RETROSPECTIVE AFFECTIVE BELIEF REVISION: WHEN NEW EVIDENCE CHANGES HOW LLM AGENTS INTERPRET PAST AFFECTIVE STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent can recall an interaction accurately and still misunderstand how a person felt. Later disclosures can change the meaning of an earlier silence, hesitation, or affective response, creating a need to revisit past judgments. To address this issue, we introduce Retrospective Affective Belief Revision (RABR), a framework that connects newly available evidence to historical affective interpretations while preserving the original records and the history of belief changes. A compact trigger determines when memory retrieval is needed, a reasoning agent grounds the relevant historical target, and an evidence-based updater distinguishes between revising a judgment and suspending its application. To study when such retrospection is warranted, we construct RABRBENCH, spanning compact controlled situations, extended histories, and source-preserving longitudinal conversations. The benchmark pairs retrieval decisions with evidence-turn annotations and dependency-aware evaluation units, making selective access to the past an explicit learning and evaluation task. Integrated experiments show that selective retrieval reduces downstream model calls, while controlled correction studies provide persistent evidence support and targeted evidence retraction. RABR and RABRBENCH establish a framework and benchmark for studying when to revisit past affective judgments and how to revise them through traceable evidence updates.

1 INTRODUCTION

Recalling what happened doesn't mean understanding how someone felt. Memory architectures allow large language model (LLM) agents to store, retrieve, and reflect on past events (Park et al., 2023), and to update memories and user representations (Zhong et al., 2024). However, reasoning about temporal and causal relationships across different sessions remains challenging (Maharana et al., 2024). Such reasoning is especially important for understanding other people's affective states, where interpretation depends on context (Barrett & Kensinger, 2010) and relevant details may only become available later. When this happens, the past event remains unchanged, but the evidence available to understand how the person felt has changed. This raises a question: how should an LLM agent review its interpretation of past affective states when new evidence becomes available?

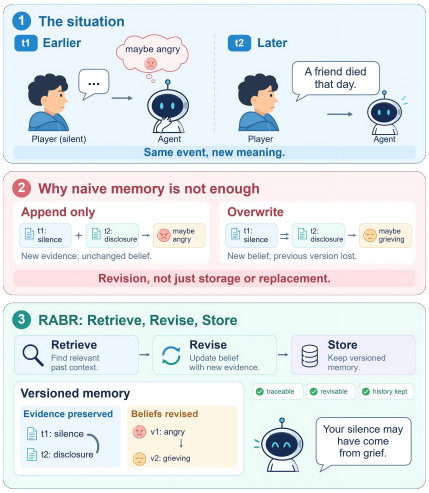

Figure 1: Retrospective reinterpretation.

Imagine a player who falls silent during a conversation, as Figure 1 shows. The agent takes this as a sign of anger and responds accordingly. Later, the player reveals that they had learned of a friend's passing that day, making sadness another plausible explanation for the silence. An agent can remember an exchange but continue reasoning from its original interpretation. Merely storing the disclosure updates the memory record, not the judgment the agent treats as active. Yet, overwriting that judgment would

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

eliminate the interpretation that produced the agent's earlier responses, removing the source needed to explain them.

This problem matters in player modeling (Yannakakis & Togelius, 2025; Melhart et al., 2025), affect-sensitive tutoring (D'Mello & Graesser, 2012; Zhang & Long, 2025), emotional support dialogue (Liu et al., 2021; Kang et al., 2024), and long-term relational agents (Bickmore & Picard, 2005; Zhong et al., 2024). A learner's later explanation of their confusion may prompt a tutor to reinterpret the learner's earlier frustration. Likewise, learning about a difficult life event may help a companion agent reinterpret earlier withdrawal. Linking such disclosures to past behavior can help the agent distinguish a situational response from a stable personal characteristic. Still, not every new observation warrants a memory search, and retrieved evidence may confirm rather than overturn an earlier judgment. The core challenge is to decide when to search past memories and whether the available evidence warrants revising the earlier judgment.

In this work, we introduce Retrospective Affective Belief Revision (RABR), a framework for revisiting earlier affective judgments as new evidence becomes available. RABR preserves the original event records while keeping them separate from revisable beliefs about past affective states. A compact trigger determines when to retrieve historical evidence and what to retrieve. The framework then uses the selected evidence to reassess the relevant beliefs, stores successive versions of each judgment, and links revisions to the evidence supporting them. It also distinguishes revising a judgment from suspending its use when the evidence remains inconclusive. The resulting history shows what the agent believed at the time and how it understands the same event now, allowing downstream systems to adapt while retaining a traceable record of how that understanding changed.

