how memory changes accumulate. Algorithm 1 (Appendix Section B.4) gives the per-turn pipeline.

2.2 Memory primitives

Dialogue and memory state. At turn $t = 1, 2, \ldots$, the user input is u_t, the agent response is y_t, and Q_{t-1} is a rolling working-memory window. Long-term memory is $\mathcal{M}_t = (\mathcal{B}_t, \mathcal{E}_t, \mathcal{T}_t)$. The belief store $\mathcal{B}_t$ contains editable facts and hypotheses with confidence weights. The episodic cache $\mathcal{E}_t$ stores immutable raw dialogue traces with source pointers. The optional topic table $\mathcal{T}_t$ is only a retrieval prior. This keeps beliefs editable and evidence traceable.

Items, confidence, and hypothesis lifecycle. A memory item m has content $c(m)$, type $\tau(m) \in \{\texttt{FACT}, \texttt{HYPOTHESIS}\}$, and retrieval metadata such as embeddings, topic labels, entities, timestamps, and source pointers into $\mathcal{E}_t$. For belief items, $w_t(m) \in [0, 1]$ is an operational support score. Episodic traces are fixed at $w_t(e) \equiv 1$ and used only as provenance, not in confidence-weighted Top-K ranking. A `HYPOTHESIS` is a provisional belief: it is reinforced under supporting evidence, decayed when evidence stays absent, and promoted to `FACT` when confidence crosses τ_{promote}. Type transitions are one-way. Contradicted facts are softly decayed by `CORRECT` (Section 2.4) rather than demoted,

yielding low-confidence "deprecated" facts that confidence-weighted retrieval naturally deprioritizes. Appendix Section C.7 confirms this retention is accuracy-neutral.

2.3 Memory leveraging: evidence recall and response generation

In **Memory Leveraging**, PreCoMem recalls a compact evidence set from long-term memory and leverages it to generate the current response.

Evidence recall. Given u_t and short-term context Q_{t-1}, PreCoMem ranks candidates in $\mathcal{B}_t$ with a confidence-modulated score. The score combines three retrieval signals: semantic similarity, lexical overlap (BM25), and topic relevance. Omitting Q_{t-1} in the notation,

$$s(u_t, m) = w_t(m) \sum_{j=1}^{3} \alpha_j s_j(u_t, m), \quad m \in \mathcal{B}_t, \tag{1}$$

where $(s_1, s_2, s_3) = (s_{\text{sem}}, s_{\text{fact}}, s_{\text{topic}})$ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^\top$ is a convex weight vector. The multiplier $w_t(m)$ biases retrieval toward high-confidence items. The Top-K belief items form $\mathcal{C}_t = \{m_i\}_{i=1}^{K}$ with retrieval scores $\mathbf{s}_t$. Linked episodic traces from $\mathcal{E}_t$ are appended as provenance context, but they do not participate in Top-K ranking. The same scores $\mathbf{s}_t$ are retained for the reliability estimate in Section 2.4.

Evidence-conditioned response. The recalled evidence texts $\{c(m) \mid m \in \mathcal{C}_t\}$, together with u_t and the short-term window Q_{t-1}, condition the response model:

$$y_t = \texttt{Respond}(u_t, Q_{t-1}, \mathcal{C}_t). \tag{2}$$

We reuse the same recall outputs $(\mathcal{C}_t, \mathbf{s}_t)$ to perform memory evolution in Section 2.4.

2.4 Memory evolution: regime-gated belief updating

Given the recall outputs $(\mathcal{C}_t, \mathbf{s}_t)$ and the new interaction (u_t, y_t), PreCoMem computes a prediction-error signal adjusted by retrieval reliability, and applies a regime-gated update to the belief store. It also writes the new turn back into the episodic cache for future provenance and recall.

Retrieval reliability. Surprisal computed against recalled evidence can be misleading when recall is uncertain, so we derive a reliability coefficient from the entropy of retrieval scores. Let $\mathbf{s}_t = \{s_i\}_{i=1}^{|\mathcal{C}_t|}$ denote the Top-K retrieval scores returned with $\mathcal{C}_t$, and let $\hat{s}_i = (s_i + \epsilon)/\sum_j (s_j + \epsilon)$ be the smoothed normalization. We define the normalized entropy H_t and the reliability coefficient ρ_t as

$$H_t = \frac{-\sum_i \hat{s}_i \log \hat{s}_i}{\log \max\{|\mathcal{C}_t|, 2\}} \in [0, 1], \qquad \rho_t = 1 - H_t. \tag{3}$$

Because H_t is minimized when one item dominates and maximized under near-uniform scores, ρ_t measures score concentration as a proxy for evidence precision. Diffuse recall (small ρ_t) downweights deviation signals to avoid over-updating; concentrated recall (large ρ_t) preserves them for correction. Thus the gate becomes aggressive only when recall is concentrated and the raw contradiction/deviation signal is large. Since ρ_t enters only multiplicatively, it can defer an update but never trigger one on its own; whether `CORRECT` fires still depends on the contradiction term below, and Appendix Section C.8 verifies empirically that ρ_t tracks evidence precision. The constant $\epsilon > 0$ ensures numerical stability.

Raw surprisal proxy. We define:

$$S_{\text{raw}}(u_t) = \lambda\, d_{\text{sem}}(u_t, \mathcal{C}_t) + (1 - \lambda)\, d_{\text{conf}}(u_t, \mathcal{C}_t), \tag{4}$$

where $\lambda \in (0, 1)$, d_{sem} is the normalized semantic deviation between u_t and score-weighted recalled evidence, and d_{conf} is a verifier-based contradiction score quantifying how strongly u_t conflicts with recalled beliefs (see Appendix Section B.2).

Effective surprisal. We define:

$$S_{\text{eff}}(u_t) = \rho_t\, S_{\text{raw}}(u_t), \tag{5}$$

which corresponds to a precision-weighted prediction error that becomes conservative when recall is unreliable. We use $x_t := S_{\text{eff}}(u_t)$ as the scalar gate score below. Appendix Section B.1 gives an optimization view that interprets x_t as a Monte Carlo surrogate of a long-horizon predictive objective.

Regime-gated evolution. Given two ordered thresholds $\theta_{\text{low}} < \theta_{\text{high}}$, PreCoMem maps the gate score x_t into three regimes:

$$a_t = \begin{cases} \texttt{MAINTAIN}, & x_t \le \theta_{\text{low}} \\ \texttt{PROFILE}, & \theta_{\text{low}} < x_t \le \theta_{\text{high}} \\ \texttt{CORRECT}, & x_t > \theta_{\text{high}}. \end{cases} \tag{6}$$

Two-threshold gate. Low, medium, and high x_t correspond to consistent, ambiguous, and reliably contradictory evidence. `PROFILE` handles the middle regime that a binary maintain/correct rule must either ignore or overwrite. We fix $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$ throughout the main experiments to keep behavior controllable and the comparison reproducible across benchmarks; the cost-sensitive break-even derivation, operator ablations, threshold sensitivity, and dynamic-threshold variant are in Appendix Sections B.3 and C.4 to C.5.

`MAINTAIN` **(reinforcement).** When surprisal is low, PreCoMem strengthens supported beliefs in $\mathcal{C}_t$ via:

$$w_{t+1}(m) = w_t(m) + \eta_t(m)\,(1 - w_t(m)), \quad m \in \mathcal{C}_t \cap \mathcal{B}_t, \tag{7}$$

where $\eta_t(m) \in (0, 1]$ is the per-item EVIDENCE_STRENGTH returned by the `Maintain-Agent`; weakly supported items barely move, while strongly supported ones are pulled toward 1. The optimization view in Appendix Section B.3 treats $\eta_t(m)$ as a scalar step size.

`PROFILE` **(hypothesis writing).** When surprisal is medium, the observation is novel but not decisively contradictory. PreCoMem generates a low-confidence hypothesis h and writes it as a `HYPOTHESIS` belief:

$$\begin{aligned} h &= \texttt{GenerateHypothesis}(u_t, \mathcal{C}_t), \qquad \tau(h) = \texttt{HYPOTHESIS}, \\ b_t^{(c)} &= w_{\min}^{(c)} + \sigma(x_t)\big(w_{\max}^{(c)} - w_{\min}^{(c)}\big), \qquad w_0(h) = \tfrac{1}{2}\, b_t^{(c)} + \tfrac{1}{2}\, \text{clip}\big(q_t, [w_{\min}^{(c)}, w_{\max}^{(c)}]\big), \end{aligned} \tag{8}$$

where `GenerateHypothesis` is the `Profile-Agent` under a fixed prompt, $\sigma(x) = 1/(1 + e^{-x})$, $c \in \{\text{EXPLICIT}, \text{LATENT}\}$ is the class label assigned by the agent, and q_t is the agent's INITIAL_CONFIDENCE, clipped to the class band $[w_{\min}^{(c)}, w_{\max}^{(c)}]$. The gate score thus supplies a reproducible prior $b_t^{(c)}$ inside the band, and the agent scalar can shift w_0 by at most half the band width. *Explicit-event* hypotheses paraphrase stated facts or actions and use the band $[0.60, 0.70]$. *Latent-inference* hypotheses capture deduced traits, moods, or hidden goals and use $[0.50, 0.59]$. These initial values are low relative to stable `FACT`s, which sit at $w \ge 0.80$ and meet the promotion threshold $\tau_{\text{promote}} = 0.80$. A fresh hypothesis therefore ranks below committed facts in confidence-weighted retrieval (Equation (1)) and still needs repeated support to graduate. In later turns, hypotheses are reinforced under `MAINTAIN` when supported and decayed when evidence remains absent:

$$w_t(h) \leftarrow w_t(h) \exp\big(-\gamma\, a_t(h)\big), \qquad a_t(h) = t - \ell(h), \tag{9}$$

where $\ell(h)$ is the turn at which h was created or last received supporting evidence and $\gamma > 0$ is a decay rate. The rule is applied once per turn to unsupported hypotheses: any supporting update resets $\ell(h)$ and skips the decay in that turn, and hypotheses flagged as DECAY_PROTECTION by the agent (explicit events) are exempt, so the compounding decay over consecutive unsupported turns (cumulative factor $\exp(-\gamma\, n(n+1)/2)$ after n turns; Appendix Section B.3) prunes only guesses that are never revalidated. Hypotheses are promoted to facts once confidence crosses τ_{promote}, with $0 \le w_{\min}^{(c)} < w_{\max}^{(c)} < \tau_{\text{promote}} \le 1$ enforced for every class so that promotion always requires repeated support.

`CORRECT` **(soft correction).** When surprisal is high and recall is reliable, the new observation likely contradicts existing high-confidence beliefs. PreCoMem records the new observation as a fresh belief node (a `FACT` with initial weight 0.85) and separately decays the retrieved beliefs it contradicts. The decay side is:

$$w_{t+1}(m) = w_t(m) \exp\big(-\beta\, x_t\, \zeta_t(m)\big), \quad m \in \mathcal{C}_t \cap \mathcal{B}_t, \tag{10}$$

where $\beta > 0$ controls the correction strength and $\zeta_t(m) \in [0, 1]$ is the per-item *contradiction severity* returned by the `Correct-Agent`; non-contradictory items receive $\zeta_t(m) \approx 0$ and are essentially unchanged, enabling selective correction without extra thresholds. We deliberately separate this context-aware decay magnitude from the fixed NLI contradiction score that feeds d_{conf} (Equation (22) in Appendix Section B.2): the NLI score drives the gate via S_{raw}, whereas ζ_t localizes how strongly each retrieved belief should decay. A derivation is provided in Appendix Section B.3.

Table 1: Main evaluation on LoCoMo and LongMemEval-S. Accuracies are reported as mean ± std over three seeds; within each backbone block, **bold** marks the best score per column and underline marks the second-best. *Avg. Tokens* reports the average response-model input token usage per query. LongMemEval-S columns are question types: SSU/SSA/SSP = single-session user/assistant/preference, MS = multi-session, KU = knowledge update, and TR = temporal reasoning.

LoCoMo (1,540 questions; four splits)

Method	Avg. Tokens	Single-Hop	Multi-Hop	Temporal	Open-Domain	Overall
BACKBONE: *gpt-4o-mini*						
MemoryOS	5.5k	66.68 ± 0.85	54.83 ± 0.87	52.90 ± 0.59	32.16 ± 0.11	59.58 ± 0.06
Mem0	**1.6k**	62.17 ± 0.77	52.91 ± 0.70	59.44 ± 0.37	33.89 ± 0.96	58.36 ± 0.31
A-Mem	2.7k	45.15 ± 0.53	23.99 ± 0.46	53.67 ± 0.23	30.71 ± 0.84	42.18 ± 0.65
GAM	3.5k	68.24 ± 0.76	55.47 ± 0.43	58.19 ± 0.54	33.50 ± 0.09	61.62 ± 0.22
MemBox	3.3k	70.89 ± 0.68	53.31 ± 0.15	62.70 ± 0.57	35.18 ± 0.77	63.30 ± 0.48
MemOS	2.5k	77.05 ± 0.93	68.34 ± 0.79	61.27 ± 0.01	39.26 ± 0.07	70.20 ± 0.66
EverMemOS	2.6k	**83.69 ± 0.30**	59.97 ± 0.72	66.00 ± 0.39	36.95 ± 0.56	70.13 ± 0.55
O-Mem	2.9k	78.59 ± 0.04	**71.08 ± 0.34**	60.20 ± 0.91	37.41 ± 0.87	70.88 ± 0.22
TiMem	2.8k	76.64 ± 0.60	60.43 ± 0.80	**76.74 ± 0.48**	38.86 ± 0.94	71.88 ± 0.61
PreCoMem	3.3k	81.41 ± 0.51	68.49 ± 0.98	66.21 ± 0.12	**39.63 ± 0.25**	**73.57 ± 0.28**
BACKBONE: *gpt-4.1*						
MemoryOS	5.2k	72.21 ± 0.38	63.66 ± 0.18	63.10 ± 0.83	43.33 ± 0.88	66.02 ± 0.25
Mem0	**1.8k**	72.59 ± 0.73	62.37 ± 0.49	67.85 ± 0.20	42.98 ± 0.89	67.17 ± 0.51
A-Mem	2.9k	50.63 ± 0.09	35.96 ± 0.39	59.50 ± 0.24	34.30 ± 0.54	48.90 ± 0.22
GAM	3.4k	70.35 ± 0.38	67.45 ± 0.02	68.47 ± 0.64	39.03 ± 0.05	68.81 ± 0.11
MemBox	3.4k	76.76 ± 0.10	67.47 ± 0.33	68.40 ± 0.52	41.78 ± 0.82	71.29 ± 0.92
MemOS	2.4k	84.57 ± 0.70	80.14 ± 0.56	70.79 ± 0.43	46.93 ± 0.87	78.62 ± 0.76
O-Mem	2.7k	86.81 ± 0.73	**84.32 ± 0.42**	68.04 ± 0.03	42.24 ± 0.75	80.08 ± 0.14
EverMemOS	2.5k	**92.13 ± 0.94**	77.75 ± 0.52	74.72 ± 0.91	43.32 ± 0.31	82.27 ± 0.86
TiMem	2.7k	87.00 ± 0.58	83.95 ± 0.15	**85.92 ± 0.71**	45.82 ± 0.60	84.04 ± 0.84
PreCoMem	3.1k	90.41 ± 0.35	83.48 ± 0.29	84.74 ± 0.80	**48.23 ± 0.65**	**85.06 ± 0.07**

LongMemEval-S (500 questions; six splits)

Method	Avg. Tokens	SSU	SSA	SSP	MS	KU	TR	Overall
BACKBONE: *gpt-4.1* *(gpt-4o-mini results in Appendix Section C.3)*								
MemoryOS	6.8k	62.77 ± 0.01	63.53 ± 0.19	48.71 ± 0.00	59.27 ± 0.57	58.94 ± 0.67	54.95 ± 0.14	58.49 ± 0.04
MemOS	**2.6k**	80.88 ± 0.48	73.29 ± 0.20	58.43 ± 0.15	70.10 ± 0.65	75.05 ± 0.55	68.32 ± 0.21	72.74 ± 0.40
TiMem	3.5k	85.06 ± 0.03	79.18 ± 0.54	71.22 ± 0.62	76.58 ± 0.84	81.96 ± 0.50	**89.86 ± 0.41**	79.63 ± 0.79
GAM	4.2k	80.90 ± 0.70	83.73 ± 0.26	70.82 ± 0.02	80.81 ± 0.46	78.30 ± 0.51	79.28 ± 0.23	80.91 ± 0.34
MemBox	4.0k	82.07 ± 0.42	85.13 ± 0.64	73.68 ± 0.61	80.56 ± 0.38	79.59 ± 0.78	81.52 ± 0.89	81.24 ± 0.60
O-Mem	3.4k	91.25 ± 0.12	86.76 ± 0.74	74.69 ± 0.07	83.89 ± 0.64	87.05 ± 0.29	82.77 ± 0.88	85.36 ± 0.08
EverMemOS	3.1k	**96.31 ± 0.92**	83.33 ± 0.97	72.45 ± 0.47	80.66 ± 0.37	86.25 ± 0.11	79.17 ± 0.75	86.16 ± 0.69
PreCoMem	3.8k	94.09 ± 0.98	**90.72 ± 0.12**	**83.93 ± 0.85**	**87.44 ± 0.99**	**91.80 ± 0.87**	84.35 ± 0.83	**89.39 ± 0.76**

Trace and topic write-back. After generating y_t, PreCoMem appends the turn trace (u_t, y_t) to the episodic cache $\mathcal{E}_{t+1}$ with a fixed-capacity eviction policy; optionally, new topics are added to $\mathcal{T}_{t+1}$ when detected, yielding $\mathcal{M}_{t+1} = (\mathcal{B}_{t+1}, \mathcal{E}_{t+1}, \mathcal{T}_{t+1})$. When `PROFILE/MAINTAIN/CORRECT` creates or updates belief items, we record source pointers to supporting traces so beliefs remain editable while evidence remains traceable.

3 Experiments

3.1 Experimental setup

Benchmarks. We evaluate on LoCoMo [34] (1,500+ questions in long contexts), LongMemEval-S [16] (500 questions covering six question types over five long-term memory abilities), and PersonaMem-v2 [35] (1,000 personas and 20,000+ implicit preferences, evaluated under both 32k

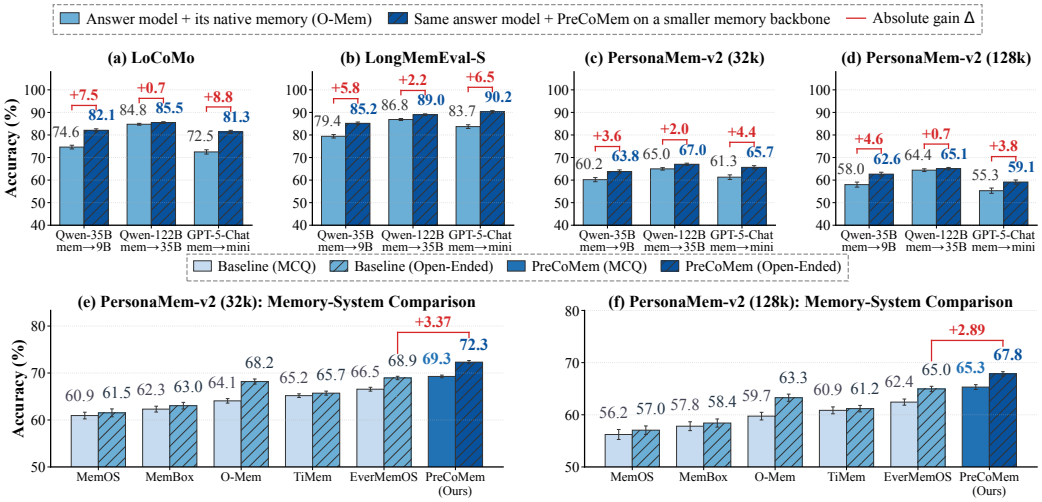

Figure 3: **Backbone generalization and PersonaMem-v2 system comparison. (a)–(d)** Scale-vs-mechanism comparison on LoCoMo, LongMemEval-S, PersonaMem-v2 (32k), and PersonaMem-v2 (128k). Within each group the answer model is fixed (Qwen3.5-35B, Qwen3.5-122B, or GPT-5-Chat): the left bar runs it with its native O-Mem memory, while the right bar keeps the same answer model and delegates memory construction to PreCoMem on a one-tier-smaller backbone (Qwen3.5-9B, Qwen3.5-35B, GPT-5-mini); the protocol is detailed in Appendix Section C.2. **(e)–(f)** Head-to-head with the five strongest baselines on PersonaMem-v2 under *gpt-4.1*, in both MCQ and Open-Ended formats. Error bars are ±std over three seeds.

and 128k contexts; Figure 3(c)–(f)). To stress-test belief drift under sparse signals we additionally introduce DiMoBench, which adds controlled non-stationarity and decoy retrieval (Section 3.3). Two third-party benchmarks, AMemGym [36] and LoCoMo-Plus [37], are evaluated under their official protocols in Appendix Section C.1.

Baselines. We compare PreCoMem against O-Mem [21], MemoryOS [38], MemOS [18], Mem0 [26], A-Mem [24], GAM [25], MemBox [23], TiMem [22], and EverMemOS [19], each under its officially recommended configuration; we standardize the backbone LLM and embedding model across methods whenever applicable.

Evaluation protocol. We adopt the LLM-as-a-judge protocol from MemOS [18]: each answer is scored by a primary judge (*gpt-4.1-mini*) and two auxiliary *gemini-2.5-flash* judges (run with distinct system prompts to reduce judge-correlation) in a blinded setup, with accuracy reported as the average of the three binary judgments [39–43]. Appendix Section C.3 reports human–LLM agreement on LoCoMo, LongMemEval-S, and DiMoBench plus a cross-method spot-check on baseline outputs.