A key prerequisite for RABR is knowing when retrospection is warranted. To study this decision, we introduce RABRBENCH, a benchmark for learning and evaluating when to use historical evidence to inform retrospective affective belief revision. RABRBENCH includes three tracks that cover compact controlled situations, extended histories, and longitudinal conversations that preserve the original dialogue. It pairs retrieval decisions with annotations that identify supporting turns and account for dependencies between related examples during evaluation.

Together, RABR and RABRBENCH advance a retrospective perspective on affective computing, in which an agent's understanding of a person's past affective states remains open to revision as new evidence emerges. Our main contributions are: **1) Selective and traceable affective belief revision.** We introduce RABR, a framework that separates preserved event records from revisable interpretations of past affective states. By linking selective retrieval to evidence-attributed belief updates, it allows agents to reconsider earlier judgments without erasing the interpretations that guided their previous responses. **2) A benchmark for when and how to revisit the past.** We construct RABRBENCH to evaluate retrieval demand and final-affect revision across controlled situations and long dialogue histories. Evidence-turn annotations support retrieval-decision scoring, while a separate target-ranking diagnostic characterizes historical localization difficulty. **3) Empirical evidence for improved retrospective reasoning.** RABR achieves the highest revision agreement across the evaluated packages and improves final-affect prediction on the long-context package, with an 11.8% relative gain in retrieval-weighted final-affect correctness over the strongest baseline. A separate gating comparison shows that selective memory access reduces downstream computation relative to always requesting memory, supporting targeted retrospection.

2 RELATED WORK

Recent work on LLM agent memory explores how stored experience evolves as information flows. A-MEM (Xu et al., 2025) organizes memories as linked notes and updates existing contextual representations when new memories arrive. Memory-R1 (Yan et al., 2026) learns memory addition, updating, deleting, and retention through reinforcement learning, together with agent that selects and reasons over stored entries. LongMemEval (Wu et al., 2024) evaluates information extraction, multi-session and temporal reasoning, knowledge updates, and abstention across extended chat histories. RABR introduces an explicit mechanism for revising historical affective beliefs by linking new evidence to prior judgments while preserving both evidence trail and belief versions sequence.

Interpreting the affective meaning of experience also requires reasoning about the speakers and the conversational context. LaERC-S (Fu et al., 2025) uses two-stage learning to infer speaker charac-

teristics and track conversational emotions. Li et al. (2025) compare first-party emotion reports with human and LLM annotations, showing that inferred emotions can diverge from authors' reported states. ECFlow (Liang et al., 2026) combines appraisal-guided generation with graph-structured reinforcement learning to extract emotion–cause triplets and capture consistency across emotional transitions and shared events. RABR maintains an explicit version history of an agent's interpretation of past affective states, attributing each revision to the later evidence that supports it.

Tracking how these interpretations evolve as new evidence becomes available links affective inference to belief revision and temporal representation. AGM theory (Alchourrón et al., 2016) formalizes contraction and revision of logical theories, while Doyle's truth maintenance system (Doyle, 1979) records justifications and maintains beliefs as supporting assumptions change. Temporal databases distinguish when facts hold from when they are recorded (Snodgrass & Ahn, 1986). Bringing temporal representation into agent memory, Zep (Rasmussen et al., 2025) uses a temporal knowledge graph to integrate conversational and business data while retaining historical relationships. RABR adapts these ideas to affective judgments by indexing evidence-attributed revisions along two temporal dimensions: historical target time and belief-version time. Crucially, it separates revising a committed interpretation from suspending its operational use.

The consequences of such memory updates are particularly visible in personalized emotional support. ES-MemEval (Chen et al., 2026) evaluates memory in multi-session emotional support, including temporal reasoning, conflict detection, and user modeling. Fang et al. (2026) show that changes to remembered user profiles can alter affective interpretations of otherwise identical scenarios. Chu et al. (2026) use appraisal-guided reasoning chains to retrieve support responses according to psychological situation similarity. RABR complements this direction by preserving a traceable history of revised affective interpretations enabling later disclosures to refine the agent's understanding of earlier interactions and support personalization.