Implementation details. PreCoMem uses *gpt-4.1* (or *gpt-4o-mini*) for all LLM-based reasoning and memory operations and `Qwen3-Embedding-0.6B` for dense embeddings across methods. Unless otherwise specified, k=10, $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$, λ=0.5, β=1.0, $\boldsymbol{\alpha}$=$(0.4, 0.4, 0.2)$, and γ=0.05; sensitivity studies and the adaptive variant are in Appendix Section C.5.

3.2 Main results

Main results are reported in Table 1: the upper block on LoCoMo and the lower block on LongMemEval-S. We make three observations.

(1) **Overall performance.** PreCoMem achieves the best overall accuracy on both tabulated benchmarks. On LoCoMo it leads the strongest baseline (TiMem) by +1.69 with *gpt-4o-mini* (73.57 vs. 71.88) and +1.02 with *gpt-4.1* (85.06 vs. 84.04); both gaps exceed the cross-seed standard deviation of either system reported in Table 1. On LongMemEval-S it reaches 89.39 under *gpt-4.1* (+3.23 over EverMemOS), and the *gpt-4o-mini* appendix block follows the same pattern (84.28, +1.96; Appendix Section C.3). On PersonaMem-v2 (Figure 3(e)–(f)) under *gpt-4.1*, PreCoMem reaches 72.31 Open-Ended at 32k and 67.84 at 128k, +3.37 and +2.89 over EverMemOS. The advantage

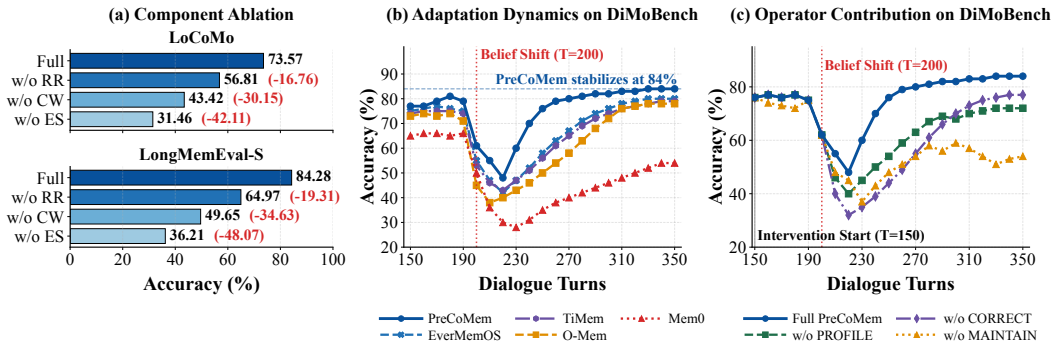

Figure 4: **Component ablation and DiMoBench dynamics. (a)** Component ablation on LoCoMo and LongMemEval-S. **(b)** Moving-average accuracy under implicit drift on DiMoBench. **(c)** Operator-level ablation on DiMoBench.

also holds on two third-party benchmarks we did not design: 0.458 Overall / 0.329 normalized Memory on AMemGym against 0.430 / 0.296 for the strongest system reported there, and 22.4 on LoCoMo-Plus against 21.05 for full-context GPT-4o (Appendix Section C.1). Figure 3(a)–(d) further shows that, with the answer model held fixed, delegating memory construction to PreCoMem on a one-tier-smaller backbone still beats the larger backbone's native memory in all twelve comparisons.

(2) **Where gains come from.** On LoCoMo, the gain concentrates on the Open-Domain split (PreCoMem 48.23 under *gpt-4.1*, +1.30 over MemOS; consistently best under *gpt-4o-mini*), whose questions typically ask for a latent state that is not stated in a single turn and must be inferred from cues spread across sessions (Appendix Section C.3), while EverMemOS, O-Mem, and TiMem retain narrow leads on Single-Hop / Multi-Hop / Temporal under *gpt-4.1* from their dedicated read-path designs. On LongMemEval-S (*gpt-4.1*), the largest gains occur on the question types whose answers depend on updating or aggregating user-specific memory: **SSP** (single-session preference, +9.24 over O-Mem), **KU** (knowledge update, +4.75 over O-Mem; every KU question carries an old value superseded by a newer one), and **MS** (multi-session aggregation, +3.55). The **SSA** gain (assistant-side single-session information, +3.96) reflects selective memory use: the single target session lies among 48.9 sessions on average, and confidence-weighted retrieval acts as an implicit reranker. PersonaMem-v2 reproduces this pattern: PreCoMem's lead is largest on anti-stereotypical, dynamic, and ownership-disambiguation categories that most stress write-path quality, and shrinks to under two points on stereotypical or static cues where population priors suffice. PreCoMem's advantage is therefore strongest when answers require selective memory use, belief updating, or cross-session aggregation; a structural analysis attributes the gains to two design properties absent from contemporary baselines—continuous per-belief confidence that buffers ambiguous cues, and a confidence-weighted retrieval score that acts as an implicit reranker (Appendix Section C.10).

(3) **Accuracy–efficiency trade-off.** Figure 5(a) shows the LoCoMo performance–cost frontier as we vary k: PreCoMem peaks at $k = 10$ with moderate token usage, while larger k yields diminishing returns. Supporting diagnostics on hypothesis lifecycle, Top-k sensitivity, and memory footprint are reported in Appendix Sections C.3, C.5 and C.7.

3.3 Ablation studies and belief-drift adaptation

Component ablation. We ablate three core components while keeping the backbone, prompts, and hyperparameters fixed. ES removes the effective-surprisal gate by setting the gate score to 0; RR removes retrieval-reliability adjustment via $\rho_t \equiv 1$; CW removes confidence weighting globally, in retrieval and in every lifecycle test, by setting all belief weights to 1. Figure 4(a) reports LLM-judge accuracy on LoCoMo and LongMemEval-S. Removing ES causes the largest drop on LoCoMo (73.57 $\rightarrow$ 31.46; similar on LongMemEval-S): with the gate score zeroed every turn is routed to `MAINTAIN`, so this reinforce-only variant marks the lower bound of *surprisal-gated evolution*, the primary mechanism under non-stationarity, while route-matched controls that isolate the gate's marginal value are reported in Appendix Section C.4. Removing CW ($\rightarrow$ 43.42) and RR ($\rightarrow$ 56.81) also substantially reduces accuracy, with the same trend on LongMemEval-S. Letting the same LLM

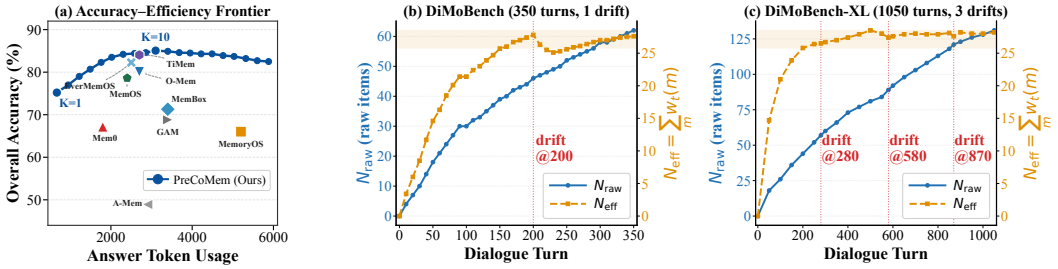

Figure 5: **Hyperparameter and memory-scalability analysis. (a)** LoCoMo accuracy–efficiency frontier as the retrieval budget k varies; overall accuracy peaks at $k = 10$, beyond which additional context yields diminishing returns and dilutes retrieval reliability ρ_t. **(b, c)** Memory scalability under long-horizon belief drift: raw memory grows linearly, while effective memory quickly plateaus on both DiMoBench and DiMoBench-XL (panel (c) shows one representative persona).

emit the entire write decision end-to-end instead of bounded scalars lowers accuracy by 2.75 / 3.15 points on LoCoMo / LongMemEval-S and raises the cross-seed standard deviation $20\times$ on LoCoMo (under *gpt-4.1*; Appendix Section C.4). Appendix Section C.4 shows the NLI contradiction term drives most gains.

DiMoBench setup. **DiMoBench** (Dialogue Implicit Memory Drift Benchmark) is a controlled diagnostic setting for gradual attitude and preference drift. A user's latent stance reverses around a change point, while keyword-matched chatter and entity confounders pollute the stream so naive retrieval and overwrite-style updates become brittle. The released suite has *five independent* 350-turn dialogues, each driven by a distinct persona, with 200 QA pairs per dialogue (1,000 total; results averaged across personas); DiMoBench-XL extends this to 1050 turns and three sequential drifts, with 420 QA pairs each, 2,100 in total (Appendix Section C.6).

Update dynamics. Figure 4(b) plots LLM-judge accuracy across dialogue turns. Before the change point all methods are stable; afterwards PreCoMem experiences a transient drop but recovers fastest and stabilizes at 84%, while EverMemOS, TiMem, and O-Mem plateau at 78–80% and Mem0 fails to recover (54%). End-to-end traces are provided in Appendix Section C.11.

Operator ablation on DiMoBench. Ablating each operator individually (Figure 4(c)) yields qualitatively distinct failure modes: removing `PROFILE` caps the asymptote at 72% (vs. 84%); removing `CORRECT` delays recovery (turn 320 vs. 250) and lowers the plateau to 77%; removing `MAINTAIN` fails to stabilize within the horizon (54% at the end), confirming that the three operators play distinct stabilizing roles. Appendix Section C.4 provides the full analysis, and the same three qualitative signatures reproduce on the third-party AMemGym benchmark (Appendix Section C.1).

3.4 Hyperparameter analysis

We sweep the retrieval budget $k \in \{1, \ldots, 20\}$ on LoCoMo (*gpt-4.1*). Accuracy rises sharply through $k \approx 6$, peaks at $k = 10$ (Overall 85.06%), then drops mildly. Larger k flattens the Top-k score distribution, lowering ρ_t and dampening the effective surprisal x_t that drives `CORRECT`. It also increases the chance that one retrieved item spuriously appears contradictory under the max-based contradiction component d_{conf}. We therefore fix $k = 10$; the full sweep and retrieval-latency profile are reported in Appendix Sections C.5 and C.9. Because the write path acts only on retrieved beliefs, Appendix Section C.8 further measures how often the belief due for an update actually enters the Top-k and what a miss costs.

3.5 Memory scalability

A natural concern for any long-term memory system is whether storage and retrieval cost scale gracefully with the conversation length. We examine this on three settings spanning natural and stress-test horizons: the LoCoMo benchmark (ten multi-session dialogues, mean $\bar{N} = 588$ turns), DiMoBench (350 turns, 1 drift), and DiMoBench-XL (1050 turns, 3 drifts; five personas, of which a representative one is plotted, the others staying within ± 1.4 of it; Appendix Section C.6). Two quantities are tracked every 10 turns on DiMoBench and every 50 turns on DiMoBench-XL:

Table 2: **Memory footprint vs. accuracy on the ten LoCoMo dialogues** (mean turns $\bar{N} = 588$). All numbers are measured under the same backbone (*gpt-4o-mini*) and embedding model. "Per-item conf." indicates whether the system maintains a continuous confidence weight per stored item; under discrete commit decisions, the effective count N_{eff} and the average Top-10 weight are not defined. PreCoMem (A1) is the optional pruning configuration that drops items with $w_t < 0.05$.

Method	Per-item conf.	$N_{\text{raw}}^{\text{end}}$	$N_{\text{eff}}^{\text{end}}$	Top-10 avg. w	LoCoMo Overall
O-Mem [21]	✗ (ADD/IGNORE/UPDATE)	232	–	–	70.88
EverMemOS [19]	✗ ($[t_s, t_e]$ expiry)	264	–	–	70.13
PreCoMem (A0, default)	✓ ($w_t \in [0, 1]$)	287	95.7	0.81	**73.57**
PreCoMem (A1, $w < 0.05$ prune)	✓ ($w_t \in [0, 1]$)	**168**	95.6	0.83	73.51

- *Raw count* $N_{\text{raw}}(t) = |\mathcal{B}_t|$: the number of belief items physically stored, including *deprecated facts*. Deprecated facts are previously committed `FACT`s whose confidence has been softly decayed below 0.1 by repeated `CORRECT` updates (Equation (10)); we keep them in $\mathcal{B}_t$ as provenance hooks rather than deleting them.
- *Effective count* $N_{\text{eff}}(t) = \sum_{m \in \mathcal{B}_t} w_t(m)$: the confidence-weighted total mass. Intuitively, this is the equivalent number of fully-supported items the agent "acts on" through the confidence-weighted retrieval score in Equation (1). A deprecated fact at $w = 0.05$ contributes only 0.05 to N_{eff}, which matches its negligible influence on Top-K ranking.

Figure 5(b, c) shows N_{raw} grows roughly linearly with the dialogue, but N_{eff} saturates by turn 200 (at 25.8–28.5 on the representative persona, with the other personas within ± 1.4 of it) and, apart from a transient dip at each drift, stays in that band. Moreover, deprecated facts occupy only 1.0% of Top-10 slots on average even after 1050 turns and three drifts (Appendix Table 19). Confidence weighting therefore turns linearly growing storage into a roughly constant retrieval surface.

Table 2 extends this picture to physical storage on LoCoMo, where we measure the final $\mathcal{B}_t$ size of each system under matched backbone and embedding protocols. By default (A0) PreCoMem keeps deprecated facts as provenance hooks for traits that may resurface later, since confidence-weighted retrieval already filters them out of the Top-K; the optional A1 configuration ($w < 0.05$ prune) trades this provenance for tighter storage and matches A0 within seed variance (Appendix Section C.7). Both O-Mem and EverMemOS commit each interaction through a discrete decision (ADD/IGNORE/UPDATE in O-Mem; episodic-trace formation with a binary $[t_{\text{start}}, t_{\text{end}}]$ validity in EverMemOS), so neither maintains a continuous per-item confidence and N_{eff} is undefined. Even on raw count alone, A1 stores fewer items than both baselines while still leading on accuracy by $+2.6$ and $+3.4$ overall points, so storage remains bounded in both the soft sense (N_{eff} plateau) and the hard sense. This suggests the memory gain is not purchased by unchecked accumulation.

4 Conclusion

PreCoMem treats long-horizon conversational memory not as a transcript to be stored, but as confidence-weighted beliefs about the user's state that must be continually checked. A reliability-adjusted surprisal signal routes each turn into `MAINTAIN` (reinforce supported beliefs), `PROFILE` (buffer ambiguous cues as low-confidence hypotheses), or `CORRECT` (softly decay contradicted facts). The `PROFILE` waiting room is the central design choice: it lets evidence accumulate before commitment, so the agent stays consistent under noise without freezing under genuine drift. Across LoCoMo, LongMemEval-S, PersonaMem-v2, and DiMoBench, this discipline yields the best overall accuracy while keeping memory bounded over a thousand turns; qualitative case studies (Appendix Section C.11) show calibrated recommendations beyond retrieval-only baselines. We see continuous confidence and explicit hypothesis lifecycles as a useful baseline for long-term memory systems, where **what to commit matters as much as what to recall**. Our evaluation covers English dialogue benchmarks; other languages and modalities are left for future work.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and the area chair for their constructive feedback.

Funding. This research is supported by the National Natural Science Foundation of China under Grant 62406163, the Tianjin Science and Technology Plan Project under Grant 25ZXSFSN00150, and the Fundamental Research Funds for the Central Universities under Grant 63261039.

Competing interests. The authors declare no competing interests.

References

[1] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[2] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024.

[3] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China information sciences*, 68(2):121101, 2025.

[4] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19724–19731, 2024.

[5] Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025.

[6] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

[7] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in neural information processing systems*, 36:74530–74543, 2023.

[8] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632 (8026):768–774, 2024.

[9] Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022.

[10] Raquib Bin Yousuf, Aadyant Khatri, Shengzhe Xu, Mandar Sharma, and Naren Ramakrishnan. Can an llm induce a graph? investigating memory drift and context length. In *2025 IEEE International Conference on Knowledge Graph (ICKG)*, pages 379–388. IEEE, 2025.

[11] Vardhan Dongre, Ryan A Rossi, Viet Dac Lai, David Seunghyun Yoon, Dilek Hakkani-Tür, and Trung Bui. Drift no more? context equilibria in multi-turn llm interactions. *arXiv preprint arXiv:2510.07777*, 2025.

[12] Dietmar Jannach, Lukas Lerche, and Markus Zanker. Recommending based on implicit feedback. In *Social information access: systems and technologies*, pages 510–569. Springer, 2018.

[13] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*, volume 2024, pages 29862–29883, 2024.

[14] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[15] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 719–729, 2024.

[16] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

[17] Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in llm agents. *arXiv preprint arXiv:2604.27003*, 2026.

[18] 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.

[19] Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, et al. Evermemos: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 45836–45853, 2026.

[20] Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

[21] 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.

[22] Kai Li, Xuanqing Yu, Ziyi Ni, Yi Zeng, Yao Xu, Zheqing Zhang, Xin Li, Jitao Sang, Xiaogang Duan, Xuelei Wang, et al. Timem: Temporal-hierarchical memory consolidation for long-horizon conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 21700–21720, 2026.

[23] Dehao Tao, Guoliang Ma, Yongfeng Huang, and Minghu Jiang. Membox: Weaving topic continuity into long-range memory for llm agents. *arXiv preprint arXiv:2601.03785*, 2026.

[24] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38: 17577–17604, 2026.

[25] BY Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*, 2025.

[26] Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

[27] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[28] Andy Clark et al. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204, 2013.

[29] Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.

[30] Clara Meister, Ryan Cotterell, and Tim Vieira. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, 2020.

[31] Charan Ranganath and Gregor Rainer. Neural mechanisms for detecting and remembering novel events. *Nature Reviews Neuroscience*, 4(3):193–202, 2003.

[32] Alyssa H Sinclair and Morgan D Barense. Prediction error and memory reactivation: How incomplete reminders drive reconsolidation. *Trends in neurosciences*, 42(10):727–739, 2019.

[33] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

[34] 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.

[35] Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, et al. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025.

[36] Jiayang CHENG, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu SONG, and Xunliang Cai. Amemgym: Interactive memory benchmarking for assistants in long-horizon conversations. In *Paper presented at The 14th International Conference on Learning Representations (ICLR 2026), Rio de Janeiro, Brazil*, 2026.

[37] Yifei Li, Weidong Guo, Lingling Zhang, Rongman Xu, Muye Huang, Hui Liu, Lijiao Xu, Yu Xu, and Jun Liu. Locomo-plus: Beyond-factual cognitive memory evaluation framework for llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25085–25100, 2026.

[38] Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25972–25981, 2025.

[39] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024.

[40] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 7(6), 2026.

[41] Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge model is not a general substitute for gpt-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, 2025.

[42] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, 2025.

[43] Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. Learning llm-as-a-judge for preference alignment. In *International Conference on Learning Representations*, volume 2025, pages 3537–3564, 2025.

[44] Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025.

[45] Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, volume 2026, pages 39458–39486, 2026.

[46] Haoran Sun, Zekun Zhang, and Shaoning Zeng. Preference-aware memory update for long-term llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 783–793, 2026.

[47] Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026.

[48] Zisu Huang, Muzhao Tian, Xiaohua Wang, Jingwen Xu, Zhengkang Guo, Qi Qian, Kaitao Song, Jiakang Yuan, Changze Lv, and Xiaoqing Zheng. Controllable memory usage: Balancing anchoring and innovation in long-term human–agent interaction. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14699–14719, 2026.

[49] Junfeng Lu and Yueyan Li. Dynamic affective memory management for personalized llm agents. *arXiv preprint arXiv:2510.27418*, 2025.

[50] Zeyu Liu, Jian Zhong, Rongduo Han, Ziyang Wu, Shunye Tang, Chenghao He, Yaxuan Yang, Yihang Qiu, Ailing Wang, Xiao Liang, et al. Evalmem: An operation-level diagnostic framework for long-term memory systems. *arXiv preprint arXiv:2609.22231*, 2026.

[51] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, et al. Secom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, volume 2025, pages 91851–91885, 2025.

Appendix contents

A Related work **15**

B Additional method details **16**

B.1 From expected predictive loss to a per-turn on-policy loss 16

B.2 Retrieval scoring and surprisal component definitions 17

B.3 Operator and gate derivations 18

B.4 Per-turn pipeline (hybrid execution) 20

C Additional experimental details **20**

C.1 Third-party benchmarks: AMemGym and LoCoMo-Plus 20

C.2 Backbone-generalization protocol and token usage 22

C.3 Extended results and judge validation 22

C.4 Component, operator, and control ablations 24

C.5 Hyperparameter sensitivity and adaptive variants 28

C.6 DiMoBench: construction and extended-horizon stress test 31

C.7 Memory dynamics: footprint, contamination, and retention 34

C.8 Retrieval dependence and reliability calibration 36

C.9 Retrieval and end-to-end latency analysis 38

C.10 Structural comparison with existing memory systems 39

C.11 System inspectability, traces, and qualitative analyses 41

D Prompts **44**

A Related work

Long-term memory for conversational LLMs. Early work attacks the fixed context window with externalized hierarchical or virtual memory [4, 6]. More recent systems either organize the store more aggressively or learn richer write rules. *Structured organizations* include interlinked note stores with associative links [24], time-aware knowledge graphs [20], OS-style hierarchical scheduling [18, 19], and dedicated memory-box layouts that segment a user's history into typed shards [23]. *Learning-based update rules* optimize memory operations against downstream rewards [25, 44, 45], while *interpretable triggers* target a single failure mode such as preference change [46], redundancy [22, 47], controllable reliance [48], or affective state [49]. A systematic gap shared by these systems is the *stability–plasticity* regime: append-only memories [24] treat every observation as equally valid and dilute retrieval; overwrite memories compress aggressively but lose information that later turns out to be relevant [26]; OS-style memories [18, 19] schedule access well but still fall back on importance heuristics for writes; and reinforcement-trained memories [25, 44, 45] face sparse credit assignment, since most reward signal is several turns downstream of the responsible read/write decision. PreCoMem is complementary to all of these: it treats the write path as a single *regime-gated* decision driven by a calibrated prediction-error signal, instead of relying on recency heuristics or task-specific reward shaping.