3 RABR Framework

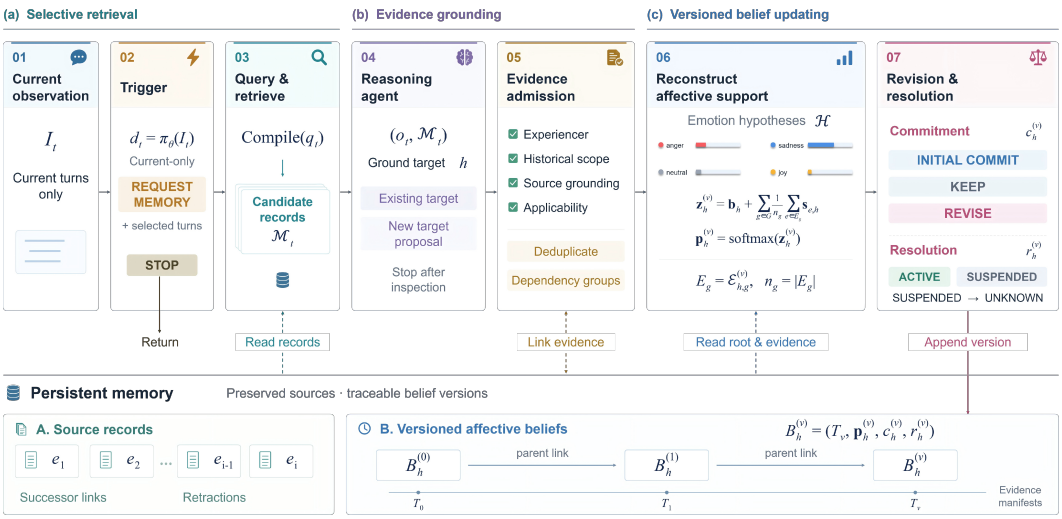

Figure 2: RABR framework. I_t: current observation; d_t: retrieval decision; q_t: query basis; $\mathcal{M}_t$: retrieved records; h: historical target; E_g: admitted evidence group; $B_h^{(v)} = (T_v, \mathbf{p}_h^{(v)}, c_h^{(v)}, r_h^{(v)})$: versioned affective belief.

3.1 Problem Setup and Overview

As Figure 2 shows, RABR revises an agent's interpretation of past affective states as new evidence arrives. A *historical target* h identifies a person and past episode: the episode time τ stays fixed, while the belief may change at later times T. For a fixed set $\mathcal{H}$ of K candidate affective interpretations, belief version v for target h is

$$B_h^{(v)} = \big(T_v, \mathbf{p}_h^{(v)}, c_h^{(v)}, r_h^{(v)}\big), \qquad r_h^{(v)} \in \{\textsc{Active}, \textsc{Suspended}\}, \tag{1}$$

where T_v is the storage time, $\mathbf{p}_h^{(v)}$ is normalized evidence support over $\mathcal{H}$, $c_h^{(v)}$ is the committed interpretation, and $r_h^{(v)}$ indicates whether it's available for downstream use. In summary, RABR has these: a compact trigger decides whether to retrieve history; a larger reasoning agent links and assesses evidence for the target; and a deterministic updater retains or revises the commitment and separately determines its availability. Original records and successive belief versions are preserved.

3.2 Selective Retrieval and Target Localization

At checkpoint t, the trigger π_θ receives only I_t: local turn identifiers, speakers, and verbatim text, without history, reference labels, provenance, or view metadata. It returns either a stop decision or a request-memory decision with basis q_t, where q_t selects current turns motivating retrieval, not historical records. A deterministic compiler converts q_t into a query, and the retriever returns a bounded candidate set:

$$d_t = \pi_\theta(I_t), \qquad \mathcal{M}_t = \begin{cases} \varnothing, & d_t = \textsc{Stop}, \\ \mathrm{Retrieve}\big(\mathrm{Compile}(q_t)\big), & d_t = \textsc{RequestMemory}(q_t). \end{cases} \tag{2}$$

The compact sequence-to-sequence trigger jointly generates the action and, for memory requests, its evidence basis as one JSON sequence. All parameters are trained with a standard sequence-to-sequence loss, without separate prediction heads (Appendix A). After retrieval, the reasoning agent examines $(I_t, \mathcal{M}_t)$ and either stops, links grounded evidence to an existing target by person and episode, or proposes a new target for validation and de-duplication before initialization. Evidence need not explicitly name an affective state; retrieval alone does not guarantee an update.

3.3 Evidence Admission and Organization

For each localized target, the reasoning agent constructs an *evidence application* linking a source record to the target and scoring support for every interpretation in $\mathcal{H}$. Records preserve observations; applications describe their target-specific relevance. Admission checks source grounding, the relevant person and episode, and applicability to the update. Only admitted applications can contribute and admission does not certify the affective interpretation. Records and target links remain unchanged: corrections create successor records, while retractions and supersessions are lifecycle transitions. Applications start pending and become incorporated when used in an update. Each evidence identifier contributes once; exact duplicates from the same source and context remain as non-contributing aliases. Applications sharing an evidential dependency form a group, preserving explicit justifications (Doyle, 1979); each group contributes one mean score vector.

3.4 Reconstructing Affective Support

At update v, let $\mathcal{E}_{h,g}^{(v)}$ contain the unique, active applications for target h in dependency group g incorporated by that update. In the reference mechanism, each application scores every interpretation in $\mathcal{H}$ using five ordered levels mapped to $\{-2, -1, 0, 1, 2\}$; subtracting its component mean gives the centered vector $\mathbf{s}_{e,h}$. For a fixed target and update, write $E_g = \mathcal{E}_{h,g}^{(v)}$, $n_g = |E_g|$, and let G contain the nonempty groups. Support combines their mean vectors with a fixed centered root profile $\mathbf{b}_h$ over the same interpretations:

$$\mathbf{z}_h^{(v)} = \mathbf{b}_h + \sum_{g \in G} \frac{1}{n_g} \sum_{e \in E_g} \mathbf{s}_{e,h}, \qquad \mathbf{p}_h^{(v)} = \mathrm{softmax}\Big(\mathbf{z}_h^{(v)}\Big). \tag{3}$$

The softmax normalizes relative evidence support; its values are not assumed to be calibrated probabilities over affective categories. Each update reconstructs support from the fixed root and active applications rather than accumulating prior support. Retractions recompute the affected group means; empty groups contribute nothing, and without active applications only the root remains. Earlier versions retain their evidence configurations and support vectors.

3.5 Revision and Operational Resolution

Let ℓ_v be the unique maximizer of $\mathbf{p}_h^{(v)}$ (none under an exact tie), P_v the largest support, and M_v the gap between the two largest supports. Before the first commitment, a unique leader must satisfy

$P_v \geq \alpha_{\text{resume}}$ and $M_v \geq \beta_{\text{resume}}$. This establishes an initial commitment and stores $\mathbf{p}_h^{(v)}$ as anchor $\mathbf{a}_h$. Until then, the commitment remains empty and unavailable, without an anchor comparison. For an existing commitment c, revision requires:

$$\textsc{Revise} \iff \ell_v \neq \varnothing, \quad \ell_v \neq c, \quad \frac{1}{2}\sum_{k=1}^{K} |p_{h,k}^{(v)} - a_{h,k}| \geq \delta, \quad p_{h,\ell_v}^{(v)} - p_{h,c}^{(v)} \geq \gamma. \tag{4}$$

Here δ sets the minimum support shift from the anchor, and γ the new leader's minimum advantage over c. Revise replaces the commitment with ℓ_v and the anchor with $\mathbf{p}_h^{(v)}$; keep retains both despite possible support changes. The anchor stays fixed between commitments, allowing evidence across updates to support a later revision; it does not enter support reconstruction (Equation 3). Operational resolution then determines whether the resulting commitment $c_v = c_h^{(v)}$ is available:

$$r_h^{(v)} = \textsc{Active} \iff \ell_v \neq \varnothing, \quad \ell_v = c_v, \quad P_v \geq \alpha_\rho, \quad M_v \geq \beta_\rho, \tag{5}$$

where ρ is the preceding resolution state and $(\alpha_\rho, \beta_\rho)$ selects hold thresholds after an active resolution and resume thresholds after a suspended resolution. Resume thresholds are no lower than their corresponding hold thresholds. All six parameters—δ, γ, α_{hold}, β_{hold}, α_{resume}, and β_{resume}—are selected on development data. Both retained and revised commitments may be suspended. Suspended outputs display unknown, denoting unavailability rather than another affective category in $\mathcal{H}$; the commitment and evidence history remain stored.