Predictive coding and surprisal as write signals. Predictive coding theory holds that perceptual systems update internal models when observations deviate from prediction [27, 28], and information theory gives this deviation a precise scalar form: surprisal [29, 30]. The same logic has long shaped accounts of episodic memory: novel events are encoded preferentially, and reactivation drives reconsolidation of related traces [31, 32]. Translating this into long-term conversational memory

requires two extra ingredients absent from prior memory systems. First, surprisal must be *calibrated against retrieval reliability*, otherwise diffuse recall produces noisy update signals [13, 33]. Second, surprisal must be *regime-gated*, since consistent, ambiguous, and reliably contradictory evidence call for qualitatively different memory edits. PreCoMem operationalizes both ingredients: S_{eff} multiplies raw surprisal by an entropy-based reliability coefficient, and a two-threshold gate maps the result to `MAINTAIN/PROFILE/CORRECT`.

Benchmarking long-term memory. LoCoMo [34] provides 1,500+ questions over multi-session dialogues with explicit single-hop, multi-hop, temporal, and open-domain splits. LongMemEval-S [16] tests five core abilities (information extraction, multi-session reasoning, knowledge updates, temporal reasoning, abstention) with 500 questions over chat histories of roughly 115k tokens each. PersonaMem-v2 [35] stresses *implicit* personalization across $1{,}000$ personas and $20{,}000+$ preferences, where most signal is buried in everyday task-oriented turns. Two more recent benchmarks move closer to the write path: AMemGym [36] evaluates memory on-policy against a simulated user whose latent task state changes across periods, and LoCoMo-Plus [37] probes implicit cognitive constraints (causal, state, goal, and value) over long dialogues; we report both in Section C.1. EvalMem [50] complements these end-task benchmarks with an operation-level diagnostic framework that scores individual memory operations such as adding, updating, and deleting items. These benchmarks probe the read path or the end result of memory updates; none of them, however, isolates the *write-path* decision under continuous, gradual drift with adversarial retrieval interference. Our companion benchmark **DiMoBench** fills this gap with five independent 350-turn dialogues, one per persona, whose latent attitudes reverse gradually across change points; an extended variant, **DiMoBench-XL**, contains five independent 1050-turn dialogues, each with three sequential drifts.

B Additional method details

B.1 From expected predictive loss to a per-turn on-policy loss

Conceptual predictive loss. Let $h_t = (u_{\le t}, y_{<t})$ denote the dialogue history up to turn t. The conceptual predictive loss

$$\mathcal{L}_{\text{pred}}(t) = \mathbb{E}_{u \sim p(u \mid h_t)}[S_{\text{eff}}(u)] \tag{11}$$

captures the expected effective surprisal under the (unknown) interaction distribution $p(u \mid h_t)$, and the long-horizon goal is to update the memory state to reduce future expected losses,

$$\min_{\mathcal{M}_{t+1}} \ \mathbb{E}[\mathcal{L}_{\text{pred}}(t+k) \mid \mathcal{M}_{t+1}], \qquad k \ge 1. \tag{12}$$

Monte Carlo surrogate. Because $p(u \mid h_t)$ is generally unknown, under a fixed response policy we treat the observed turn u_t as a Monte Carlo sample and use the estimator

$$\widehat{\mathcal{L}}_{\text{pred}}(t) := S_{\text{eff}}(u_t). \tag{13}$$

Defining the per-turn on-policy loss $x_t := \widehat{\mathcal{L}}_{\text{pred}}(t)$ yields an unbiased estimate

$$\mathbb{E}[x_t \mid h_t] = \mathcal{L}_{\text{pred}}(t). \tag{14}$$

In the main text we write $S_{\text{eff}}(u_t)$ without explicitly conditioning on the memory state; implicitly, S_{eff} depends on the retrieved evidence produced by $\texttt{Retrieve}(\mathcal{M}_t, u_t)$.

Greedy regularized surrogate for memory updates. The long-horizon objective in Equation (12) can be approximated on-policy by folding "future benefit" into a myopic, regularized surrogate:

$$\mathcal{M}_{t+1} \approx \arg\min_{\mathcal{M}} \ x_t(\mathcal{M}) + D(\mathcal{M}, \mathcal{M}_t) + \Omega(\mathcal{M}), \tag{15}$$

where $x_t(\mathcal{M})$ denotes the effective-surprisal loss one would obtain if evaluated with memory state $\mathcal{M}$, D is an edit/stability cost (e.g., penalizing large changes or changes to high-confidence beliefs), and Ω is a capacity/complexity penalty that discourages unbounded growth of provisional beliefs. Concretely, we instantiate Ω with a lightweight pruning rule for the hypothesis pool: belief items with $\tau(m) = \texttt{HYPOTHESIS}$ are removed once their confidence falls below a small threshold (we use $w_t(m) < 0.1$), which controls memory size while retaining only actively supported hypotheses. In

contrast, belief items with $\tau(m) = \texttt{FACT}$ are not directly deleted; due to the one-way promotion scheme in Section 2.2, outdated facts are instead handled by soft-decay correction that reduces their confidence and thus deprioritizes them under confidence-weighted retrieval. The Memory Evolution phase instantiates a discrete, interpretable approximation of Equation (15): action selection corresponds to cost-sensitive choice (Section B.3), while operator updates admit simple optimization views (Section B.3).

Why low-confidence facts are retained rather than pruned. We deliberately apply the $w_t(m) < 0.1$ pruning rule only to hypotheses, not to facts. Two design considerations motivate this asymmetry. First, because retrieval is confidence-weighted, an outdated fact with $w \approx 0.02$ contributes roughly $50\times$ less to the retrieval score than a currently supported fact with $w \approx 1$; such items are naturally pushed out of the Top-K evidence set by currently supported beliefs and become *functionally inactive* under confidence-weighted retrieval even though they remain in storage. Second, long-horizon dialogue often revisits past user traits after long gaps (e.g., a user stops drinking coffee for months and then mentions it again); retaining the deprecated fact at low confidence preserves the reasoning chain and episodic-cache links needed to recover such context, whereas hard deletion would force the system to rebuild that trait from scratch. The resulting *effective memory*, i.e., the subset of items that can still influence Top-K retrieval, therefore remains controlled even if raw storage accumulates history. Under truly extreme horizons, raw storage can be further bounded by a timestamped time-to-live on long-deprecated facts without modifying the core operators; we leave this as an engineering safeguard rather than a core mechanism.

B.2 Retrieval scoring and surprisal component definitions

Retrieval scoring. Let $\mathbf{e}(\cdot)$ be a text embedding function and let $\mathcal{P}_t$ be the merged candidate pool recalled from $\mathcal{B}_t$ and $\mathcal{E}_t$. We use three reproducible relevance signals:

$$s_{\text{sem}}(u_t, m) = \tfrac{1}{2}\left(1 + \cos(\mathbf{e}(u_t), \mathbf{e}(c(m)))\right) \in [0, 1], \tag{16}$$

$$s_{\text{fact}}(u_t, m) = \frac{\text{BM25}(u_t, c(m))}{\max_{m' \in \mathcal{P}_t} \text{BM25}(u_t, c(m')) + \epsilon} \in [0, 1], \tag{17}$$

where $\epsilon > 0$ is a stability constant. For topical relevance, we maintain topic prototypes $\{\mathbf{t}_z\}_{z=1}^{Z_t}$ (e.g., $\mathbf{t}_z = \mathbf{e}(\ell_z)$ for topic description ℓ_z) and define

$$\begin{aligned} s_{\text{topic}}(u_t, m) &= p(z(m) \mid u_t), \\ p(z \mid x) &= \text{softmax}_z(\cos(\mathbf{e}(x), \mathbf{t}_z)) \,. \end{aligned} \tag{18}$$

We assign each item a topic label at write time as $z(m) = \arg\max_z p(z \mid c(m))$, and Z_t denotes the number of topics at turn t.

Surprisal components. Let $\hat{s}_i$ be the normalized retrieval weights from Equation (3). We compute a score-weighted evidence embedding

$$\mathbf{e}(\mathcal{C}_t) = \sum_{i=1}^{K} \hat{s}_i \, \mathbf{e}(c(m_i)), \tag{19}$$

and define semantic deviation as a cosine distance:

$$d_{\text{sem}}(u_t, \mathcal{C}_t) = \tfrac{1}{2}\left(1 - \cos(\mathbf{e}(u_t), \mathbf{e}(\mathcal{C}_t))\right) \in [0, 1]. \tag{20}$$

For contradiction, let

$$\pi_t(m) := \text{Verif}_{\text{conf}}(u_t, c(m)) \in [0, 1] \tag{21}$$

be the *calibrated* contradiction probability predicted by a fixed NLI verifier. In all experiments, $\text{Verif}_{\text{conf}}$ is instantiated as the `cross-encoder/nli-deberta-v3-base` cross-encoder: given the premise–hypothesis pair $(c(m), u_t)$, let $\mathbf{z}(u_t, c(m)) \in \mathbb{R}^3$ denote its logits over $\{\text{entail}, \text{neutral}, \text{contr}\}$. We fit a scalar temperature $T > 0$ on the MultiNLI (MNLI) matched development split and define $\pi_t(m) = \text{softmax}(\mathbf{z}(u_t, c(m))/T)_{\text{contr}}$, and set

$$d_{\text{conf}}(u_t, \mathcal{C}_t) = \max_{m_i \in \mathcal{C}_t} \pi_t(m_i) \in [0, 1]. \tag{22}$$

Memory item fields. A memory item m is a typed belief node (`FACT` or `HYPOTHESIS`) with content $c(m)$, confidence weight $w_t(m) \in [0,1]$, and the retrieval metadata listed in Section 2.2; the three metadata fields are constructed as follows. *Entities and keywords* are proposed by the write-path sub-agent at write time (the `Profile-Agent`'s EVIDENCE_KEYWORDS); no separate entity-recognition model is involved, and they are stored as a structured field that is used only by the lexical term s_{fact} of Equation (1). *Topic labels* are assigned at write time as $z(m) = \arg\max_z p(z \mid c(m))$ with the prototypes of Equation (18); the topic table $\mathcal{T}_t$ keeps one running-mean centroid per topic and acts only as the topic prior s_{topic}. *Dense embeddings* are computed solely from the belief text, $\mathbf{e}(c(m))$ with `Qwen3-Embedding-0.6B` (normalized); entity and topic information never enter the embedding and contribute only through their own terms. Episodic traces are embedded from their own dialogue text and serve as provenance only. Section C.11 shows a serialized snapshot.

Cold start. When the belief store is still empty, retrieval returns no evidence and the surprisal terms above are undefined. By convention the first observations of a user are then handled by `PROFILE`: they enter the store as provisional hypotheses that later turns can reinforce, correct, or let decay, exactly as for a populated store. Section C.8 reports how reliably first mentions are written under this convention.

B.3 Operator and gate derivations

`MAINTAIN` **update.** The `MAINTAIN` update in Equation (7) is a single gradient-descent step on the convex consolidation loss $\ell^+(w) = \frac{1}{2}(1-w)^2$. For a scalar weight w,

$$w_{t+1} = w_t - \eta \nabla \ell^+(w_t) = w_t + \eta(1 - w_t),$$

which matches Equation (7). If $w_t \in [0,1]$ and $\eta \in (0,1]$, then $w_{t+1} \in [0,1]$ and the residual shrinks as $1 - w_{t+1} = (1-\eta)(1-w_t)$.

`CORRECT` **update.** For a retrieved belief item m, let $x_t = S_{\text{eff}}(u_t)$ denote the on-policy loss and let $\zeta_t(m) \in [0,1]$ be the per-item severity returned by the `Correct-Agent` (Section D); operationally $\zeta_t(m)$ plays the role of a context-aware contradiction weight, and reduces to a calibrated NLI score $\pi_t(m)$ when the agent is replaced by the fixed verifier in Equation (21). Consider the KL-proximal mirror-descent update:

$$w_{t+1} = \arg\min_{w \geq 0} \; x_t \, \zeta_t(m) \, w + \tfrac{1}{\beta} \mathrm{KL}(w \| w_t), \tag{23}$$

where for scalar weights,

$$\mathrm{KL}(w \| w_t) = w \log \frac{w}{w_t} - w + w_t. \tag{24}$$

Taking the first-order optimality condition of Equation (23) gives $x_t \, \zeta_t(m) + \frac{1}{\beta} \log \frac{w_{t+1}}{w_t} = 0$, hence

$$w_{t+1} = w_t \exp(-\beta x_t \, \zeta_t(m)) \,, \tag{25}$$

which matches Equation (10).

Two-threshold gate. We interpret the gate in Equation (6) as selecting an action that minimizes an on-policy risk plus an action cost:

$$a_t = \arg\min_{a \in \{\mathrm{M},\mathrm{P},\mathrm{C}\}} \widehat{J}_t(a), \qquad \widehat{J}_t(a) = \widehat{L}_t(a) + c(a). \tag{26}$$

Using the proxy $\widehat{L}_t(a) = x_t - \Delta_a(x_t)$ with a linear improvement model $\Delta_a(x) \approx g_a x$ yields

$$\widehat{J}_t(a) \approx (1 - g_a) x_t + c(a).$$

The action boundaries are given by pairwise equality of these lines: $\widehat{J}_t(\mathrm{P}) = \widehat{J}_t(\mathrm{M})$ gives

$$x_t = \frac{c(\mathrm{P}) - c(\mathrm{M})}{g_{\mathrm{P}} - g_{\mathrm{M}}} =: \theta_{\text{low}},$$

and $\widehat{J}_t(\mathrm{C}) = \widehat{J}_t(\mathrm{P})$ gives

$$x_t = \frac{c(\mathrm{C}) - c(\mathrm{P})}{g_{\mathrm{C}} - g_{\mathrm{P}}} =: \theta_{\text{high}},$$

Algorithm 1 PreCoMem per-turn pipeline (hybrid execution). Deterministic steps (retrieval, ρ_t, S_{raw}, gate, w updates, lifecycle) are interleaved with three regime-specific LLM sub-agents that supply per-item targets and bounded scalar scores. Equation references point to the main text and, for d_{sem} and d_{conf}, to Section B.2.

Require: $(u_t,\, Q_{t-1},\, M_t)$ with $M_t = (B_t, E_t, T_t)$
Ensure: $(y_t,\, M_{t+1})$
// Phase A: Memory Leveraging (deterministic)
$(C_t, s_t) \leftarrow \textsc{Retrieve}(M_t, u_t)$ ▷ Equation (1): w_t-modulated tri-signal
$y_t \leftarrow \textsc{Respond}(u_t, Q_{t-1}, C_t)$ ▷ Equation (2)
// Phase B: Reliability + Surprisal (deterministic)
$\rho_t \leftarrow 1 - \textsc{NormalizedEntropy}(s_t)$ ▷ Equation (3)
$d_{\text{sem}} \leftarrow 0.5\,(1 - \cos(e(u_t), e(C_t)))$ ▷ Equation (20)
$d_{\text{conf}} \leftarrow \max_{m \in C_t} \text{NLI_CONTRADICT}(u_t, c(m))$ ▷ Equation (22)
$S_{\text{raw}} \leftarrow \lambda\, d_{\text{sem}} + (1-\lambda)\, d_{\text{conf}}$ ▷ Equation (4)
$x_t \leftarrow \rho_t \cdot S_{\text{raw}}$ ▷ Equation (5)
// Phase C: Regime-gated update (hybrid: agent + formula)
if $x_t \le \theta_{\text{low}}$ **then** ▷ `MAINTAIN` regime
  $\{(m_i, \text{evidence_strength}_i, \text{refinement}_i)\} \leftarrow \textsc{MaintainAgent}(C_t, u_t, x_t)$
  for each $(m_i, \eta_i, \text{ref}_i)$ **do**
    $\eta_t(m_i) \leftarrow \text{evidence_strength}_i$
    $w_{t+1}(m_i) \leftarrow w_t(m_i) + \eta_t(m_i)\,(1 - w_t(m_i))$ ▷ Equation (7)
    if ref_i **then**
      $\textsc{AttachRefinement}(m_i, \text{ref}_i)$
    end if
  end for
else if $x_t \le \theta_{\text{high}}$ **then** ▷ `PROFILE` regime
  $(\text{class}_c, \text{content}, q_t, \text{protect}) \leftarrow \textsc{ProfileAgent}(u_t, C_t, x_t)$ ▷ band $[w^c_{\min}, w^c_{\max}]$ from class_c
  $h \leftarrow \textsc{NewHypothesis}(\text{content},\ \text{protect})$
  $b_t \leftarrow w^c_{\min} + \sigma(x_t)\,(w^c_{\max} - w^c_{\min})$; $\quad w_0(h) \leftarrow \frac{1}{2}\, b_t + \frac{1}{2}\, \text{clip}(q_t, [w^c_{\min}, w^c_{\max}])$ ▷ Equation (8)
  $B_{t+1} \leftarrow B_t \cup \{h\}$
else ▷ `CORRECT` regime
  $\{(m_i, \text{severity}_i, \text{op}_i)\} \leftarrow \textsc{CorrectAgent}(C_t, u_t, x_t)$
  for each $(m_i, \text{severity}_i)$ **do**
    $\zeta_t(m_i) \leftarrow \text{severity}_i$
    $w_{t+1}(m_i) \leftarrow w_t(m_i)\, \exp(-\beta\, x_t\, \zeta_t(m_i))$ ▷ Equation (10)
  end for
  Record the new observation as a fresh belief node.
end if
// Phase D: Hypothesis lifecycle + write-back (deterministic)
for each $h \in B_{t+1}$ with $\tau(h) = \textsc{Hypothesis}$ **do**
  if h not protected and not supported this turn **then**
    $w_t(h) \leftarrow w_t(h)\, \exp(-\gamma\, a_t(h))$, $a_t(h) = t - \ell(h)$ ▷ Equation (9)
  else
    $\ell(h) \leftarrow t$ if supported this turn ▷ reset anchor, skip decay
  end if
  if $w_t(h) \ge \tau_{\text{promote}}$ **then**
    $\tau(h) \leftarrow \textsc{Fact}$
  end if
  if $w_t(h) < 0.1$ **then**
    Prune h
  end if
end for
$E_{t+1} \leftarrow E_t \cup \{(u_t, y_t)\}$ ▷ episodic write-back

which gives the two thresholds:

$$\theta_{\text{low}} = \frac{c(\text{P}) - c(\text{M})}{g_{\text{P}} - g_{\text{M}}}, \qquad \theta_{\text{high}} = \frac{c(\text{C}) - c(\text{P})}{g_{\text{C}} - g_{\text{P}}}. \tag{27}$$

We treat $(c(a), g_a)$ as interpretable design abstractions rather than quantities to be estimated end-to-end: they motivate a *two-threshold* rule and explain why the boundaries should be ordered, but the operating point $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$ is calibrated empirically via the sensitivity sweep in Section C.5, and a personalized data-driven instantiation is given in Section C.5.

`PROFILE` **initialization.** Equation (8) fuses a gate-level prior with the agent's scalar. The class label c returned by the `Profile-Agent` selects the band $I_c = [w_{\min}^{(c)}, w_{\max}^{(c)}]$; the deterministic layer computes the prior $b_t^{(c)} = w_{\min}^{(c)} + \sigma(x_t)\big(w_{\max}^{(c)} - w_{\min}^{(c)}\big)$, clips the agent's INITIAL_CONFIDENCE q_t to I_c, and averages the two. For example, with $I_c = [0.60, 0.70]$, $x_t = 0.4$ and $q_t = 0.65$, one obtains $b_t^{(c)} = 0.6599$ and $w_0(h) = 0.6549$. Because q_t is clipped before it is halved, two agent scalars that differ by the full band width move w_0 by at most 0.05, so the number of `MAINTAIN` updates needed for promotion under Equation (7) changes by at most one, and $w_{\max}^{(c)} < \tau_{\text{promote}}$ guarantees that no single agent output can commit a hypothesis as a `FACT`. The narrow range of $\sigma(x_t)$ inside the band is therefore intentional: the class band carries the semantics, and the within-band value only fine-tunes the starting point. The agent additionally returns the DECAY_PROTECTION flag that governs Equation (9).

Hypothesis decay horizon. Equation (9) is applied once per turn to every unprotected hypothesis that received no supporting evidence in that turn, with the exponent $a_t(h)$ measured from the last evidence-bearing update $\ell(h)$. For a hypothesis that is never reinforced the factors compound, so after n consecutive unsupported turns

$$w_{\ell+n}(h) = w_\ell(h) \exp\Big(-\gamma \frac{n(n+1)}{2}\Big), \tag{28}$$

which is faster than a fixed per-turn factor. This is intended: it instantiates the capacity penalty Ω of Equation (15) (hypotheses are pruned below $w = 0.1$) and is why effective memory plateaus over the dialogue (Section 3.5). With the decay rate $\gamma = 0.05$ used in our experiments, an unsupported latent hypothesis started inside its band is pruned after eight consecutive unsupported turns. The horizon counts only consecutive unsupported turns: a single reinforcement resets $\ell(h)$ and restarts the window, and explicit-event hypotheses are decay-protected, so only guesses that are never revalidated are culled.

B.4 Per-turn pipeline (hybrid execution)

Algorithm 1 makes the hybrid execution discussed in Section 2.4 explicit. Each turn invokes (i) a deterministic `Retrieve` step, (ii) a deterministic computation of $\rho_t, S_{\text{raw}}, x_t$, (iii) a regime-specific LLM sub-agent (one of `Maintain-Agent`, `Profile-Agent`, `Correct-Agent`; prompts in Section D) that emits per-item scalar scores, and (iv) a deterministic application of the closed-form updates in Equations (7), (8) and (10). The sub-agents only inject *which* items to touch and *how strongly*; gating, soft-decay, reinforcement, and hypothesis-promotion mechanics are entirely symbolic.