3.6 Versioned Memory and Revision Attribution

Each update stores a new version with support, commitment, resolution, a parent reference, and a manifest of incorporated evidence and dependency groups. Evidence additions, retractions, and supersessions are recorded as lifecycle transitions; revision events link the old and new commitments to these records. Following the valid-time/transaction-time distinction (Snodgrass & Ahn, 1986), queries for episode time τ and information time T return the latest version stored by T:

$$B_h(\tau; T) = B_h^{(v^\star)}, \qquad v^\star = \max\{v : T_v \leq T\}. \tag{6}$$

No belief is returned if none exists by T. Historical answers use stored versions, excluding evidence that arrived later or was still pending at T. Current queries return the latest version. Downstream modules may register direct dependencies on belief versions; revisions expose the identifiers of affected registered artifacts for subsequent handling.

4 RABRBench: Retrieval Demand and Retrospective Revision

To fill the data gap, we introduce RABRBench to evaluate two linked stages of retrospective affective belief revision: deciding whether the current observation warrants access to prior interaction evidence, and using that evidence to update an affected historical interpretation. The benchmark combines controlled contrasts with source-preserving long histories to evaluate retrieval decisions and final-state revision; a separate target-ranking diagnostic measures localization difficulty.

4.1 Task Design and Benchmark Construction

The first task evaluates retrieval demand from the current observation alone. The trigger requests memory when eligible history is needed, stops when the current evidence is sufficient, or reports that the decision is not identifiable. A memory request also identifies the visible evidence-turn basis that motivates retrieval. The second task begins after memory access. Given a current evidence window and 12–49 eligible historical items, a system uses historical evidence to predict the final affective state of the referenced target. We derive the keep/revise action from the initial–final label pair. The reported joint score combines retrieval-decision quality with final-affect correctness for that target.

We construct three complementary packages around these tasks. The short-context package (SC) uses a handcrafted core and controlled expansion to isolate retrieval decisions. The extended-context package (EC) adapts EvoMem histories from MemoryAgentBench (Hu et al., 2026) to test the same

decision boundary with long, distractor-rich histories. The long-context package (LC) draws on AMI meeting transcripts, LoCoMo, and Multi-Session Chat (Carletta et al., 2006; Maharana et al., 2024; Xu et al., 2022) to test final-state revision with longer histories and many candidate targets. SC and EC instantiate the retrieval-demand task, whereas LC instantiates the post-retrieval task. Appendix C gives the full construction, filtering, and evidentiary boundaries.

Table 1: Three complementary RABRBENCH packages.

Package	Primary role	Construction	Scale	Supervision
SC	Controlled retrieval mechanisms	Handcrafted core plus filtered expansion	311 families; 622 cases	345 request; 207 stop; 70 not identifiable
EC	Long-history retrieval specificity	Reverse-constructed matched histories	5 histories; 1,012 cases	517 request; 495 stop
LC	Locate & revise final state	Source-grounded context	64 families; 2,411 cards	1,245 keep; 1,166 revise; 707 evaluation gold

4.2 DATASET COMPOSITION AND PROVENANCE

Table 1 reports the most important information for interpreting the benchmark. We describe detailed construction, filtering, and dependency-aware train/dev/test partitions in Appendix C. We also reviewed supervision coverage, and analyzed affect composition separately in Appendix C.6.

The three packages answer different questions. SC isolates the local retrieval mechanism; EC measures long-history demand and specificity; LC evaluates final-state revision with a separate target-ranking diagnostic. We report package-specific denominators and supervision boundaries.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Data. We evaluate on 728 test trajectories from RABRBENCH: 36 from SC, 209 from EC, and 483 from LC. The EC set includes 102 no-target controls, which remain in the evaluation but do not contribute to final-affect scores. Final-affect evaluation uses the subsets with reviewed reference labels: 35 trajectories from SC, 100 from EC, and 467 from LC. Table 10 details the evaluation subsets and exclusions. For affective comparison, we focus on the LC test, which has balanced emotion labels.

Implementation. The training-free downstream methods use a *T5Gemma-2-270M* trigger (Zhang et al., 2025) and *Qwen3.8-27B-FP8* (Qwen Team, 2026) for evidence interpretation and final-state prediction. *RABR (tuned)* uses Qwen3.5-4B throughout its learned trigger, request, review, and updater stages. Its training combines LC trajectory supervision with SC/EC source-labeled trigger and downstream examples. Downstream inference runs on NVIDIA A100 80G GPUs. We provide model and inference settings in Appendix A and the RABR (tuned) data-construction and training procedure in Appendix B.

Baselines and method variants. We evaluate two RABR variants against four baselines. *Fixed Initial* retains the initial affective judgment without updating it. *Full-history Reclassification* reads all eligible history at each memory request and directly predicts the target's final affect. *Mem0* (Chhikara et al., 2025) maintains its own memory store and predicts final affect from memories retrieved using the compiled trigger query. *RABR (tuned)* learns the complete trigger–request–review–updater trajectory using Qwen3.5-4B and maintains its own evolving affect label. Our two variants apply the same evidence-based belief updater but differ in the history provided: *RABR (Full history)* uses all eligible history, whereas *RABR (Retrieved history)* uses at most three items selected from the initialization history by BM25 (Robertson & Zaragoza, 2009). Both variants retain the complete current observation. The three original baselines and both RABR variants share the T5Gemma-2-270M trigger; RABR (tuned) uses its own learned trigger. Detailed settings appear in Appendix A.

Evaluation protocol. We evaluate SC, EC, and LC separately. Methods start from the same initial judgments and maintain independent belief states. The three original baselines and both RABR

variants share precomputed T5Gemma-2-270M trigger outputs and Qwen3.8-27B-FP8 downstream inference. RABR (tuned) runs its Qwen3.5-4B trigger and downstream stages with its own model state. Each trajectory is processed chronologically; downstream inference occurs only when memory is requested. Trigger scores use annotated observations in SC/EC and each new dialogue turn after initialization in LC.

Metrics. **Retrieval decision accuracy (RDA)** scores correct retrieval actions; **Retrieval-cue overlap (RCO)** scores overlap between predicted and annotated cue turns at reference requests. **Affect** requires an exact final-label match for the referenced historical target, whereas **revision** scores the keep/revise decision. **Joint** multiplies binary affect correctness by the trajectory's mean retrieval-decision score. Retrieval metrics and joint weight related families equally; affect and revision average eligible trajectories. All scores lie in $[0, 1]$, and unavailable predictions score zero on fixed subsets. Appendices A, D, and E give the full rules.

Table 2: T5Gemma-2-270M trigger performance on SC, EC, and LC for the training-free downstream methods.

Package	trigger events	RDA	RCO	Req. recall	Stop specificity	Balanced action acc.
SC	36	0.974	0.478	0.972	0.286	0.629
EC	107	0.832	0.813	0.832	0.971	0.901
LC	8218	0.838	0.974	0.998	0.626	0.812
Macro-average	–	0.881	0.755	0.934	0.627	0.781

Notes. **RDA** (retrieval decision accuracy) averages retrieval-decision correctness. **RCO** (retrieval-cue overlap) evaluates the current turns motivating retrieval, not the retrieved history: at reference request boundaries, it is the overlap $|B \cap R|/|B \cup R|$ between the evidence turns B the trigger returns and the evidence turns R the reference marks. A wrong action, a returned turn outside the visible prefix, or a malformed output scores zero. **Req. recall**: true positive rate, $\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. **Stop specificity**: true negative rate, $\mathrm{TN}/(\mathrm{TN} + \mathrm{FP})$. **Balanced action acc.**: their mean. All scores use three decimal places; higher is better.

5.2 Main Results

Trigger performance. The shared T5Gemma-2-270M trigger recognizes designated retrieval opportunities across all three packages (Table 2). On LC it also identifies the evidence turn that motivates the request, so finding a positive boundary and grounding its cue are closely aligned. The harder decision is package dependent: SC combines high request recall with low stop specificity, whereas EC shows the reverse balance. Thus, retrieval demand cannot be summarized by request recall alone. The LC stop score reflects its sequential structural labels (Appendix D). These differences matter operationally because unnecessary requests still activate the much larger downstream model and expand the evidence-processing workload.[1]

Affective belief revision. LC gives the clearest final-affect result (Table 3). RABR (Full history) leads on affect and joint, with an 11.8% relative joint gain over the strongest baseline; bounded retrieval retains a smaller gain. LC revision agreement also rises with history. EC favors RABR (tuned), leading all three downstream scores with less input context. On LC, it approaches retrieved history on affect but trails on joint and revision. Deciding whether to revise and identifying final affect are distinct tasks. Full-history Reclassification and Mem0 each reach 0.400 affect, versus 0.114 for both deterministic RABR variants and 0.286 for RABR (tuned). Yet RABR (Retrieved history) reaches 0.682 revision agreement. The two updater variants achieve identical affect despite different history scopes and share the trigger with direct reclassification. This suggests difficulty interpreting SC evidence or assigning complete label sets; mixed-affect references further complicate exact-set scoring. The strict-affect subset and aggregate scores cannot isolate the cause. RABR (tuned) reaches 0.273 revision agreement, below Fixed Initial's 0.409, so its EC advantage does not extend uniformly to SC.