C Additional experimental details

C.1 Third-party benchmarks: AMemGym and LoCoMo-Plus

To test whether the write-path advantage transfers to data we did not design, we evaluate PreCoMem on two public benchmarks built by other groups: AMemGym [36], an on-policy benchmark with gold state transitions in which a simulated user's latent task state (for example a project's preparation stage or budget band) changes across periods, and LoCoMo-Plus [37], a static benchmark of 401 implicit cognitive-constraint questions. Both are run under their official protocols with the same backbone for all compared systems.

AMemGym. We use the official on-policy `v1.base` release (code snapshot `ffcd188`, data `4b8f64f`) at the official 20-profile / 11-period setting (ten gold state transitions per profile; 943 sessions and 2,200 questions per run) with three paired seeds $(13, 42, 1234)$. The user simulator is *gpt-4.1* (T=0.2); PreCoMem and all of its variants use *gpt-4.1-mini* (T=0) with `text-embedding-3-small`, Top-K=10, $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$ and $\lambda = 0.5$. Each AMemGym session is a single user turn, so one gate event is logged per session. We report the benchmark's two headline scores, the Overall accuracy over the 2,200 questions and the normalized Memory score defined by AMemGym (Overall rescaled between the benchmark's random-answer and upper-bound references), together with its native write / read / utilization failure rates. Table 3 compares PreCoMem with the strongest systems reported in the AMemGym paper under the same *gpt-4.1-mini*

Table 3: **AMemGym** (official `v1.base`, 20 profiles $\times$ 11 periods, *gpt-4.1-mini* for all memory systems). Overall accuracy and normalized Memory score; PreCoMem is mean $\pm$ std over three seeds. Rows marked † are the values reported in the AMemGym paper under the same backbone; AWE was additionally run locally only for the probe metrics quoted in the text.

System	Overall ↑	Memory ↑
No memory (gpt-4.1-mini)†	0.368	0.203
A-Mem†	0.378	0.220
Nemori†	0.385	0.231
Mem0-Graph†	0.424	0.284
AWE†	0.430	0.296
PreCoMem	**0.458** $\pm$ 0.006	**0.329** $\pm$ 0.009

Table 4: **Operator ablation on AMemGym** (mean over three seeds). Top: the benchmark's native scores and failure rates (the three failure rates sum to 1−Overall). Bottom: the state-query probes defined in the text (203 changed-state and 397 stable-state probes per seed). Paired 95% CIs of Full minus each variant exclude zero on every metric.

Variant	Overall↑	Memory↑	Write fail.↓	Read fail.↓	Util. fail.↓
Full PreCoMem	**0.458**	**0.329**	**.300**	.170	.072
w/o `PROFILE`	0.444	0.309	.320	**.165**	**.071**
w/o `CORRECT`	0.439	0.303	.312	.178	**.071**
w/o `MAINTAIN`	0.418	0.273	.310	.190	.082

Variant	Changed-state↑	First-write↑	Stale-state↓	Recovery (periods)↓	Stable-state↑	Old-state suppr.↑
Full PreCoMem	**0.704**	**76.4%**	**10.8%**	**1.0**	**0.762**	**83.1%**
w/o `PROFILE`	0.654	67.8%	14.4%	1.8	0.746	74.5%
w/o `CORRECT`	0.631	70.1%	19.2%	3.0	0.745	64.2%
w/o `MAINTAIN`	0.586	65.5%	21.5%	3.2	0.704	76.5%

backbone: PreCoMem leads the strongest of them (AWE) by $+0.028$ Overall and $+0.033$ Memory, so the advantage reproduces on data we did not design.

To see which operator carries the gain, we remove each operator in turn on the same runs (Table 4). Besides the native scores we track, per seed, 203 probes that query an attribute in the first period after its transition (changed-state probes) and 397 probes on unchanged attributes (stable-state probes). *Changed-state* and *stable-state accuracy* are the accuracies on these two probe sets; *first-write* is the share of changed-state probes for which the belief that dominates retrieval already encodes the new value; *stale-state error* is the share of changed-state probes answered with the superseded value; *recovery* is the mean number of periods until a changed-state probe that was not immediately correct is answered correctly; and *old-state suppression* is the share of changed-state probes for which the superseded belief is absent from the answering Top-10 or ranked below the new one. The three operators leave three distinct signatures, matching Figure 4(c): removing `PROFILE` mainly costs first-write success and changed-state accuracy; removing `CORRECT` raises the stale-state error, stretches recovery from one period to three, and degrades old-state suppression the most; removing `MAINTAIN` degrades stable-state accuracy together with the highest read and utilization failure. Paired 95% confidence intervals of Full minus each variant (three seeds, t-distribution) exclude zero on every metric; for the Memory score they are $0.020\,[0.011, 0.029]$, $0.026\,[0.012, 0.040]$ and $0.056\,[0.040, 0.072]$ for the three variants. Under the same probes, our local run of AWE gives a stale-state error of 15.6% and a recovery of 2.0 periods, against 10.8% and 1.0 for PreCoMem. In w/o `PROFILE`, medium-surprisal observations bypass the waiting room and are committed directly as beliefs; this is the same variant as in Figure 4(c) and Table 5.

Operator ablations of this kind need a gold transition timeline that splits accuracy into stable, change-point and recovery phases. Static question-answering benchmarks such as LoCoMo and LongMemEval-S yield a single end number per run and cannot attribute an effect to one operator, which is why we report operator ablations on AMemGym and DiMoBench and complement them with the route-matched controls of Section C.4.

Table 5: **LoCoMo-Plus by constraint type** (constraint-consistency accuracy, %, GPT-4o for all systems; mean ± std over three seeds; Overall is the question-weighted aggregate over the 401 questions). Baseline overall scores are those reported in the LoCoMo-Plus paper.

Type	n	**PreCoMem**	**w/o** `PROFILE`	Δ
goal	100	22.3 ± 1.1	17.3 ± 2.0	$+5.0$
state	100	22.6 ± 0.9	17.8 ± 1.2	$+4.8$
value	100	21.6 ± 0.8	18.6 ± 0.9	$+3.0$
causal	101	23.0 ± 0.6	21.9 ± 1.0	$+1.1$
Overall	401	**22.4** ± 0.8	18.9 ± 1.2	$+3.5$
Reported overall baselines: full-context GPT-4o 21.05; A-Mem 17.2; Mem0 15.8; SeCom 14.9				

LoCoMo-Plus. LoCoMo-Plus contains 401 questions that probe implicit cognitive constraints of four types (causal 101, state 100, goal 100, value 100) and scores each answer by constraint-consistency accuracy (%), averaged over repeated judgments so that scores are not integer counts. All systems answer with GPT-4o. PreCoMem reaches 22.4 ± 0.8 overall, above A-Mem (17.2), Mem0 (15.8), SeCom [51] (14.9) and even full-context GPT-4o (21.05), as reported in the LoCoMo-Plus paper. Table 5 breaks the score down by type against the w/o `PROFILE` control: the gains are largest on the non-preference types goal ($+5.0$) and state ($+4.8$), smaller on value ($+3.0$), and within seed noise on causal ($+1.1 \pm 0.5$), whose questions depend more on backbone reasoning than on the write path. Together with the task-state targets tracked by AMemGym, this shows the mechanism operating beyond preference modeling: nothing in the effective surprisal, the gate or the closed-form updates is preference-specific, since a belief is any compact claim carrying a confidence weight.

C.2 Backbone-generalization protocol and token usage

Scale-vs-mechanism protocol behind Figure 3(a)–(d). Each x-axis group of Figure 3(a)–(d) fixes the model that answers the questions: Qwen3.5-35B, Qwen3.5-122B or GPT-5-Chat. The two bars of a group differ only in how memory is built and maintained. The left bar runs the answer model with its own native memory, instantiated by O-Mem at the same scale, so every such bar already includes a full memory system. The right bar keeps the answer model unchanged and delegates memory construction (the write path and retrieval) to PreCoMem running on a one-tier-smaller backbone: Qwen3.5-9B for the Qwen3.5-35B group, Qwen3.5-35B for the Qwen3.5-122B group, and GPT-5-mini for the GPT-5-Chat group. The right bar exceeds the left one in every group and on every dataset (all twelve $\Delta > 0$): memory construction can be delegated to a much cheaper model and still beat the larger model's native memory, i.e., with the answer model held equal the mechanism substitutes for roughly one tier of model scale.

Token usage across benchmarks. Table 6 reports the per-query token usage of the response model on all three benchmarks under the same accounting as the *Avg. Tokens* column of Table 1: the mean number of input tokens sent to the response model per query (retrieved memory, short-term window, question and instructions), excluding memory-construction calls. PreCoMem's prompt is bounded by construction (Top-K=10 beliefs, a six-turn window and at most eight episodic excerpts), so its budget grows only from 3.1k on LoCoMo to 3.8k on LongMemEval-S under *gpt-4.1*, and by under 9% when the PersonaMem-v2 context quadruples from 32k to 128k. On every benchmark no baseline is both cheaper and more accurate than PreCoMem, while MemOS and EverMemOS use 0.6–1.4k fewer tokens per query at lower accuracy.

C.3 Extended results and judge validation

LongMemEval-S results under *gpt-4o-mini*. For completeness, Table 7 reports the LongMemEval-S results under the *gpt-4o-mini* backbone (omitted from the main Table 1 to save space). The pattern mirrors *gpt-4.1*: PreCoMem leads on overall accuracy (84.28 ± 0.05, $+1.96$ over the strongest baseline EverMemOS) and on SSA, SSP, MS, and KU. These columns cover assistant-side single-session recall (SSA), preference-conditioned response generation (SSP), cross-session aggregation (MS), and knowledge updates (KU), while EverMemOS retains an SSU lead from its read-path optimization and TiMem leads on temporal reasoning (TR).

Table 6: **Response-model input tokens per query (thousands)** across benchmarks and backbones; PersonaMem-v2 under *gpt-4.1*, averaged over MCQ and Open-Ended queries. Token usage is reported for all nine baselines, including those whose accuracy is not tabulated on a given benchmark in Table 1 or Figure 3. The cross-seed standard deviation is below 0.15k for every entry.

	LoCoMo		**LongMemEval-S**		**PersonaMem-v2**	
Method	gpt-4o-mini	gpt-4.1	gpt-4o-mini	gpt-4.1	32k	128k
MemoryOS	5.5	5.2	7.2	6.8	5.8	7.2
Mem0	1.6	1.8	1.8	2.0	1.9	2.1
A-Mem	2.7	2.9	3.3	3.6	3.2	3.8
GAM	3.5	3.4	4.3	4.2	3.7	4.4
MemBox	3.3	3.4	3.9	4.0	3.6	4.3
MemOS	2.5	2.4	2.7	2.6	2.5	2.7
O-Mem	2.9	2.7	3.7	3.4	3.0	3.6
TiMem	2.8	2.7	3.6	3.5	2.9	3.6
EverMemOS	2.6	2.5	3.2	3.1	2.7	3.2
PreCoMem	3.3	3.1	4.1	3.8	3.5	3.8

Table 7: **LongMemEval-S under *gpt-4o-mini*.** Mean ± std over three seeds; **bold** best, underline second-best. *Avg. Tokens* is the average response-model input token usage per query, as in Table 1. Columns are question types: SSU/SSA/SSP = single-session user/assistant/preference, MS = multi-session, KU = knowledge update, and TR = temporal reasoning.

Method	**Avg. Tokens**	**SSU**	**SSA**	**SSP**	**MS**	**KU**	**TR**	**Overall**
MemoryOS	7.2k	58.21 ± 0.62	59.60 ± 0.92	46.03 ± 0.33	53.68 ± 0.35	54.71 ± 0.23	51.45 ± 0.06	54.85 ± 0.80
MemOS	**2.7k**	70.15 ± 0.57	68.95 ± 0.67	56.02 ± 0.96	62.53 ± 0.17	66.54 ± 0.26	64.70 ± 0.41	64.66 ± 0.78
TiMem	3.6k	81.79 ± 0.93	75.01 ± 0.36	69.99 ± 0.30	72.97 ± 0.91	77.63 ± 0.46	$\mathbf{85.27 \pm 0.08}$	75.51 ± 0.90
GAM	4.3k	77.61 ± 0.94	79.52 ± 0.72	68.18 ± 0.31	76.16 ± 0.43	74.09 ± 0.65	76.50 ± 0.48	76.19 ± 0.83
MemBox	3.9k	79.40 ± 0.67	$\underline{83.26 \pm 0.77}$	$\underline{71.19 \pm 0.46}$	76.16 ± 0.34	75.44 ± 0.53	78.55 ± 0.11	77.37 ± 0.39
O-Mem	3.7k	88.97 ± 0.99	82.28 ± 0.42	70.55 ± 0.45	$\underline{79.61 \pm 0.01}$	$\underline{84.68 \pm 0.12}$	78.08 ± 0.69	81.78 ± 0.36
EverMemOS	$\underline{3.2k}$	$\mathbf{94.10 \pm 0.24}$	79.82 ± 0.20	68.44 ± 0.49	76.98 ± 0.59	82.87 ± 0.42	75.47 ± 0.86	$\underline{82.32 \pm 0.75}$
PreCoMem	4.1k	$\underline{91.58 \pm 0.04}$	$\mathbf{86.38 \pm 0.19}$	$\mathbf{79.40 \pm 0.81}$	$\mathbf{84.84 \pm 0.22}$	$\mathbf{87.13 \pm 0.56}$	$\underline{80.34 \pm 0.73}$	$\mathbf{84.28 \pm 0.05}$

LLM-as-a-judge validation across benchmarks. To validate the reliability of our automated evaluation, we conduct a per-category human–LLM agreement study on all three benchmarks. Table 8 pools the results: across LoCoMo (199-QA stratified audit), LongMemEval-S (full 500 QAs), and DiMoBench (200-QA audit covering one canonical persona), the *gpt-4.1-mini* primary judge agrees with expert human re-annotation at 96.9% overall (93.8–98.5% per question type) with Cohen's κ between 0.81 and 0.96 in every question type (0.957, 0.878 and 0.821 at the dataset level for LoCoMo, LongMemEval-S and DiMoBench). κ is lowest on DiMoBench, which combines the lowest agreement (95.4%) with a high chance-agreement baseline: both raters mark about 85% of its answers correct, so more of the agreement is expected by chance than on LoCoMo, where about 72% are marked correct. To rule out the concern that the judge might be calibrated only on PreCoMem outputs, we additionally re-ran the same human-vs-judge study on 50 outputs each from the two strongest baselines (EverMemOS and TiMem) on LoCoMo and LongMemEval-S; agreement remained within a ± 0.7-point band of the pooled PreCoMem agreement (96.4–97.5%), confirming that the protocol is stable across systems. This supports the use of LLM-as-a-judge in our experiments.

Hypothesis lifecycle statistics. To make self-evolution *observable*, we treat hypotheses as lifecycle objects that can be created, implicitly verified, promoted, or expired (illustrated in Figure 6). Table 9 reports aggregate statistics on LoCoMo (averaged over 10 samples). Removing retrieval-reliability adjustment (w/o C1) increases hypothesis creation but reduces promotion precision, while removing the intermediate `PROFILE` regime (w/o C2) weakens gradual adaptation. Without the hypothesis lifecycle (w/o C3), these lifecycle metrics become undefined, since medium-surprisal information is written as facts directly.

Category-level annotation supporting the attribution. The attribution of the gains in the main text rests on the benchmark annotations. On LoCoMo, Open-Domain is the only split whose questions

Table 8: **Per-category LLM-judge validation on LoCoMo, LongMemEval-S, and DiMoBench.** For each question type we report the LLM judge's verdict on PreCoMem outputs (*gpt-4o-mini* backbone), the expert human re-annotation accuracy on the same outputs, the agreement rate, and Cohen's κ, the chance-corrected agreement between the LLM judge and the human rater. For LongMemEval-S, SSU/SSA/SSP denote single-session user/assistant/preference, MS denotes multi-session, KU denotes knowledge update, and TR denotes temporal reasoning. Following the human-agreement check of Wu et al. [16], we find the *gpt-4.1-mini* primary judge highly consistent with human raters across all three benchmarks; an additional cross-method spot-check on EverMemOS and TiMem outputs (described in the text) confirms the protocol is not calibrated to PreCoMem outputs alone, so the LLM-judged numbers in Table 1 faithfully reflect human assessments.

Dataset	Question Type	#Q	LLM-judge (%)	Human (%)	Agreement (%)	Cohen's κ
LoCoMo	Single-Hop	109	81.41	82.73	98.5	0.949
	Multi-Hop	30	68.49	70.40	98.0	0.953
	Temporal	40	66.21	67.65	98.2	0.959
	Open-Domain	20	39.63	41.55	97.5	0.948
	Overall	199	71.96	73.30	98.3	0.957
LongMemEval-S	SSU	70	91.58	92.86	97.4	0.819
	SSA	56	86.38	87.50	97.8	0.903
	SSP	30	79.40	82.50	95.6	0.858
	MS	133	84.84	86.41	96.9	0.874
	KU	78	87.13	88.46	97.2	0.869
	TR	133	80.34	82.65	95.8	0.861
	Overall	500	84.28	85.78	96.9	0.878
DiMoBench	Current-State Inference	50	86.18	87.83	96.2	0.832
	History Recall	50	87.74	89.62	96.4	0.821
	Change-Point Detection	50	78.31	80.07	93.8	0.812
	Hypothesis vs. Fact	50	83.93	85.61	95.1	0.810
	Overall	200	84.04	85.78	95.4	0.821

Table 9: Hypothesis lifecycle statistics on LoCoMo (averaged over 10 samples).

Config	Created	Promoted	Expired	Promotion rate	Promotion precision
PreCoMem (full)	123	43	81	0.34	0.78
w/o C1	149	42	108	0.28	0.62
w/o C2	138	40	93	0.30	0.70
w/o C3	0	0	0	0.00	–

are inferential in form (20.8% begin with "Would/Might/What would…") and whose gold answers are hedged inferences (14.6%); every other split scores 0.0% on both. Its target is typically a state not stated in a single turn: 49.0% of its questions carry two or more evidence pointers and 32.3% span two or more sessions (mean gap 4.51 sessions). Session span alone does not explain this: Multi-Hop spans more sessions (10.27 on average) yet has no hedged answers, since it joins co-valid stated facts, whereas Open-Domain must infer and revise a latent state. Temporal resolves from a single evidence pointer in 87.5% of cases, which is why no gain appears there. On LongMemEval-S, all 78 KU questions have exactly two answer sessions, an old value and the value superseding it, so an un-updated memory is wrong by construction; SSA questions instead have one answer session whose gold evidence lies in assistant turns (51 of 56), a single target among 48.9 sessions on average, so the SSA gain comes from confidence-weighted retrieval acting as an implicit reranker. We therefore attribute the drift-related gains to KU, SSP, cross-session MS and Open-Domain, and describe SSA as selective memory use.

C.4 Component, operator, and control ablations

Fine-grained ablation of raw-surprisal components. The raw-surprisal term $S_{\text{raw}} = \lambda\, d_{\text{sem}} + (1-\lambda)\, d_{\text{conf}}$ mixes a semantic-gap signal and a verifier-based contradiction signal (Equation (4),

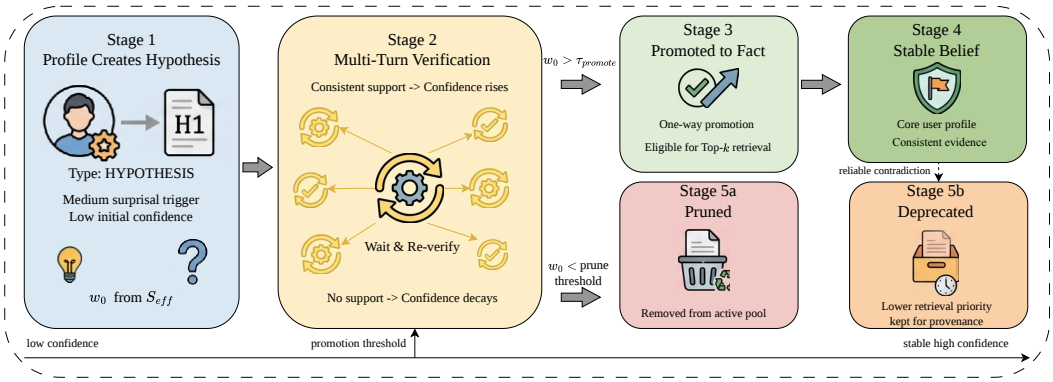

Figure 6: **Hypothesis lifecycle under PreCoMem.** A hypothesis created by `PROFILE` (Stage 1) enters multi-turn verification (Stage 2): consistent support raises its confidence toward the promotion threshold τ_{promote} (Stage 3 $\rightarrow$ Stage 4, becoming a stable core-profile belief that ranks above hypotheses in confidence-weighted retrieval), while no/weak support lets its weight decay; once w_t falls below the prune threshold the hypothesis is removed from the active pool (Stage 5a); a committed `FACT` that is later reliably contradicted is instead softly decayed and kept for provenance (Stage 5b). The horizontal axis is confidence, from low confidence through the promotion threshold to stable high confidence.

Table 10: **Fine-grained ablation of the raw-surprisal components.** Accuracy (%) on LoCoMo and LongMemEval-S (*gpt-4o-mini* backbone). w/o d_{sem} (λ=0) keeps only the NLI-based contradiction signal; w/o d_{conf} (λ=1) keeps only the semantic-gap signal. Both signals matter, but d_{conf} contributes nearly twice as much because it directly detects reliable negation.

Variant	LoCoMo	LongMemEval-S
Full PreCoMem	**73.57**	**84.28**
w/o d_{sem} (λ=0)	67.83 (-5.74)	77.65 (-6.63)
w/o d_{conf} (λ=1)	63.49 (-10.08)	72.71 (-11.57)

Section B.2). To disentangle their individual contributions, we ablate each component by forcing one branch to zero. w/o d_{sem} sets $\lambda = 0$, keeping only the contradiction verifier; w/o d_{conf} sets $\lambda = 1$, keeping only the semantic gap. As shown in Table 10, both components are useful, but d_{conf} contributes substantially more: removing it costs -10.08 points on LoCoMo and -11.57 points on LongMemEval-S, roughly double the drop from removing d_{sem} (-5.74 and -6.63 points respectively). This matches the design intuition: the semantic gap alone cannot distinguish a genuinely contradictory utterance from one that is merely off-topic relative to current memory, whereas the NLI-based contradiction score directly detects reliable negation and drives `CORRECT`. Keeping both signals, on the other hand, gives the best trade-off between sensitivity (detecting drift cues via d_{sem}) and precision (avoiding spurious corrections via d_{conf}).

NLI verifier robustness (sanity check). Because d_{conf} uses a $\max$ statistic over $\pi_t(m)$, a natural concern is whether the system is over-reliant on the specific NLI model behind $\text{Verif}_{\text{conf}}$. We probe this by replacing the default `cross-encoder/nli-deberta-v3-base` with two alternative MNLI-trained verifiers (each re-calibrated with a fresh temperature on the MNLI matched dev split, leaving every other PreCoMem component untouched) and one *random-*π_t control that draws $\pi_t(m) \sim \text{Beta}(2, 5)$ independently per item. To keep the sweep tractable we use LoCoMo Sample 1 (400 turns, 105 QAs; the same sample as the threshold-sensitivity study in Section C.5) under the *gpt-4o-mini* backbone, three seeds.

Table 11 shows that both alternative NLI checkpoints stay within a ± 1 point band of the default; a $\text{Beta}(2, 5)$-randomized π_t, in contrast, costs 11 points, slightly more than removing d_{conf} altogether (-11.11 against -10.08 in Table 10). We read this as a two-sided sanity check: the $\max$-based d_{conf} is genuinely informative (random replacement is much worse than dropping the signal), but the gate is robust to the precise verifier as long as it is calibrated on a standard NLI distribution. Note that the

Table 11: **NLI verifier sanity check on LoCoMo Sample 1.** Sample-1 LLM-judge accuracy (%) when $\mathrm{Verif}_{\mathrm{conf}}$ is replaced. The two alternative MNLI-trained verifiers stay within 1 point of the default; a randomized control collapses by 11 points, confirming that d_{conf} is load-bearing yet that PreCoMem is not coupled to a specific verifier checkpoint.

$\mathrm{Verif}_{\mathrm{conf}}$	Sample-1 Overall	Δ
`cross-encoder/nli-deberta-v3-base` (default)	76.19 ± 0.42	–
`facebook/bart-large-mnli`	75.74 ± 0.38	-0.45
`roberta-large-mnli`	75.40 ± 0.51	-0.79
random $\pi_t \sim \mathrm{Beta}(2,5)$ (control)	65.08 ± 0.79	-11.11

absolute accuracies in this table are on a single 105-QA sample and therefore not comparable to the full-LoCoMo numbers in Table 1; the comparison is a within-row Δ across verifiers.

DiMoBench evolution-operator ablations. We ablate PreCoMem's three evolution operators on DiMoBench: `PROFILE` (hypothesis writing), `CORRECT` (surprisal-gated soft correction), and `MAINTAIN` (reinforcement/smoothing). We report accuracy every 10 turns from turn 150 to 350. As shown in Figure 4(c), removing `PROFILE` lowers the final accuracy ceiling (72%), removing `CORRECT` substantially slows post-drift recovery, and removing `MAINTAIN` leads to high-variance dynamics that do not stabilize within our evaluation horizon (ending at 54% accuracy).

Operator-level analysis. DiMoBench is designed to separate three regimes that stress different update behaviors: a stable pre-change phase (turns ≈150–200), a short change-point phase with mixed/ambiguous cues and decoy retrieval (turns ≈200–240), and a stable post-change phase where the latent attitude has reversed (turns ≈240–350). Under this structure, the three operators contribute in complementary ways.

`PROFILE`**: buffered adaptation under implicit drift.** Many change-point cues are neither clean confirmations nor decisive contradictions: they are novel, partially ambiguous, and often entangled with confounders. `PROFILE` converts these medium-S_{eff} signals into low-confidence, testable hypotheses, enabling the system to accumulate multiple weak cues before committing. This "waiting room" is especially important on DiMoBench because questions explicitly probe trend/change-point reasoning and hypothesis-vs-fact distinction. When `PROFILE` is removed, the system loses this intermediate representation and commits weak cues without a waiting room (premature commitment under decoys), which reduces the asymptotic ceiling (72% vs. 84% for full PreCoMem).

`CORRECT`**: targeted suppression of outdated beliefs.** After the drift, previously correct high-confidence beliefs become systematically misleading, and decoy retrieval makes stale evidence likely to reappear. `CORRECT` provides a selective revision mechanism by downweighting contradicted items via the $\zeta_t(m)$-modulated soft decay while leaving non-contradictory items largely unchanged. This targeted decay is crucial for reducing old-memory pull and preventing interference from pre-change beliefs. Without `CORRECT`, outdated facts remain competitive in retrieval, so the agent must rely on accumulating new nodes and indirect effects to "outvote" entrenched beliefs, yielding a substantially slower recovery (reaching 75% only around turn 320 vs. turn 250) and a lower final plateau (77% vs. 84%).

`MAINTAIN`**: consolidation for stability.** `MAINTAIN` acts as a consolidation step that repeatedly reinforces supported beliefs, which sharpens confidence-weighted retrieval, increases retrieval reliability ρ_t (by making the score mass more concentrated), and stabilizes the routing signal $x_t (= S_{\mathrm{eff}})$. It also underpins the hypothesis lifecycle: hypotheses created by `PROFILE` require repeated reinforcement to cross the promotion threshold and become stable facts. This stabilization is critical in the post-change phase, where the system should stop oscillating and settle into the post-change regime. When `MAINTAIN` is ablated, supported post-change beliefs cannot be progressively consolidated; retrieval remains diffuse and volatile, the reliability-adjusted loss becomes noisy, and updates keep "chasing" confounded evidence. Empirically, this yields persistent volatility: accuracy continues to fluctuate and ends at 54% at the end of our evaluation.

Route-matched and factorial controls. Setting the gate score to zero (ES in Figure 4(a)) routes every turn to `MAINTAIN`, so it measures a reinforce-only lower bound; likewise CW sets $w \equiv 1$

Table 12: **Route-matched and factorial controls** (LLM-judge accuracy, *gpt-4o-mini*, mean ± std over three seeds).

Variant	LoCoMo	LongMemEval-S
Full PreCoMem	**73.57** ± 0.28	**84.28** ± 0.05
Single-threshold gate (no `PROFILE` band; medium band → `CORRECT`)	69.92 ± 0.29	81.16 ± 0.12
Frequency-matched random routing (M/P/C = 56.2/27.4/16.4%)	61.83 ± 1.25	72.94 ± 1.08
Read-time weighting off, lifecycle confidence on	71.80 ± 0.43	83.56 ± 0.23
Lifecycle confidence off, read-time weighting on	52.88 ± 0.69	61.65 ± 0.68
Both off (= w/o CW)	43.42 ± 0.94	49.65 ± 0.91

Table 13: **LLM-direct-write baseline** (Overall LLM-judge accuracy, *gpt-4.1*, mean ± std over three seeds; the paired difference is the mean over seeds of PreCoMem minus direct write).

	LoCoMo	LongMemEval-S
PreCoMem (bounded scalars + closed-form updates)	**85.06** ± 0.07	**89.39** ± 0.76
LLM direct write	82.31 ± 1.42	86.24 ± 1.63
Paired difference	+2.75	+3.15

globally, so its drop is the total effect of confidence weighting across retrieval and the whole lifecycle (every hypothesis then trivially satisfies $w \geq 0.80$ and is promoted at its first lifecycle step, pruning never fires, and soft decay is overwritten). Table 12 isolates the marginal value of the gate and the two roles of confidence with three controls on LoCoMo and LongMemEval-S (*gpt-4o-mini*, three seeds). (i) A *single-threshold* gate without the `PROFILE` band, which sends the medium band to `CORRECT`, reaches 69.92 / 81.16 (-3.65 / -3.12 against Full); this is the marginal value of the waiting room. (ii) *Frequency-matched random routing* keeps the operator budget at Full's empirical routing frequencies (M/P/C = $56.2/27.4/16.4\%$) and leaves the read path unchanged, but assigns operators at random; it loses 11.74 / 11.34 with the update frequency held fixed, so the gain comes from the evidence-to-operator correspondence itself. (iii) A 2×2 factorial over read-time weighting (w in Equation (1)) and lifecycle confidence (w in the operator updates and in the promotion and pruning tests) shows that removing read-time weighting alone costs 1.77 / 0.72 points, removing lifecycle confidence alone costs 20.69 / 22.63, and removing both (the CW variant) costs 30.15 / 34.63; in half-coded effect terms the main effects are $+5.62$ / $+6.36$ for read-time weighting and $+24.54$ / $+28.27$ for lifecycle confidence, with an interaction of -3.85 / -5.64, i.e., each factor is worth more when the other is absent.

LLM-direct-write baseline. A natural alternative to the hybrid design of Section 2.1 is to let the same LLM decide the whole write step. The direct-write baseline matches PreCoMem in every other component (retrieval, embedding, Top-K=10, context, judge, seeds, *gpt-4.1*); the only change is that the LLM emits the entire write decision as one object, {`route`, `target_ids`, `operation`, `new_weights`, `new_belief`}, whose weights are written verbatim, replacing the gate, the closed-form updates and the $[0, 1]$ clamping. Table 13 shows that accuracy is lower in every seed (-2.75 on LoCoMo, -3.15 on LongMemEval-S) while per-query tokens are slightly higher (3.2k / 3.9k against 3.1k / 3.8k), so the loss is not offset by efficiency. The more telling difference is stability: the cross-seed standard deviation is $20\times$ higher on LoCoMo (1.42 against 0.07) and $2.1\times$ higher on LongMemEval-S, because unconstrained numeric decisions make the memory trajectory run-dependent. Replaying the write decisions on a frozen store, direct write emits a missing target id in 3.6% of decisions, a self-contradictory operation in 2.7% and invalid JSON in 1.8% (mutually exclusive categories); such malformed writes corrupt the store silently and compound, whereas the closed-form updates stay in $[0, 1]$ by construction. The loss is spread over all LoCoMo splits (Single-Hop -1.65, Multi-Hop -4.80, Temporal -2.70, Open-Domain -6.50). The deterministic layer is what keeps long-horizon memory bounded.

Turn-level versus per-item routing. PreCoMem computes one gate score per turn and routes the whole retrieved set to a single operator; the per-item magnitudes $\eta_t(m)$ and $\zeta_t(m)$ apply after routing. A turn can therefore reinforce several beliefs, or correct several beliefs, at once, but it cannot

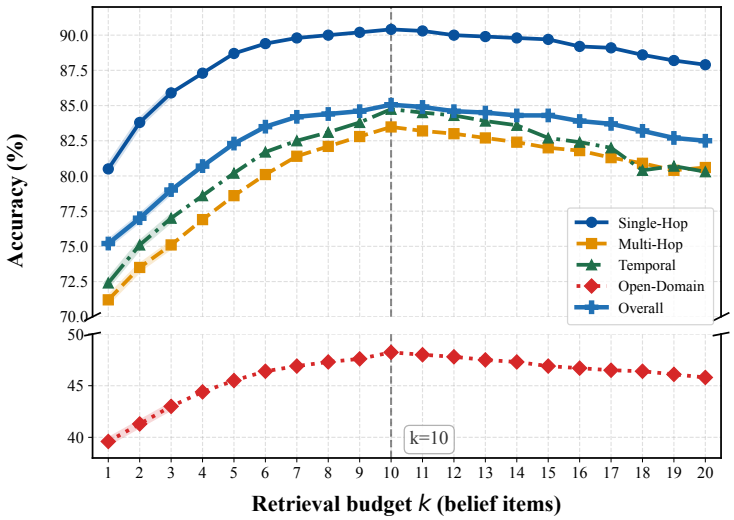

Figure 7: **Top-K sensitivity on LoCoMo (*gpt-4.1* backbone).** Per-split accuracy as k varies in $\{1, \ldots, 20\}$. The dashed vertical line marks our default k=10.

Table 14: Accuracy (%) at different retrieval sizes k on LoCoMo (*gpt-4.1* backbone). Best results are in **bold**.

k	Single-Hop	Multi-Hop	Temporal	Open-Domain	Overall
1	80.5	71.2	72.4	39.6	75.2
5	88.7	78.6	80.2	45.5	82.3
10	**90.41**	**83.48**	**84.74**	**48.23**	**85.06**
15	89.7	82.0	82.7	46.9	84.3
20	87.9	80.6	80.3	45.8	82.5

reinforce one belief, correct another and profile a third in the same step. To measure how often this matters we annotated gold operators per retrieved item on 9,006 turns (all LoCoMo dialogues plus the LongMemEval-S evidence sessions): 3,475 turns touch at least one retrieved belief, and 95.83% of them are expressible by a single route, either a single belief or several beliefs sharing one operator. The remaining 145 turns (1.61% of all turns, 21.4% of the 678 multi-item turns) are genuinely mixed. A per-item variant, identical in retrieval, backbone and closed-form updates but routing each item independently, raises the whole-turn agreement with the gold operators on these 145 turns by $+5.5$ points and their question accuracy by $+3.5$ points, at $+31\%$ write calls and $+14\%$ write tokens, for a net gain of $+0.3$ points over the full evaluation. Given the small net benefit at a materially higher cost, we keep turn-level routing as the default and regard per-item claim extraction as a direction for future work.

C.5 Hyperparameter sensitivity and adaptive variants

Sensitivity to retrieval budget k. We vary the number of retrieved memory items k on LoCoMo. Table 14 shows a clear "fast gain, slow gain, mild drop" trend: Overall accuracy improves substantially from k=1 to k=5 (75.2→82.3), increases marginally from k=5 to k=10 (82.3→85.06), and then declines at larger budgets (85.06→82.5 at k=20). Category-wise, Temporal is the most sensitive to oversized budgets (84.7 at k=10 vs. 80.3 at k=20). Figure 7 visualizes the same sweep across all four LoCoMo splits with a broken-y axis to keep the high- and low-accuracy bands legible side-by-side.

Mechanism-level analysis. We provide a more formal explanation of why oversized k can hurt PreCoMem beyond generic retrieval noise. Recall that PreCoMem uses entropy-based retrieval reliability ρ_t and the max-contradiction term d_{conf}:

$$\begin{aligned} \hat{s}_i &= \frac{s_i + \epsilon}{\sum_{j=1}^{k}(s_j + \epsilon)}, \\ H_t &= \frac{-\sum_{i=1}^{k} \hat{s}_i \log \hat{s}_i}{\log k}, \\ \rho_t &= 1 - H_t. \end{aligned} \tag{29}$$

$$\begin{aligned} S_{\text{eff}}(u_t) = x_t &= \rho_t \Big(\lambda \, d_{\text{sem}}(u_t, \mathcal{C}_t) + (1-\lambda) \, d_{\text{conf}}(u_t, \mathcal{C}_t) \Big), \\ d_{\text{conf}} &= \max_{m \in \mathcal{C}_t} \pi_t(m). \end{aligned} \tag{30}$$

(1) $k \uparrow$ flattens scores $\Rightarrow \rho_t \downarrow$ and `CORRECT` is suppressed. The reliability ρ_t decreases when the normalized score distribution $\hat{\mathbf{s}}$ becomes less concentrated. To make this explicit, consider a simple extension model where the Top-k distribution over the original k_0 items is $\mathbf{p} \in \Delta^{k_0-1}$, and increasing k adds k_1 additional tail items whose total normalized mass is $\delta \in (0,1)$ and is approximately uniform within the tail:

$$\mathbf{p}' = \big((1-\delta)p_1, \ldots, (1-\delta)p_{k_0}, \underbrace{\tfrac{\delta}{k_1}, \ldots, \tfrac{\delta}{k_1}}_{k_1 \text{ items}} \big), k = k_0 + k_1. \tag{31}$$

By concavity of Shannon entropy, injecting nonzero tail mass strictly increases entropy unless $\delta = 0$. A direct decomposition yields

$$\mathcal{H}(\mathbf{p}') = (1-\delta)\mathcal{H}(\mathbf{p}) + h(\delta) + \delta \log k_1, \tag{32}$$

where $\mathcal{H}(\cdot)$ is unnormalized entropy and $h(\delta) = -\delta \log \delta - (1-\delta)\log(1-\delta)$ is the binary entropy. Thus, for typical retrieval where the tail receives non-negligible probability mass, H_t increases with k and ρ_t decreases.

This interacts with the fixed-threshold gate: `CORRECT` is triggered when $x_t = S_{\text{eff}}(u_t) > \theta_{\text{high}}$. Since $x_t = \rho_t S_{\text{raw}}(u_t)$ with $S_{\text{raw}}(u_t) \in [0,1]$, the event can be rewritten as

$$\rho_t S_{\text{raw}}(u_t) > \theta_{\text{high}} \quad \Longleftrightarrow \quad S_{\text{raw}}(u_t) > \frac{\theta_{\text{high}}}{\rho_t}. \tag{33}$$

Therefore, as ρ_t decreases, the required raw-surprisal level $\theta_{\text{high}}/\rho_t$ increases, making reliable correction harder to trigger and strengthening old-memory pull. This effect is most visible for Temporal queries, where Top-k often mixes pre- and post-change evidence, yielding a less concentrated score mass (higher H_t) even when the underlying contradiction is real.

(2) $k \uparrow$ amplifies spurious contradictions via a max statistic. The contradiction term uses an order statistic $d_{\text{conf}} = \max_{m \in \mathcal{C}_t} \pi_t(m)$. If we view $\pi_t(m)$ for *irrelevant/mismatched* retrieved items as random variables with CDF F_0 on $[0,1]$, then for k candidates,

$$\Pr(d_{\text{conf}} > \tau) = 1 - \Pr(\pi_t(m) \le \tau)^k = 1 - F_0(\tau)^k, \tag{34}$$

which is monotonically increasing in k for any $\tau \in (0,1)$ with $F_0(\tau) \in (0,1)$. When $1 - F_0(\tau)$ is small, the increase is approximately linear:

$$1 - F_0(\tau)^k \approx k \, (1 - F_0(\tau)). \tag{35}$$

Thus, larger evidence sets increase the probability of observing an artificially high contradiction score from at least one mismatched memory, which can (i) inflate S_{raw} through the $(1-\lambda)d_{\text{conf}}$ component and (ii) distract response generation by presenting a "hard contradiction" that is not actually relevant to the query. Together with (1), this explains why performance can decline for large k despite higher coverage.

Implication. The empirical peak at k=10 reflects a regime where coverage is sufficient to retrieve the right evidence, while ρ_t remains high enough to permit timely `CORRECT` actions and the max-contradiction term remains less exposed to spurious outliers. This motivates using k=10 as a robust default under a moderate compute budget.

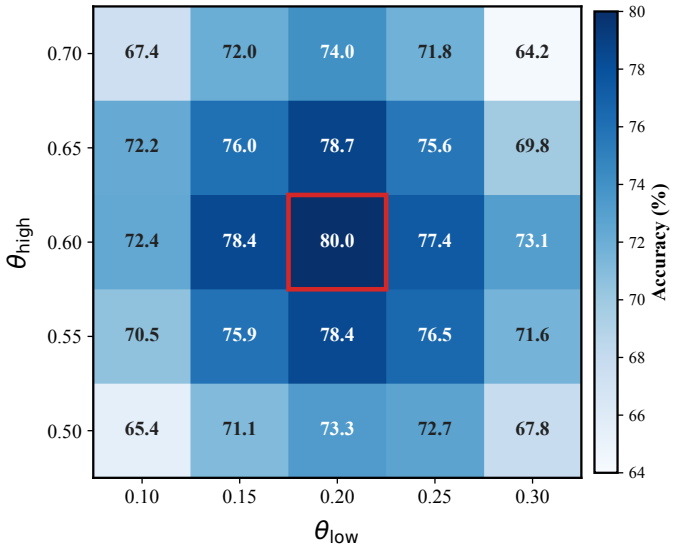

Figure 8: **Gating threshold sensitivity.** LoCoMo accuracy (Sample 1) over a 5×5 grid of $(\theta_{\text{low}}, \theta_{\text{high}})$ with step 0.05 (25 configurations in total). The red box marks the deployed default $(0.20, 0.60)$, which sits at the center of a high-performing plateau; the four immediate neighbors of the peak stay within 1.3–2.6 accuracy points, while accuracy degrades smoothly toward the grid corners.

Sensitivity to gating thresholds. Theoretically, one may dynamically compute or adapt the thresholds (e.g., by estimating the action costs/effects in Equation (27)), which could yield better per-sample or per-stage performance. However, such on-policy adaptation introduces additional computation and confounds controlled evaluation because the observed behavior becomes entangled with an evolving threshold schedule.