[1] RABR (tuned) uses its own trigger. Across SC, EC, and LC, its package macro-average RDA is 0.899, RCO 0.738, request recall 0.932, stop specificity 0.824, and balanced action accuracy 0.878.

Table 3: Final-affect and revision results for four baselines and two RABR variants on reference-annotated test trajectories.

	Method	Affect score	Joint score	Revision score	Mean downstream input/output tokens†
SC	Fixed Initial	0.000	0.000	0.409	N/A
	Full-history Reclassification	0.400	0.193	0.409	1,854/176
	Mem0	0.400	0.193	0.409	1,560/143
	RABR (Full history)	0.114	0.066	0.591	1,874/200
	RABR (Retrieved history)	0.114	0.066	0.682	1,850/192
	RABR (tuned)	0.286	0.145	0.273	520/20
EC	Fixed Initial	0.240	0.190	0.260	N/A
	Full-history Reclassification	0.340	0.280	0.290	37,278/209
	Mem0	0.360	0.310	0.320	4,657/200
	RABR (Full history)	0.320	0.270	0.450	34,538/181
	RABR (Retrieved history)	0.380	0.330	0.520	13,137/189
	RABR (tuned)	0.420	0.390	0.630	9,106/22
LC	Fixed Initial	0.527	0.390	0.527	N/A
	Full-history Reclassification	0.743	0.496	0.754	32,426/213
	Mem0	0.724	0.430	0.730	4,638/163
	RABR (Full history)	0.872	0.605	0.983	25,543/253
	RABR (Retrieved history)	0.812	0.571	0.929	8,982/214
	RABR (tuned)	0.814	0.541	0.863	1,686/20

Notes. **Affect** and **Revision** are proportions of correct final-affect and KEEP/REVISE predictions on their respective fixed evaluation subsets. **Joint** averages per-trajectory products of mean retrieval score and binary Affect correctness with equal weight per family (groups of related trajectories). Failed or unavailable predictions score zero. All scores use three decimal places; higher is better. Evaluation subset sizes are reported in Table 10. † are per completed downstream call; output-token definitions differ across methods (Appendix G). N/A denotes no downstream call. Scoring rules and execution scope appear in Appendices E and G.

5.3 Analysis and Ablation Studies

LC affect accuracy by revision requirement. To see what the LC affect score captures, we separate its strict test trajectories by the source-declared keep/revise action (Table 4). RABR (Full history) reaches 0.996 on keep and 0.733 on revise cases. RABR (Retrieved history) reaches 1.000 and 0.602, while RABR (tuned) reaches 0.972 and 0.638. The full-history variant's overall advantage over the retrieved-history variant is therefore concentrated in cases requiring a different affective category. It reads more input per completed downstream call (Table 3). These are descriptive outcomes under the methods' evaluated pipelines; the tuned method also uses its own trigger.

Table 4: Exact final-affect accuracy for the three RABR methods on LC, conditioned on the source-declared revision action.

RABR method	KEEP ($n = 246$)	REVISE ($n = 221$)	Overall ($n = 467$)
Full history	0.996	**0.733**	**0.872**
Retrieved history	**1.000**	0.602	0.812
Tuned	0.972	0.638	0.814

Notes. Cells report exact final-label accuracy (three decimals) on fixed reference subsets, counting failed/unavailable predictions as errors, regardless of predicted revision. Source actions indicate whether reference final labels differ from initial labels. Overall matches LC Affect in Table 3; Appendix F lists the eight final-affect categories within REVISE.

The learned trajectory and token use. Table 4 shows RABR (tuned) reads less input than either updater variant in all three packages, most markedly on LC. In EC, this compact trajectory produces the strongest affect, joint, and revision scores. On LC its overall affect nearly matches bounded retrieval with far less input, but the similar totals hide different outcomes: tuned performs better on revise cases and worse on keep cases. It trails on revision and joint, which also reflect its action decisions and own trigger. SC exposes the other side of the trade-off: its very short trajectory produces weaker affect than direct reclassification and weaker revision than retaining the initial judgment. Trajectory supervision structures retrieval, review, and update into concise decisions; the EC and LC results show that short outputs can still support strong final-affect prediction.