We therefore adopt a *fixed* threshold pair $(\theta_{\text{low}}, \theta_{\text{high}})$ in all experiments and justify a reasonable choice via a sensitivity study. For controllable and cost-bounded ablation, we select a representative LoCoMo sample (Sample 1; 400 dialogue turns and 105 QA pairs) whose question types are relatively balanced and whose scale supports rapid iteration. We sweep $\theta_{\text{low}} \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$ and $\theta_{\text{high}} \in \{0.50, 0.55, 0.60, 0.65, 0.70\}$ in steps of 0.05 ($5\times5 = 25$ configurations) and evaluate LLM-judge accuracy on this sample. As shown in Figure 8, the deployed default $(0.20, 0.60)$ (red box) sits at the center of a high-performing plateau: its four direct neighbors $(0.15, 0.60)$, $(0.25, 0.60)$, $(0.20, 0.55)$ and $(0.20, 0.65)$ all stay within a narrow 1.3–2.6-point band of the peak. Accuracy decays smoothly toward the corners (down to roughly 64–67% near θ_{low}=0.30, θ_{high}=0.70 and the symmetric corner), confirming that the gate is locally robust rather than knife-edged.

Importantly, this fixed choice is a trade-off: we do not claim $(0.2, 0.6)$ is universally optimal for every LoCoMo sample. Rather, it provides a stable and interpretable operating point with competitive accuracy under a fixed computational budget, and we keep it constant across experiments for fair comparison and reproducible observations.

Adaptive thresholding for personalization. One natural question is whether the fixed gating thresholds $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$ are appropriate for all users, or whether different user populations would benefit from different thresholds. Intuitively, users differ in *belief stability*: some keep stable preferences so that short-term deviations are noise, while others change frequently so that the same deviations reflect real drift. A fixed threshold cannot simultaneously satisfy both regimes. We therefore discuss and evaluate a simple adaptive-thresholding extension of PreCoMem.

User stability index. We summarize each user's recent dynamics with a scalar *stability index* $\psi_t \in [0, 1]$, computed from interaction statistics over a sliding window of length W:

$$\psi_t = \omega_1 \psi_{\text{corr}} + \omega_2 \psi_{\text{life}} + \omega_3 \psi_{\text{var}}, \tag{36}$$

Table 15: **Adaptive vs. fixed thresholds on LoCoMo** (accuracy %, *gpt-4o-mini* and *gpt-4.1*). PreCoMem (DT) personalizes $(\theta_{\text{low}}, \theta_{\text{high}})$ via the user stability index in Equations (36) and (37); PreCoMem (fixed) uses $(0.2, 0.6)$. DT improves the overall average on both backbones, with the largest gain on the drift-heavy *Open* category under *gpt-4o-mini*, without any retraining.

Backbone	Method	Single	Multi	Temp.	Open	Avg.
gpt-4o-mini	PreCoMem (fixed)	81.41	68.49	66.21	39.63	73.57
	PreCoMem (DT)	**82.27**	**69.31**	**67.04**	**43.55**	**74.55**
gpt-4.1	PreCoMem (fixed)	90.41	83.48	84.74	**48.23**	85.06
	PreCoMem (DT)	**91.42**	**85.16**	**85.81**	47.78	**86.21**

where (i) $\psi_{\text{corr}} = 1 - N^{(W)}_{\text{correct}}/N^{(W)}_{\text{total}}$ captures how rarely `CORRECT` fires in the recent window, (ii) $\psi_{\text{life}} = \min\big(1, \bar{L}_{\text{belief}}/L_{\text{ref}}\big)$ captures average belief lifespan relative to a reference L_{ref}, and (iii) $\psi_{\text{var}} = \exp\Big(-\kappa \cdot \text{Var}\Big[S^{(W)}_{\text{eff}}\Big]\Big)$ captures the variance of effective surprisal. The coefficients satisfy $\omega_1 + \omega_2 + \omega_3 = 1$, and we use $\omega_1{=}0.5, \omega_2{=}0.3, \omega_3{=}0.2$ in the experiments below (κ and ω_i are specific to this variant and unrelated to the γ and α_j of the main text).

Threshold adaptation. We then map ψ_t to thresholds by linear interpolation between a conservative upper configuration and an aggressive lower configuration:

$$\begin{aligned} \theta_{\text{low}}(t) &= \theta^{\min}_{\text{low}} + \psi_t\big(\theta^{\max}_{\text{low}} - \theta^{\min}_{\text{low}}\big)\,, \\ \theta_{\text{high}}(t) &= \theta^{\min}_{\text{high}} + \psi_t\big(\theta^{\max}_{\text{high}} - \theta^{\min}_{\text{high}}\big)\,, \end{aligned} \tag{37}$$

with $\theta_{\text{low}} \in [0.10, 0.35]$ and $\theta_{\text{high}} \in [0.40, 0.80]$. Intuitively, a stable user ($\psi_t \to 1$) receives larger thresholds, so the system requires stronger evidence before updating; a dynamic user ($\psi_t \to 0$) receives smaller thresholds, so the system reacts faster. To avoid oscillation, we smooth the thresholds with an exponential moving average with coefficient $\mu{=}0.1$.

Theoretical view. From a Bayesian perspective, the thresholds encode a prior on belief persistence: stable users have a stronger prior that beliefs are unchanged, so a higher θ_{high} corresponds to requiring stronger likelihood evidence to revise them, and vice versa for dynamic users. From a signal detection perspective, θ_{low} and θ_{high} trade off false-update rate (triggering `CORRECT` on noise) against missed-update rate (failing to trigger on genuine drift); since these two error rates have different relative costs for different users, the Bayes-optimal decision boundary is user-dependent. Equation (37) is a minimal realization of this idea that only reads summary statistics PreCoMem already maintains and introduces no extra trainable parameters.

Empirical results on LoCoMo. We evaluate the adaptive variant, denoted **PreCoMem (DT)**, on the full LoCoMo benchmark with two backbones (*gpt-4o-mini* and *gpt-4.1*), keeping every other setting identical to the main experiments. As shown in Table 15, relaxing the fixed thresholds to Equation (37) yields consistent gains in overall accuracy ($+0.98$ on *gpt-4o-mini*, $+1.15$ on *gpt-4.1*), with the largest improvement on the *Open-Domain* subset under *gpt-4o-mini* ($+3.92$), which is the category most exposed to long-horizon drift. The same trend holds on LongMemEval-S, where PreCoMem (DT) lifts the overall score from 84.28 to 85.36 under *gpt-4o-mini*, with the largest gains on **KU** (knowledge update, $+1.62$) and **SSP** (single-session preference, $+1.95$); these are precisely the question types that require per-user calibration of how aggressively to update beliefs.

These results show that adaptive thresholding is an almost-free extension that further personalizes PreCoMem for users with different stability profiles. We keep the fixed-threshold setting as the default in the main paper for controllable comparison, but we consider adaptive thresholding a natural deployment-time configuration, especially in long-horizon and cross-user settings.

C.6 DiMoBench: construction and extended-horizon stress test

Construction pipeline and characteristics. DiMoBench (Dialogue Implicit Memory Drift Benchmark) is designed to isolate *belief-drift adaptation* under sparse and implicit cues. The released suite consists of *five* 350-turn two-speaker dialogues, each built around a distinct persona drawn from a

Table 16: **DiMoBench construction pipeline.** Generator and verifier are deliberately different models; a 20% stratified human review provides final quality control.

Stage	Name	Model	Output
1	Persona construction	GPT-4o	User profile + core trait slot
2	Life skeleton & turn	GPT-4o	Event timeline + drift trigger
3	Dialogue generation	GPT-4o	35 sessions $\times$ 10 turns
4	QA generation	GPT-4o	200 QA pairs across 4 categories
5	Consistency verification	Gemini-2.5-Pro	Cross-validated QA (6 criteria)
6	Human–machine review	Human + LLM	20% stratified audit, $\kappa \geq 0.75$

different combination of Big-Five extremes and life stages; for clarity the exposition below focuses on one canonical persona, but every quantitative result on DiMoBench (the dynamics in Figure 4(b) and the operator ablation in Figure 4(c)) is averaged across the five personas to avoid a single-persona artifact; the human–judge agreement audit in Table 8 uses the canonical persona, whose per-type accuracies lie within 0.15 points of the five-persona means. Within each persona the user's latent attitude toward a target topic remains stable in turns 1–200, gradually reverses in turns 201–210 via 3–4 implicit events, and is then reinforced with denser behavioral signals in turns 211–240 before entering a low-frequency maintenance stage in turns 241–350.

To stress long-horizon robustness, DiMoBench injects confounders: every 30 turns we insert irrelevant chit-chat or hypothetical narration that shares surface keywords with the target topic, and we introduce multiple similar entity names to induce attribution ambiguity. Each persona contributes 200 QA pairs distributed across the dialogue (denser sampling near the change point and sparser in the long maintenance tail; $1{,}000$ in total across the five personas), covering (i) current-state inference, (ii) history recall, (iii) change-point detection, and (iv) hypothesis vs. fact distinction. Importantly, questions avoid explicit sentiment keywords and instead require reasoning from behavioral cues (plans, actions, reactions), making drift detection and correction rely on implicit evidence chains rather than lexical matches.

Why DiMoBench is needed. Existing long-horizon dialogue benchmarks predominantly evaluate the *read path*: whether a system can correctly retrieve and reason over a static conversational record (LoCoMo [34]) or whether it can infer implicit personalization from accumulated preferences (PersonaMem-v2 [35]). None of them explicitly stress-test the *write path*, i.e., whether a memory system can correctly decide *when to maintain, update, or correct* its stored beliefs under gradual, implicit drift. DiMoBench is designed to fill this gap along three axes that are absent from existing benchmarks simultaneously: (i) *continuous non-stationarity*, where the latent attitude reverses gradually over hundreds of turns without any explicit announcement; (ii) *implicit behavioral cues*, where the drift manifests only through activity choices, reactions to invitations, and hedging language rather than sentiment keywords; and (iii) *adversarial retrieval interference*, where five tagged interference types (`entity_confusable`, `keyword_trap`, `temporal_decoy`, `hypothetical_vs_fact`, `none`) systematically probe two stability–plasticity failure modes: *old-memory pull* and *premature correction*. This write-path orientation is what lets DiMoBench reveal the distinct contributions of `MAINTAIN`, `PROFILE`, and `CORRECT` in Section C.4, which cannot be isolated on LoCoMo or PersonaMem-v2 alone.

Six-stage construction pipeline. DiMoBench is constructed by a six-stage progressive pipeline with an explicit *generator–verifier separation* (GPT-4o for generation, Gemini-2.5-Pro for verification) to avoid self-reinforcing biases. Figure 9 visualizes the full pipeline as an end-to-end flow, and Table 16 summarizes the stages; details follow below.

Stage 1 (Persona). Each persona is characterized by one *core emotional/behavioral slot* (e.g., "attitude toward social activities") with a well-defined initial state and target state, a set of Big-Five personality dimensions, a small social circle of 2–3 named characters (some deliberately confusable to enable interference evaluation), routine daily activities, and a psychologically plausible drift trigger.

Stage 2 (Life skeleton). The 350-turn trajectory is organized into 35 sessions of 10 turns each. Sessions 1–20 (turns 1–200) establish baseline behavior; session 21 (turns 201–210) introduces the drift trigger via 3–4 implicit events; sessions 22–24 (turns 211–240) reinforce the new behavior

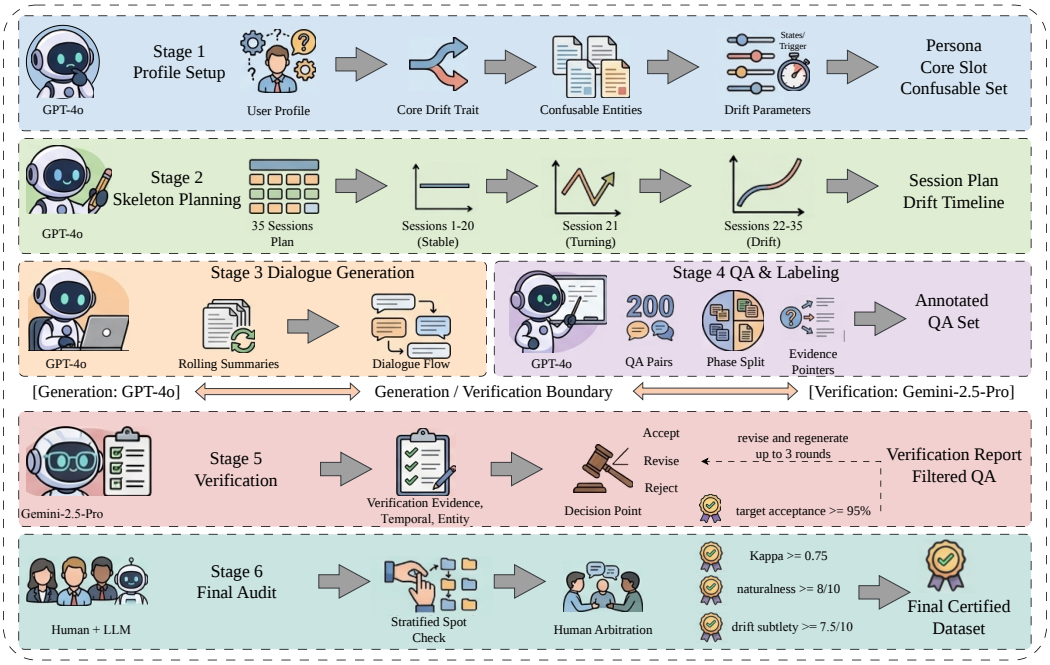

Figure 9: **DiMoBench six-stage construction pipeline.** A generator (GPT-4o) produces personas, life skeletons, dialogues, and QA pairs; an independent verifier (Gemini-2.5-Pro) cross-validates each QA pair along six criteria, and a 20% stratified human audit with $\kappa \geq 0.75$ provides final quality control.

with denser cues; sessions 25–35 (turns 241–350) form a low-frequency maintenance stage with monotonically reduced divergence from a steady post-drift baseline.

Stage 3 (Dialogue generation). Sessions are generated sequentially under cumulative context management. Generation rules enforce natural small talk and topic shifts, implicit expression through activity choices and hedging language, no meta-commentary (the user never says "I've changed"), and frequent temporal markers ("last weekend", "this month") for temporal reasoning.

Stage 4 (QA generation). 200 QA pairs span all temporal phases (pre-change: 90; change-point: 10; post-change: 100). Each QA carries explicit evidence pointers (`D{session}:{turn}`), an interference tag, and a reasoning-type label.

Stage 5 (Consistency verification). Each QA is cross-validated by Gemini-2.5-Pro along six dimensions: (i) *answer grounding* (is the answer supported by cited evidence?), (ii) *evidence completeness* (are all relevant turns cited?), (iii) *temporal consistency* (does the answer reflect the correct phase?), (iv) *entity accuracy*, (v) *interference validity* (does the tag match the question?), and (vi) *drift coherence* (is the behavioral change gradual and plausible?). QA pairs marked `REVISE` are regenerated by GPT-4o using verifier feedback (max 3 rounds); pairs marked `REJECT` are discarded. We target an $\geq 95\%$ `ACCEPT` rate on the auto-verification pass.

Stage 6 (Human–machine review). Three graduate-level annotators with NLP research experience underwent a 2-hour calibration session covering implicit emotional expression, temporal reasoning, and entity disambiguation. All annotators independently label the same 20 QA pairs, and independent annotation proceeds only after inter-annotator agreement reaches $\kappa \geq 0.75$ (Cohen's κ). A 20% stratified sample (stratified by temporal phase, interference type, and QA category) receives full human review; in addition, one annotator per persona performs a full 350-turn dialogue-coherence assessment. Quality targets are answer correctness $\geq 95\%$, dialogue naturalness $\geq 8.0/10$, and drift subtlety $\geq 7.5/10$.

Representative QA examples. To make DiMoBench concrete, we include three canonical QA pairs illustrating each temporal phase and several interference tags:

Pre-change QA (no interference, `MAINTAIN`-oriented)

Q: What kind of activities does Mike like?
A: He likes group activities like KTV, parties, and escape rooms.
Evidence: `D1:3` **Phase:** pre-change **Interference:** none **Category:** current-state inference

Change-point QA (implicit behavior, `PROFILE`-oriented)

Q: Is Mike really that busy or is it an excuse?
A: Work is indeed busy, but previously he wouldn't decline because of this.
Evidence: `D21:5`, `D21:7` **Phase:** change-point **Interference:** none **Category:** change-point detection

Post-change QA (temporal decoy, `CORRECT`-oriented)

Q: Mike previously said he likes lively places; is that still true?
A: Not anymore; his preference has changed.
Evidence: `D7:5`, `D27:9` **Phase:** post-change **Interference:** `temporal_decoy` **Category:** current-state inference

Additional traces that walk through the operator decisions on DiMoBench, including the retrieved Top-K evidence and the resulting memory edits, are provided in Section C.11. All five 350-turn dialogues, the per-persona QA pairs, generation prompts, verification prompts, and human-review scripts are released with the paper at `https://github.com/ZhongWishing/PreCoMem`.

DiMoBench-XL construction. To stress-test the system beyond DiMoBench's 350-turn horizon and across more personas, we constructed **DiMoBench-XL**, a suite of *five* 1050-turn dialogues, each with three sequential implicit drifts. Following the DiMoBench pipeline in Section C.6, we instantiate five user personas (each with a distinct demographic and dispositional profile, sampled to span both Big-Five extremes and different life stages); within each persona we keep five independent attributes (drink preference, exercise habit, home city, employment status, social style) and apply three drifts at turns $\{280, 580, 870\}$, each shifting one or two attributes via 20–30 behavioral cues distributed across the surrounding session. Per persona we sample 20 evaluation QA pairs every 50 turns (420 pairs each, 2,100 in total), balanced across four reasoning types and five interference tags. The whole suite is verified by Gemini-2.5-Pro with the same six-criterion protocol as DiMoBench, achieving a 96.4% average `ACCEPT` rate across personas (range 94.8–97.6%) after one revision round.

Trajectory data. Table 17 reports the full N_{raw} / N_{eff} / per-bin counts every 100 turns and at the three drift points on a representative DiMoBench-XL persona. N_{eff} enters its 25.8–28.5 plateau by turn 200, dips by at most 1.1 points at a drift on this persona (at most 1.6 across the five personas), and recovers within $\sim$50 turns to the same band. N_{active} (items with $w_t \geq 0.8$) stabilizes at 17–18 across all three drift epochs, indicating that each drift replaces the previous active set rather than accreting onto it. Across the five personas the per-bin N_{eff} trajectory stays within ± 1.4 of the values reported here, and N_{active} never leaves the 16–19 band after turn 200, confirming the same qualitative behavior without persona-level outliers.

QA accuracy stability. Table 18 aggregates QA accuracy across the five DiMoBench-XL personas in five time windows (means across the five personas). Across 1050 turns and three drifts, Overall accuracy decays only 2.3 points (from 84.3% to 82.0% on average) and the average Top-10 retrieval confidence stays at or above 0.78 in every persona, confirming that long-horizon retention is gracefully degraded by confidence weighting rather than catastrophically collapsed; persona-level standard deviations stay below 1.0 accuracy point in every window.

C.7 Memory dynamics: footprint, contamination, and retention

Headline metrics and retrieval contamination. Table 19 summarizes the headline scalability numbers used in Section 3.5. Below we further examine whether deprecated FACTs leak into Top-k retrieval.

Table 17: **DiMoBench-XL memory trajectory (representative persona).** Every 100 turns (and at the three drift points) on one of the five DiMoBench-XL personas; per-persona deviations from this trajectory stay within ± 1.4 on N_{eff} throughout the run. The four count columns partition $\mathcal{B}_t$ by confidence: N_{active} ($w_t \geq 0.8$), N_{stable} ($0.5 \leq w_t < 0.8$), N_{weak} ($0.1 \leq w_t < 0.5$) and N_{depr} ($w_t < 0.1$); the last column is the ratio $N_{\text{eff}}/N_{\text{raw}}$. Drift turns ($t \in \{280, 580, 870\}$) are marked with †.

t	N_{raw}	N_{eff}	N_{active}	N_{stable}	N_{weak}	N_{depr}	$N_{\text{eff}}/N_{\text{raw}}$
100	26	21.0	13	8	3	2	0.81
200	44	25.8	16	11	7	7	0.59
280†	57	26.5	17	11	9	20	0.46
300	60	26.7	17	12	9	22	0.45
400	73	27.4	17	12	8	36	0.38
500	81	28.5	18	13	8	42	0.35
580†	89	27.4	17	11	10	51	0.31
600	92	27.6	17	12	9	54	0.30
700	103	28.0	18	12	9	64	0.27
800	113	27.9	18	12	8	75	0.25
870†	121	27.6	17	11	9	84	0.23
900	123	28.2	18	12	8	85	0.23
1000	128	28.0	18	12	8	90	0.22
1050	131	28.2	18	12	8	93	0.22

Table 18: **QA accuracy and retrieval quality on DiMoBench-XL across time windows.** Numbers are averaged across the five DiMoBench-XL personas; per-window standard deviations across personas are ≤ 1.0 accuracy point and ≤ 0.02 for the Top-10 weight, and are omitted for readability. Average Top-10 retrieval confidence stays at or above 0.78 throughout 1050 turns; Overall accuracy decays by only 2.3 points.

Turn window	#QA	Acc. (%)	Avg Top-10 w	Contam.@10 (%)
50–250	80	84.3	0.82	0.4
250–500	100	83.6	0.80	0.6
500–750	100	83.1	0.79	0.9
750–1000	100	82.4	0.78	1.2
1000–1050	40	82.0	0.78	1.3

A second concern is whether deprecated FACTs ($w_t < 0.1$) leak into Top-k retrieval and silently bias the response. We post-analyze *every* evaluation query on DiMoBench and DiMoBench-XL with three different retrieval budgets $k \in \{5, 10, 25\}$, recording (i) the score-mass ratio $\sum_{m \in \mathcal{C}_t,\, w_t < 0.1} s(u_t, m) / \sum_{m \in \mathcal{C}_t} s(u_t, m)$, (ii) the fraction of Top-k slots occupied by deprecated FACTs, and (iii) the average w_t of retrieved items. Results in Table 20 confirm that even at the most aggressive setting (DiMoBench-XL, $k = 25$), deprecated FACTs occupy only 2.6% of slots and contribute 0.94% of total score mass; at our default $k = 10$ the numbers drop to 1.0% and 0.58%. The avg retrieved w remains in $[0.71,\ 0.84]$ across all settings, showing the score is dominated by high-confidence beliefs.

Per-persona end states. Table 21 lists, for each DiMoBench persona, the end-of-dialogue statistics behind the averages in Table 19: accuracy varies by at most 1.6 points across personas, and $N_{\text{eff}}^{\text{end}}$ and the Top-10 contamination stay within 0.3 and 0.2 points of their means. On DiMoBench-XL the per-persona N_{eff} trajectories stay within ± 1.4 of the representative persona of Table 17, $N_{\text{eff}}^{\text{end}}$ within 0.4 and Contam.@10 within 0.25 of the five-persona means.

FACT pruning ablation. The default PreCoMem (config **A0**) prunes `HYPOTHESIS` items at $w_t < 0.1$ but never deletes `FACT`s; this is intentional because deprecated `FACT`s serve as immutable provenance hooks for the corresponding episodic-cache traces. To validate that this provenance choice is not a hidden scalability liability, we test two pruning variants: **A1** also prunes FACTs at $w_t < 0.05$, and **A2**

Table 19: **Memory scalability headline metrics.** N_{raw} grows linearly with the conversation, but N_{eff} converges to a stable plateau and the contamination of deprecated FACTs in Top-10 retrieval stays at or below 1.3% in every window even after 1050 turns and three drifts. Both rows average over their five personas; per-persona deviations are ≤ 0.5 on $N_{\text{eff}}^{\text{end}}$ and ≤ 0.4 on Contam.@10.

Setting	$N_{\text{raw}}^{\text{end}}$	$N_{\text{eff}}^{\text{end}}$	Contam.@10 (%)	QA Δ (%)
DiMoBench (350 t.)	62	27.6	0.6	-0.6
DiMoBench-XL (1050 t.)	131	28.2	1.0	-2.3

Table 20: **Retrieval contamination by deprecated FACTs ($w_t < 0.1$).** Even at $k = 25$, deprecated items occupy at most 2.6% of Top-k slots and contribute under 1% of the score mass.

Dataset	k	Score-mass (%)	Top-k depr. (%)	Avg retrieved w
DiMoBench	5	0.21	0.0	0.84
DiMoBench	10	0.34	0.6	0.81
DiMoBench	25	0.62	1.7	0.74
DiMoBench-XL	5	0.36	0.2	0.81
DiMoBench-XL	10	0.58	1.0	0.78
DiMoBench-XL	25	0.94	2.6	0.71

prunes FACTs at the same $w_t < 0.10$ used for hypotheses. Table 22 reports performance and final memory size on LoCoMo (*gpt-4o-mini*, 10-sample mean) and on DiMoBench (350-turn end-state).

A1 cuts N_{raw} from 287 to 168 on LoCoMo while losing only 0.06 Overall points (well below the seed standard deviation of 0.28), confirming that the additional FACTs A0 retains were already behaviorally inactive. A2 starts pruning items that still carry residual signal and consequently loses 0.5 Overall on LoCoMo and 1.1 points on DiMoBench. We therefore default to A0 for provenance and recommend A1 only for storage-bounded deployments.

Cross-system memory footprint. A more detailed cross-system footprint comparison, including the per-item-confidence design dimension, is reported in the main paper as Table 2 (Section 3.5). In short, on the ten LoCoMo dialogues ($\bar{N} = 588$ turns), PreCoMem-A1 stores 168 items, lower than both O-Mem (232) and EverMemOS (264), while delivering $+2.6$ and $+3.4$ Overall points respectively; O-Mem and EverMemOS commit each interaction through a discrete decision, so their stores admit no per-item confidence and the effective count N_{eff} is undefined.

C.8 Retrieval dependence and reliability calibration

The write path acts only on the retrieved set $\mathcal{C}_t$: if the belief that should be revised is not retrieved, the contradiction cannot be detected and `CORRECT` cannot fire in that turn. This section quantifies that dependence and then examines what the reliability coefficient ρ_t actually tracks.

Belief-update recall and failure decomposition. We call an *update-due event* a point at which a stored belief must be revised for a later question to be answered correctly (one per evaluation question on DiMoBench and DiMoBench-XL, one per state transition with a pre-existing belief on AMemGym, and one per knowledge-update turn on LongMemEval-S KU), and we measure the Recall@10 of the belief due for update, i.e., how often it enters the write-path Top-10 at that event. Table 23 shows that recall covers most events (89–95%), leaving an explicit 5–11% blind spot, and that misses matter: question accuracy on events whose target was retrieved exceeds that on missed events by 16–22 points. Retrieval is favorable here because a contradictory utterance shares the attribute of the belief it contradicts with opposite polarity, so the target scores high on the semantic and lexical terms of Equation (1). Decomposing all update-due events pooled over the four benchmarks, 89.3% succeed; the residual 10.7% splits into unrecovered retrieval misses (5.4%; a further 1.4% of events are missed at retrieval but corrected later), gate misses (3.1%; the target was retrieved but the turn was not routed to `CORRECT`) and localization misses (2.2%; the turn was routed to `CORRECT` but the target was not decayed). Retrieval is thus the largest single source, with half of the failures (5.3 of the 10.7 points) sitting in gating and localization; on LongMemEval-S KU,

Table 21: **DiMoBench end-of-dialogue statistics per persona** (350 turns; mean over three seeds). The means reproduce the DiMoBench row of Table 19.

Persona	Acc. (%)	$N_{\text{raw}}^{\text{end}}$	$N_{\text{eff}}^{\text{end}}$	Contam.@10 (%)
P1 (canonical)	84.1	62	27.6	0.60
P2	83.2	65	27.8	0.80
P3	84.8	59	27.3	0.40
P4	84.4	61	27.5	0.55
P5	83.6	63	27.8	0.65
Mean	84.0	62	27.6	0.60

Table 22: **FACT pruning ablation.** A1 ($w < 0.05$) shrinks raw memory by 41% on LoCoMo and 34% on DiMoBench while losing ≤ 0.2 accuracy points, inside seed variance. A2 ($w < 0.10$) starts pruning items that still have residual influence and degrades accuracy by 0.5–1.1 points.

	LoCoMo (gpt-4o-mini)						**DiMoBench (350 t. end)**		
Config	Single	Multi	Temp.	Open	Overall	$N_{\text{raw}}^{\text{end}}$	Acc. (%)	$N_{\text{raw}}^{\text{end}}$	$N_{\text{eff}}^{\text{end}}$
A0 (default; no FACT prune)	81.41	68.49	66.21	39.63	**73.57**	287	84.0	62	27.6
A1 (FACT prune $w < 0.05$)	81.36	68.42	66.18	39.55	73.51	168	83.8	41	27.5
A2 (FACT prune $w < 0.10$)	80.95	68.05	65.74	39.21	73.06	142	82.9	33	26.7

where each update is stated once, retrieval misses dominate. A miss does not stall adaptation: the observation still enters the episodic cache and can form a hypothesis, and 20% of missed events (147 of 718) are corrected later by subsequent cues or hypothesis promotion.

To separate association from causation we force the target belief into the Top-10 at every missed event (displacing the lowest-ranked item) and re-run the write path and the question. This closes only 26–32% of the hit–miss gap (last column of Table 23): forcing the hit raises accuracy on the affected questions by 4.5–6.5 points, while the rest of the gap reflects that missed events are intrinsically harder and that gating and localization can still fail once the target is retrieved.

What ρ_t measures: calibration. $\rho_t = 1 - H_t$ is the normalized concentration of the Top-K retrieval scores. It does not judge whether an individual retrieved item is correct; it weights how far the surprisal computed from that recall can be trusted, and because it enters only multiplicatively it can make the gate more conservative but never more aggressive. Two cases illustrate the behavior. If several relevant memories are correctly retrieved with similar scores, ρ_t is lower and the update is dampened and deferred; no information is lost, since supporting evidence keeps accumulating through the hypothesis lifecycle and a genuine change is still committed after slightly more confirmation. If a wrong memory receives a very high score, ρ_t is high, but ρ_t is deliberately not the sole trigger: whether `CORRECT` fires still depends on d_{conf}, an NLI verifier run against the actual new utterance, and the change is a bounded, reversible soft decay that keeps provenance.

To check that concentration is a usable proxy for evidence precision, we stratify the 2,829 gate events of the AMemGym runs (943 per seed) into quintiles of ρ_t and measure, per quintile, the precision of the retrieved set against the user's current gold state, the recall of the belief due for update, the accuracy of the contradiction decision against 400 blind-annotated events (80 per quintile), the correctness of the resulting update, the share of `CORRECT` firings judged unwarranted (false-`CORRECT`), and the accuracy of the next state-query probe on the touched attribute (Table 24). Precision, contradiction accuracy, update correctness and downstream accuracy all rise monotonically from Q1 to Q5, and false corrections fall from 2.1% to 0.6% (only 5 of the 612 `CORRECT` firings were judged false overall), whereas recall is non-monotone and peaks in the middle quintiles. ρ_t therefore tracks evidence precision, as a precision weight on the prediction error should.

Correction delay during drift. A related concern is that genuine drift produces mixed retrieval evidence, old facts and new contradictions both in the Top-K, which raises H_t and may delay `CORRECT` precisely when it is needed. We measure this on the 600 gold transitions of the AMemGym runs (20 profiles $\times$ 10 transitions $\times$ 3 seeds). In the first period after a change, a fresh belief node

Table 23: **Belief-update recall and the cost of a miss.** Recall@10 of the belief due for update (mean $\pm$ std over three seeds); question accuracy (%) on events whose target was retrieved (hit) or not (miss), and on missed events after forcing the target into the Top-10 (forced hit); the last column is the share of the hit–miss gap recovered by the forced hit. AMemGym accuracies are changed-state probe accuracies.

Benchmark	**Recall@10**	**QA (hit)**	**QA (miss)**	**QA (forced hit)**	**Gap recovered**
DiMoBench	95.0 ± 0.4	84.8	64.2	70.7	32%
DiMoBench-XL	93.0 ± 0.2	84.7	63.0	69.4	29%
LongMemEval-S KU	90.0 ± 0.5	93.4	75.8	80.3	26%
AMemGym	89.1 ± 0.5	72.1	55.8	60.6	29%

Table 24: **Calibration of ρ_t on AMemGym** (2,829 gate events over three seeds, split into ρ_t quintiles from Q1, lowest, to Q5, highest; means over seeds). Precision@K and Recall@K use different references (the user's current gold state versus the single belief due for update); false-`CORRECT` is the share of `CORRECT` firings in the quintile judged unwarranted in the blind audit.

	Q1	**Q2**	**Q3**	**Q4**	**Q5**
Gold-evidence Precision@K $\uparrow$	0.58	0.68	0.76	0.82	0.87
Gold-evidence Recall@K	0.86	0.88	0.89	0.86	0.82
Contradiction accuracy $\uparrow$	0.71	0.79	0.84	0.89	0.92
Update correctness $\uparrow$	0.74	0.80	0.85	0.90	0.93
False-`CORRECT` (%) $\downarrow$	2.1	1.0	0.8	0.7	0.6
Downstream probe accuracy $\uparrow$	0.66	0.71	0.75	0.78	0.81

encoding the new state is created for 91.3% of the transitions and enters the answering Top-10 for 88.7%; the remaining 2.6% is a read-path ranking loss, since the new belief exists but does not enter the answering Top-10. A low ρ_t delays a warranted `CORRECT` in 14.2% of the transitions; of these delays 78.0% clear within one period and 94.0% within two, and only 2.8% persist to the end of the run. End-to-end, this bounded delay is what separates PreCoMem's one-period recovery from the two periods of AWE and the 1.8–3.2 periods of the ablated variants (Section C.1); removing the weighting altogether instead costs 16.76 points on LoCoMo (RR in Figure 4(a)).

The same analysis on DiMoBench, which has gold change points, uses the ρ-assessability criterion that a gold shift counts only when the outdated belief is in the Top-K and $S_{\text{raw}} > \theta_{\text{high}}$, so that $\rho_t = 1$ would route the turn to `CORRECT`; the delay d is measured in turns (Table 25). Delays are the exception and bounded on both tracks: 7.3% of the assessable shifts on DiMoBench and 14.8% on DiMoBench-XL are delayed, and 71.4% of the delays clear within one turn and 85.7% within two. DiMoBench-XL exceeds the base track, consistent with its three drifts over 1050 turns and denser decoys. Since DiMoBench counts turns whereas AMemGym counts periods, the two rates agree in magnitude without sharing a unit, which is why we report them per track.

Cold start and genuinely new topics. A genuinely new topic differs from every stored belief, so its semantic-deviation term d_{sem} is high while $d_{\text{conf}} \approx 0$, and S_{raw} is driven up by novelty; even when all absolute similarities are small, one loosely related item normally dominates the Top-K, so ρ_t stays moderate and S_{eff} lands in the `PROFILE` band. `MAINTAIN` would require an unsurprising cue with low d_{sem}, which a new topic is not. In cold-start probes on the AMemGym profiles, a first-time cue is written on an empty store 96.5% of the time (579 of 600 probes, each starting from a reset store); when the store is populated and the cue matches little, a usable belief forms within one period in 84.2% of cases and within two in 92.7% (600 new-topic probes), while only 5.6% of 180 distractor cues are written as hypotheses. A low match therefore costs at most a short confirmation delay. Section C.11 walks through an example.

C.9 Retrieval and end-to-end latency analysis

Experimental setup. We conduct all latency evaluations on the LoCoMo dataset. To ensure a rigorous comparison, we standardize the hardware and model environment across all systems:

Table 25: `CORRECT` **delay at gold change points on DiMoBench and DiMoBench-XL** (counts pooled over three seeds). A shift is ρ-assessable when the outdated belief is in the Top-K and $S_{\text{raw}} > \theta_{\text{high}}$; d is the delay in turns.

Track	Gold shifts	ρ-assessable	d=0	d=1	d=2	d>2	Any delay
DiMoBench	45	41	38	3	0	0	$3/41 = 7.3\%$
DiMoBench-XL	135	122	104	12	3	3	$18/122 = 14.8\%$

- **LLM Backbone:** *gpt-4o-mini* serves as the backbone for all generation and retrieval operations.
- **Embeddings:** All embeddings are generated using `Qwen3-Embedding-0.6B` with 1024 dimensions.
- **Configuration:** Each baseline operates under its official recommended hyperparameters.

Metrics. We report P50 and P95 latency for *both* the retrieval stage and the full end-to-end (E2E) response:

- **Retrieval latency** measures only the cost of producing the retrieved evidence set $\mathcal{C}_t$ (memory read, scoring, and any re-ranking), and isolates the architectural cost of the memory system itself.
- **End-to-end (E2E) latency** measures the full user-visible time from receiving a query u_t to emitting the final answer, including retrieval plus generation.
- **P50 (Median):** The median latency, representing the typical time required to process a query.
- **P95 (95th Percentile):** The threshold within which 95% of queries are completed, reflecting performance stability under high load or ambiguity.

Results and analysis. Table 26 summarizes the comparative performance. PreCoMem achieves a retrieval P50 / P95 of 1.95s / 3.86s and an E2E P50 / P95 of 2.35s / 4.36s.

- **Efficiency via vectorized scoring:** PreCoMem achieves a competitive P50 by leveraging a confidence-modulated hybrid scoring mechanism (Equation (1)). Rather than relying on computationally intensive generative re-ranking or iterative graph traversals, our approach efficiently aggregates semantic and lexical signals through parallelizable matrix operations. This design ensures that the computational overhead remains modest relative to pure vector retrieval, even as the complexity of the belief store increases.
- **Stability via deterministic flow:** A critical advantage of PreCoMem is its robustness against latency spikes. Unlike agentic architectures (e.g., A-Mem) that may trigger unpredictable, recursive reasoning loops to resolve ambiguities, PreCoMem employs a deterministic, non-recursive retrieval flow. This architectural choice constrains both the retrieval tail (P95: 3.86s) and the E2E tail (P95: 4.36s), ensuring consistent performance without the extreme delays (E2E P95 up to 7.68s for A-Mem) often observed in stochastic agentic systems.
- **Retrieval vs. E2E gap:** The gap between retrieval and E2E latency is dominated by generation, which is largely independent of the memory backend. PreCoMem's E2E overhead over the retrieval stage (0.4s at P50, 0.5s at P95) is comparable to other non-agentic baselines, confirming that adding `PROFILE`/`CORRECT` does not materially inflate generation cost.

C.10 Structural comparison: why PreCoMem outperforms existing memory systems

Beyond the headline accuracy gains in Table 1, two structural properties of PreCoMem account for the consistent improvement and have no direct analogue in any of the nine baselines we compare against. We summarize the contrast along the two axes that most directly govern long-horizon behavior.

Table 26: **Retrieval and end-to-end latency comparison.** Evaluated on LoCoMo with *gpt-4o-mini*. "Retrieval" is the memory read/scoring cost only; "E2E" is the full query-to-answer time (retrieval + generation). PreCoMem obtains competitive typical latency and a well-controlled tail at both stages, without the extreme P95 spikes seen for recursive agentic baselines.

Method	Retrieval		End-to-End	
	P50	P95	P50	P95
MemoryOS	1.66s	2.21s	–	–
MemOS	1.69s	3.44s	2.04s	3.89s
A-Mem	1.74s	7.23s	2.09s	7.68s
Mem0	2.44s	4.29s	2.74s	4.69s
PreCoMem (Ours)	1.95s	3.86s	2.35s	4.36s

Axis 1: Robustness to noise and faithful preservation of user core traits. Existing systems represent the importance of an incoming observation *categorically*. GAM [25] writes one compressed memo per session into an immutable page-store with no per-item weight; EverMemOS [19] validates each MemCell within a hard time interval $[t_{\text{start}}, t_{\text{end}}]$ and treats it as expired once the interval closes; TiMem [22] encodes importance through its five-level temporal hierarchy and explicitly notes the absence of a storage-time forgetting mechanism; MemBox [23] stores all events inside a sealed topic box at uniform weight; O-Mem [21] updates persona slots via an LLM that selects one of three categorical actions (ADD/IGNORE/UPDATE); Mem0 [26] commits ADD/UPDATE/DELETE/NOOP and physically removes contradicted memories on DELETE; and A-Mem [24] replaces an existing note with an LLM-rewritten note via Memory Evolution. None of these systems can defer commitment when evidence is ambiguous: a weak signal is either ignored or instantly written as a fact. PreCoMem's continuous confidence weight $w_t(m) \in [0, 1]$ paired with the `HYPOTHESIS` waiting room lets weak cues accumulate evidence across turns and only graduate into `FACT` once support concentrates. The same mechanism preserves established traits: soft-decay correction (Equation (10)) reduces the influence of contradicted facts without erasing them, so a transient noise burst cannot remove a long-established trait the way a hard DELETE or overwrite would. This explains the LongMemEval-S gains on SSP (single-session preference), MS (multi-session aggregation), and KU (knowledge update) in Table 1, where ambiguous and conflicting cues dominate the question pool and categorical commit decisions are exactly the wrong granularity.

Axis 2: Retrieval coherence and avoidance of fragmentation. Each baseline retrieval substrate carries a structural failure mode that PreCoMem sidesteps. GAM iterates a deep-research loop over its page-store, whose answer correctness is order- and budget-sensitive; EverMemOS issues dense+BM25 retrieval over Atomic Facts, fuses by RRF, and then runs a sufficiency-check loop that can leave cross-shard inconsistency between MemCells of the same scene; TiMem performs a two-stage hierarchical recall whose scope depends on a planner classifier; MemBox returns full topic boxes wholesale with no within-box reweighting; O-Mem aggregates three parallel channels (working / episodic / persona) by union, so irrelevant attribute hits dilute the context window; Mem0 falls back on a single dense-similarity threshold and discards deleted-but-relevant facts; and A-Mem performs pure cosine top-k over linked atomic notes, where the fine granularity amplifies fragmentation noise. All of these substrates therefore require an explicit reranker, sufficiency loop, or planner to suppress incoherent reads. PreCoMem instead multiplies a tri-signal relevance score by the per-item confidence $w_t(m)$ (Equation (1)), which acts as an implicit reranker at no extra LLM call: high-support items naturally rise to the Top-K, deprecated items naturally sink, and the entropy-based reliability ρ_t (Equation (3)) further damps unreliable updates when the score mass is diffuse. This is what produces the bounded retrieval contamination in Table 20 and the bounded effective memory in Figure 5(b,c), both of which scale to the 1050-turn DiMoBench-XL run without recourse to a maintenance loop.

Coupling the two axes. The two axes are complementary: Axis 1 controls *what is committed* so that the belief store stays faithful to the user's core traits under noise, while Axis 2 controls *what surfaces at read time* so that the dialogue agent sees a coherent, currently-supported view. Existing baselines optimize one axis at a time (e.g., EverMemOS's profile updates target Axis 1; GAM's deep-research targets Axis 2) but lack a single mechanism that couples them. PreCoMem couples them through the

Example Belief Store snapshot (turn 210, DiMoBench)

```
[FACT]  Mike enjoys group social activities     (w=0.82)
        sources=[D1:3, D1:5, D3:7]
[FACT]  Mike and Leo often organize KTV outings  (w=0.71)
        sources=[D2:4, D4:2]
[HYPO]  Mike may be feeling socially fatigued    (w=0.38)
        sources=[D21:6, D21:8]
```

Figure 10: **Compact view of a Belief Store snapshot.** Each entry shows the type, natural-language content, confidence weight, and pointers to the episodic cache. `FACT` items are committed, `HYPOTHESIS` items are tentative; confidences modulate retrieval priority but are never shown to the dialogue agent directly.

same scalar confidence weight w_t, which is the structural reason behind the consistent improvements observed across the four main benchmarks.

C.11 System inspectability, traces, and qualitative analyses

Memory store serialization and example snapshot. To make the contents and interface of the Belief Store concrete, we describe how a belief item is serialized in practice and show an example snapshot from a DiMoBench run at turn 210, i.e., right after the change-point session. Each belief item $m \in \mathcal{B}_t$ carries (i) a natural-language proposition $c(m)$, (ii) a type label $\tau(m) \in \{\texttt{FACT}, \texttt{HYPOTHESIS}\}$, (iii) a confidence weight $w_t(m) \in [0, 1]$, (iv) a creation timestamp and last-updated turn, and (v) pointers into the episodic cache $\mathcal{E}_t$ for provenance. A minimal serialization is shown in Figure 10.

What the dialogue agent actually sees. Confidence weights $w_t(m)$ and other metadata are *not* exposed to the response-generation LLM. They only modulate retrieval ranking through the confidence-weighted score in Equation (1), so that high-confidence items surface near the top of $\mathcal{C}_t$. The dialogue agent receives a plain-text prompt constructed from the Top-K item contents $\{c(m)\}_{m \in \mathcal{C}_t}$ together with the relevant episodic-cache excerpts. This keeps generation-time inputs close to a standard retrieval-augmented prompt while still letting the memory system use confidences internally.

Inspectability. Because every belief carries explicit type, confidence, and source pointers, snapshots like Figure 10 can be dumped at any turn for inspection, regression testing, or qualitative error analysis, without re-executing the model. This also makes it straightforward to audit which episodic-cache turns support each currently-held belief, which we use for the end-to-end traces below.

End-to-end trace examples on DiMoBench. Below we provide two end-to-end traces that expose PreCoMem's internal read–write decisions on a DiMoBench example dialogue. For readability, we list the retrieved belief set $\mathcal{C}_t$ by the episodic source turns of its items, without retrieval weights/scores, and we report only the qualitative regime of effective surprisal ($x_t = S_{\text{eff}}$) relative to $(\theta_{\text{low}}, \theta_{\text{high}})$.

Trace format. Each trace includes (i) the newly observed dialogue excerpt u_t (one or a few turns) that triggers a memory write decision, (ii) a compact view of the retrieved evidence set $\mathcal{C}_t$, (iii) the selected evolution operator (`MAINTAIN`/`PROFILE`/`CORRECT`), and (iv) a subsequent QA query answered using the updated memory state.

Trace 1 (change point): early drift signal triggers `PROFILE`. Pre-change belief snapshot (high confidence). Mike enjoys crowded gatherings and rarely skips weekend social events (e.g., `D1:5` "The more the merrier, I love that vibe ..."; `D1:7` "I wouldn't miss weekend gatherings ...").

(Write) New observation u_t (Session 21).

> `D21:6` "Someone invited me out, but I said work's been busy and didn't go."
> `D21:8` "This weekend I went to a cafe alone, found a book and read for the whole

afternoon."
D21:9 "The environment was pretty quiet, not bad."

Recall: Top-K evidence $\mathcal{C}_t$.

D1:5 "The more the merrier, I love that vibe ..."
D1:7 "...I wouldn't miss weekend gatherings ..."
D2:9 "...I really can't understand the joy of being a homebody."
D3:7 "If I missed a gathering like this, I'd definitely have FOMO, afraid of missing out on the fun."
D3:10 "Rest can wait, can't miss the weekend activities! ..."
D9:9 "If one day I don't want to socialize, the sun must be rising from the west ..."
D21:1 "Last weekend's gathering, the atmosphere got a bit awkward."

Reliability-adjusted surprisal. The retrieved set consists of strong *pre-change* evidence (social enthusiasm) and a single emerging *change-point* cue (D21:1), none of which closely matches the new observation (skipped invitation; solo activity). The score mass is therefore spread over several items (higher entropy), making retrieval reliability ρ_t moderate. Contradiction signals are also non-decisive: "work's been busy" can explain a single skip, so d_{conf} is non-zero but not saturated. Overall, x_t falls into the medium regime ($\theta_{\text{low}} < x_t \leq \theta_{\text{high}}$).

Evolution: PROFILE**.** PreCoMem writes a low-confidence hypothesis capturing a *possible* shift, e.g., "Mike may be starting to reduce large gatherings and explore quiet solo activities," and links it to D21:6/D21:8-9 in the episodic cache. Crucially, PreCoMem does *not* immediately correct the long-standing high-confidence "social enthusiast" belief, avoiding premature overwrite from a single ambiguous event.

(Read) QA query. *Q:* "Is Mike still as enthusiastic about gatherings as before?"
A: "Seems to be declining ..."
Using the newly written hypothesis plus the concrete episode (D21:6 skipped invitation; D21:8-9 solo cafe), PreCoMem answers with a *trend-aware* response (declining / becoming evasive) while remaining cautious about permanence. In later sessions, repeated consistent cues (e.g., D23:3 "declined most invitations") reinforce and eventually promote this hypothesis.

Trace 2 (post-change, temporal decoy): reliable contradiction triggers CORRECT**. Precondition.** After multiple post-change sessions, PreCoMem has accumulated consistent evidence favoring solitude (e.g., D28:2 "Without gatherings I don't feel bored anymore ..."). However, an older pre-change statement remains in memory as a potential temporal decoy:

D1:8 "If there's no gathering on the weekend, I feel like I'd be bored to death ..."

(Write) New observation u_t (Session 33).

D33:9 "Used to say I'd be bored without socializing, now being alone is actually more fulfilling."

Recall: Top-K evidence $\mathcal{C}_t$.

D1:8 "...bored to death ..."
D22:6 "This weekend I'm planning to stay home, maybe watch some shows or something."
D23:2 "Used to think quiet places like that were boring, now I find them comfortable."
D23:3 "...declined most gathering invitations ..."
D25:2 "Used to think being alone was boring, now I actually enjoy it."
D25:4 "...being alone is recharging, socializing ...drains energy."
D28:1 "Used to look forward to gatherings on weekends, now I look forward to alone time."
D28:2 "Without gatherings I don't feel bored anymore ..."
D33:2 "Reducing socializing ...not so exhausted anymore."

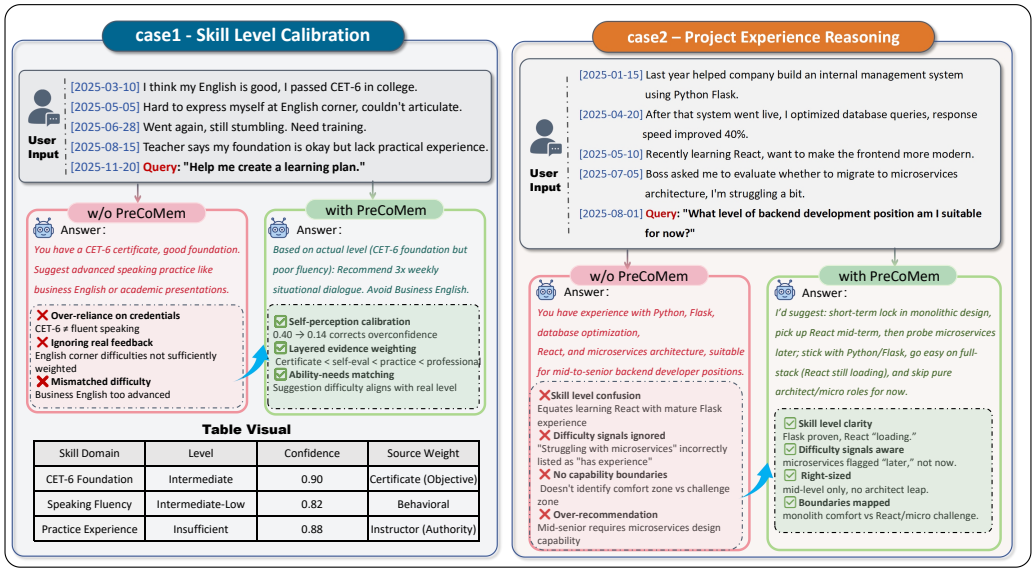

Figure 11: **Qualitative analysis of evidence consolidation. Left:** In skill calibration, PreCoMem weighs credentials against contradictory behavioral cues to avoid over-estimation. **Right:** In project reasoning, it distinguishes high-confidence competencies from tentative learning ("loading"), preventing premature recommendations.

Reliability-adjusted surprisal. Here the recall distribution is more concentrated because multiple items explicitly mention the disputed claim ("bored without socializing") and its reversal, yielding higher ρ_t. The contradiction signal is strong: `D33:9` explicitly negates the pre-change stance in `D1:8`, so d_{conf} is high. Consequently, x_t falls into the high regime ($x_t > \theta_{\text{high}}$).

Evolution: `CORRECT`**.** PreCoMem applies soft decay (Equation (10)) to the contradicted belief(s) retrieved from $\mathcal{B}_t$ that encode the old stance (e.g., "weekends without gatherings are boring"), while leaving non-contradictory items largely unchanged. Because confidence weights directly modulate future retrieval (Equation (1)), this correction reduces the chance of "old-memory pull" when later queries contain temporal-decoy phrasing.

(Read) QA query. *Q:* "Would Mike still say 'I'd be bored without socializing'?"
A: "No, being alone is actually more fulfilling now."
PreCoMem retrieves the recent, high-support solitude evidence (`D33:9`, `D28:2`) and answers accordingly, while older evidence like `D1:8` is downweighted by correction and recency, making the response robust to the temporal decoy.

Qualitative case studies beyond accuracy. Figure 11 shows two cases in which the answer depends on how evidence was consolidated; a third, constructed case walks through a new topic entering the store.

Case 1: Skill level calibration. Figure 11 (left) illustrates this case. The user initially reports a credential (CET-6), which can tempt a retrieval-only system to over-trust a single strong signal and recommend overly advanced practice. PreCoMem instead consolidates evidence across sessions: repeated behavioral cues (e.g., failing to articulate at an English corner) and professional feedback (teacher assessment) lower confidence in an over-optimistic self-evaluation, while preserving the belief that the foundation is adequate. At query time, **Memory Leveraging** recalls these calibrated beliefs (rather than a single credential trace) to condition response generation, leading to a plan that targets fluency and practice rather than premature advanced content.

Case 2: Project experience reasoning. Long-horizon career discussions often mix stable skills (proven experience) with tentative claims (currently learning) and difficulty signals (struggles with complex systems). PreCoMem uses its confidence-weighted belief store to separate what is established

(e.g., Python/Flask) from what is still emerging (e.g., React "loading") and what should be deferred (e.g., microservices architecture). This produces right-sized recommendations: it proposes a near-term path aligned with demonstrated capability, while explicitly marking a challenge zone for later progression.

Case 3: A new topic entering a populated store. Suppose the store holds the user's work and food profile but nothing about music, and the user says for the first time: "I signed up for guitar lessons and practiced for an hour last night." No belief matches and the Top-K items are only loosely related, so absolute scores are low but one still dominates, giving a moderate ρ_t and, by the mechanism described in Section C.8, `PROFILE`. The `Profile-Agent` writes the hypothesis "the user may have taken up guitar" with $w_0 < \tau_{\text{promote}}$, linked by provenance to that turn and retrieval-visible at once, so the agent can already answer cautiously. If guitar recurs, `MAINTAIN` reinforces the hypothesis until it crosses τ_{promote} and becomes a `FACT`; otherwise it remains a low-confidence hypothesis that never outranks committed facts. A new topic therefore enters provisionally, is usable immediately, and is committed only after repeated support.

Connection to the consolidation loop. Across these cases, PreCoMem's leveraging–evolution loop links response quality to memory consolidation. Evidence recalled for the current answer is also used to update confidence through `MAINTAIN`/`PROFILE`/`CORRECT` (Section 2.4), so future retrieval and recommendations reflect accumulated support, implicit verification, and soft correction under contradiction.

D Prompts

All prompts used in this paper are released verbatim with the code at `https://github.com/ZhongWishing/PreCoMem`. The three sub-agent prompts (`Profile-Agent`, `Correct-Agent`, `Maintain-Agent`) and the response-generation prompt are in `precomem/llm/prompts.py`; the QA and LLM-as-a-judge prompts for LoCoMo, LongMemEval-S, PersonaMem-v2, and DiMoBench are in `precomem/evaluation/prompts.py`; and the DiMoBench generation and verification prompts are in `precomem/dimobench/prompts.py`.

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 the Introduction (Section 1) state three contributions—(i) the PreCoMem framework casting memory as a dynamic prediction of user state, (ii) a retrieval-reliability–weighted surprisal that gates a three-operator update (`MAINTAIN/PROFILE/CORRECT`), and (iii) the evaluation on LoCoMo, LongMemEval-S, PersonaMem-v2 and the new DiMoBench drift stress-test—each of which is supported by Sections 2, 3.1, 3.3 and the corresponding appendices.

   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: Limitations are discussed alongside the relevant components: fixed-threshold sensitivity and the trade-off against adaptive thresholds (Appendix C.5); dependence on a fixed NLI verifier for d_{conf} (Appendix C.4); the choice to retain rather than prune low-confidence FACTs and its scalability implications (Appendix C.7); the diminishing-returns regime of the retrieval budget k (Appendix C.5); and the English-only, dialogue-only evaluation scope inherited from the underlying benchmarks (Section 4).

   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: [Yes]

   Justification: All theoretical components used in the method are derived in the appendix: the per-turn on-policy loss as a Monte Carlo surrogate (Appendix B.1, with the unbiasedness statement in Equation (14)); the `MAINTAIN`, `CORRECT`, and two-threshold gate updates (Appendix B.3, including the KL-proximal mirror-descent derivation Equations (23) to (25)); and the cost-sensitive interpretation of the gate (Equation (27)). Assumptions (fixed response policy for the MC sample, scalar weights in $[0, 1]$, $\eta \in (0, 1]$, $\beta > 0$) are stated where they are used.

   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: Section 2 fully specifies the architecture and update rules; Equations (1) and (3) to (10) pin down every operator with named hyperparameters $(\lambda, \beta, \eta, \gamma, \tau_{\text{promote}}, w_{\min}, w_{\max}, \theta_{\text{low}}, \theta_{\text{high}})$. The implementation details (backbones gpt-4o-mini / gpt-4.1, embedding model `Qwen3-Embedding-0.6B`, NLI verifier `cross-encoder/nli-deberta-v3-base`, k=10, $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.2, 0.6)$, λ=0.5, β=1.0, $\boldsymbol{\alpha}$=$(0.4, 0.4, 0.2)$, γ=0.05, three random seeds) appear in Section 3.1 and the appendix. All sub-agent prompts and dataset evaluation prompts are released with the code (Appendix D), and the DiMoBench construction pipeline is documented stage-by-stage in Appendix C.6.

   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: We release, at `https://github.com/ZhongWishing/PreCoMem`, the PreCoMem implementation, evaluation scripts, sub-agent prompts, the five 350-turn DiMoBench dialogues with their 1,000 QA pairs, the five 1050-turn DiMoBench-XL dialogues with their 2,100 QA pairs, the construction and verification prompts, and the human-review scripts (see Appendix C.6), together with the frozen configurations and run records of the AMemGym and LoCoMo-Plus experiments (Appendix C.1). Existing benchmarks (LoCoMo, LongMemEval-S, PersonaMem-v2, AMemGym, LoCoMo-Plus) are used at their public versions and accessed via their original URLs.

   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: Section 3.1 (Implementation Details) lists the backbones, embedding model, recall budget, and gating thresholds. Per-benchmark splits and protocols follow the original LoCoMo, LongMemEval-S, and PersonaMem-v2 conventions; the DiMoBench split design is in Appendix C.6. Hyperparameter choices are justified by sensitivity studies in Appendix C.5 (k, θ, and the adaptive variant) and Appendix C.7 (FACT pruning). PreCoMem does not involve gradient-based training; the only learned component is the temperature T on the NLI verifier, fit on the MNLI matched dev split (Appendix B.2).

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: [Yes]

Justification: All accuracy numbers in the main results (Table 1, Table 7) are reported as mean $\pm$ standard deviation over three random seeds; the dynamic-threshold table (Table 15) and the FACT pruning table (Table 22) report point estimates without error bars. Bar plots in Figure 3 show $\pm$std error bars. Reported variability captures both LLM judge stochasticity and seed-induced retrieval order differences.

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: All experiments ran on a single workstation with $4\times$NVIDIA A100 80GB GPUs and an AMD EPYC 64-core CPU (512GB system RAM). Embedding (`Qwen3-Embedding-0.6B`) and NLI (`cross-encoder/nli-deberta-v3-base`) inference required approximately 30 GPU hours per benchmark sweep. LLM-side reasoning (gpt-4o-mini and gpt-4.1) was performed via the OpenAI API ($\sim$20k requests and $\sim$32 wall-clock hours per full benchmark sweep across three seeds). The remaining models were likewise accessed through APIs or OpenAI-compatible endpoints: gpt-4.1-mini (primary judge; AMemGym backbone), GPT-4o (LoCoMo-Plus) and GPT-5-Chat / GPT-5-mini (backbone generalization) via the OpenAI API, gemini-2.5-flash (auxiliary judges) and Gemini-2.5-Pro (DiMoBench verifier) via the Google API, and Qwen3.5-9B/35B/122B (backbone generalization) through an OpenAI-compatible vLLM endpoint; per-query retrieval and end-to-end latencies are tabulated in Appendix C.9. The full research project, including the threshold and Top-k sweeps, the FACT-pruning ablation, and the DiMoBench-XL stress test, required roughly $4\times$ the reported compute due to seed and configuration repeats.

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: We have reviewed the NeurIPS Code of Ethics. The work uses publicly available benchmarks (LoCoMo, LongMemEval-S, PersonaMem-v2, AMemGym, LoCoMo-Plus) under their original licenses and a self-constructed synthetic dialogue benchmark (DiMoBench) that contains no real personal data; quality assurance was performed by project collaborators rather than crowd workers. We disclose all material details required for transparent evaluation.

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: *Positive impacts.* A more stable yet adaptive long-term memory makes conversational agents more consistent and personalized across sessions, which can improve accessibility (e.g., assistive use cases that benefit from recalling user routines and preferences) and reduce the cognitive burden of repeating context. *Risks.* Persistent user memories raise privacy concerns: detailed belief stores could be misused for profiling, targeted persuasion, or surveillance if deployed without user control. The same mechanism that softly corrects outdated beliefs could in principle be exploited by adversarial inputs to overwrite legitimate user history. *Mitigations.* PreCoMem stores beliefs as inspectable, source-attributed items (Appendix C.11) so that users can audit and selectively edit or delete entries; soft

correction is bounded (no hard deletion of facts) so that recovery from accidental drift is possible; and the operator gates are deterministic, which makes write decisions auditable. We recommend that production deployments expose user-facing controls for memory inspection, opt-out, and per-attribute deletion.

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: PreCoMem does not release a new pre-trained model, image/text generator, or scraped dataset. The released artifacts are an algorithm implementation, evaluation scripts, prompt templates, and a synthetic dialogue benchmark (DiMoBench) constructed entirely from LLM generations with no real user data, so the misuse surface is limited.

    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: All third-party assets are cited or identified. The benchmarks are LoCoMo [34], LongMemEval-S [16], PersonaMem-v2 [35], AMemGym [36], and LoCoMo-Plus [37]. The baselines O-Mem, MemoryOS, MemOS, Mem0, A-Mem, GAM, MemBox, TiMem, and EverMemOS are cited in Section 3.1 and Section A. The embedding models are `Qwen3-Embedding-0.6B` and OpenAI `text-embedding-3-small` (AMemGym runs); the NLI verifiers are `cross-encoder/nli-deberta-v3-base`, `facebook/bart-large-mnli` and `roberta-large-mnli` together with the MultiNLI development split used for temperature fitting, all identified by their public identifiers (Apache-2.0 / MIT class licenses). The OpenAI, Google and Qwen backbones are used under their published Terms of Use. Each benchmark is used at the version distributed by its original authors, and we follow their respective evaluation protocols.

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: We introduce two new assets: the PreCoMem framework implementation, and the DiMoBench / DiMoBench-XL benchmarks. The construction pipeline, persona slot, drift schedule, interference taxonomy, six-criterion verification protocol, and human-review procedure for DiMoBench are documented in Appendix C.6, including the DiMoBench-XL extension. The companion prompts for QA, judgment, and sub-agents are released with the code (Appendix D). The repository README (`https://github.com/ZhongWishing/PreCoMem`) documents structure, fields, intended use, and license.

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 crowdsourcing or external paid annotators. Quality assurance for the DiMoBench QA validation and the LLM-judge agreement study (Appendix C.3) was performed by project collaborators with NLP research experience following the calibration protocol described in Appendix C.6; no external workers were recruited.

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: As noted in Q14, the work does not involve human-subject research; annotation was performed in-house by project collaborators on synthetic dialogue and on model outputs. No personal data was collected from study participants and IRB approval is therefore not applicable.

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: LLMs are core components of PreCoMem. We use gpt-4o-mini and gpt-4.1 (via the OpenAI API) for response generation, hypothesis writing in the `PROFILE` agent, conflict diagnosis in the `CORRECT` agent, and reinforcement reasoning in the `MAINTAIN` agent (full prompts released with the code; Appendix D); the backbone-generalization study uses Qwen3.5 and GPT-5 models, and the third-party benchmarks follow their official protocols with gpt-4.1-mini (AMemGym, with gpt-4.1 as the user simulator) and GPT-4o (LoCoMo-Plus), as described in Appendix C.1. LLM-as-a-judge accuracy uses a gpt-4.1-mini primary judge and two gemini-2.5-flash auxiliary judges (validated against human re-annotation in Appendix C.3). For DiMoBench construction, GPT-4o is the generator and Gemini-2.5-Pro the independent verifier (Appendix C.6). LLM use beyond the method itself (e.g., for writing assistance) does not affect the scientific claims.

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.