Variation across final-affect categories. The LC cases requiring revision span all eight final-affect categories. The RABR variants differ across categories and in their aggregate revise accuracy. For example, RABR (tuned) reaches 0.929 on *surprise* and zero on *other*; RABR (Full history) reaches 0.857 and 0.385. The category scores in Appendix F locate this variation.

Retrieval method and depth. The matched retrieval ablation holds trigger decisions and queries fixed across rankers and depths (Table 5). Affect improves from one to three returned items for both rankers, then falls at five despite the larger evidence window, revealing a nonmonotonic context effect. Revision need not move with affect: dense cosine has its highest revision agreement at five while its affect has already declined. This pattern is consistent with extra context helping a change decision without reliably identifying the replacement affective category. Dense cosine improves affect over BM25 at matched depths, but its warmed retrieval calls take roughly 160 times as long.

Table 5: Retrieval method and depth ablation on matched held-out trajectories.

Metric	BM25			Dense cosine		
	$k=1$	$k=3$	$k=5$	$k=1$	$k=3$	$k=5$
Affect	0.614	0.672	0.647	0.723	0.739	0.689
Revision	0.844	0.856	0.856	0.883	0.883	0.901
Output tokens / retrieval	38.0	162.5	280.5	29.2	167.5	283.3
Retrieval time (ms)	1.8	1.8	1.8	291.0	292.3	291.2

Effect of trigger gating. Selective gating removes more downstream prompt tokens than completion tokens (Table 6). Avoiding a request saves the history-bearing prompt and the associated downstream call, giving a 37.4% call reduction and a 47.6% reduction in downstream tokens on this workload. This makes the retrieval decision a direct source of computational savings in the revision pipeline when prompts carry substantial history.

Table 6: Downstream operation and token accounting.

Measure	Always request	trigger-gated
Calls (times)	345	216 (↓37.4%)
Input/prompt tokens	1,601,637	832,995 (↓48.0%)
Output/completion tokens	47,829	30,904 (↓35.4%)
Total tokens	1,649,466	863,899 (↓47.6%)

6 Conclusion

We introduce RABR and RABRBench, establishing retrospective affective belief revision as an explicit task in affective computing. RABR links delayed evidence to historical judgments through selective memory access and evidence-attributed, versioned updates. It separates preserved event records from revisable interpretations, retaining earlier beliefs alongside current interpretations, and distinguishes revising a judgment from suspending its use. To our knowledge, RABRBench is the first affective-computing benchmark to jointly test when later evidence warrants retrieving past interactions and whether it supports keeping or revising historical affective judgments. Its SC, EC, and LC packages span controlled cases, extended histories, and long-context dialogue, with separate retrieval, cue-grounding, revision, and final-affect scores.

RABR methods lead revision agreement across all evaluated packages. On LC, RABR (Full history) also leads final-affect prediction and improves joint by 11.8% over the strongest baseline. A separate gating comparison shows that selective memory access reduces downstream calls and token use. These findings distinguish two requirements for successful revision: recognizing that an earlier judgment should change and identifying the interpretation supported by the evidence.

More broadly, this work advances a retrospective perspective on affective computing: understanding a person's affective states requires interpreting current behavior and refining earlier judgments as relevant context emerges. This provides a foundation for long-term user modeling, personalized emotional support, and affect-sensitive interaction that can reconsider earlier impressions through later disclosures. Together, RABR and RABRBench make revising past affective interpretations an explicit, measurable, and traceable component of affective reasoning, laying the groundwork for agents whose understanding of a person develops through continued interaction.

LIMITATIONS

Evaluation scope. RABRBENCH combines controlled retrospective scenarios with source-preserving dialogue. Most delayed disclosures in LC are constructed, and shared generation patterns and explicit affect cues may simplify some cases. The evaluation also follows earlier development in the same domain, with reference annotations refined after initial generation. Evaluation on independently collected longitudinal conversations is needed to establish generalization beyond this construction and development setting, particularly when disclosures are less explicit or more ambiguous.

Affective representation. The current implementation uses a fixed affective-label inventory and interpretable support scores to make revisions explicit and traceable. These scores express relative evidential support rather than calibrated probabilities over affective categories, and revision quality depends on the reasoning agent's interpretation of the available evidence. Extending this formulation to richer representations of mixed affective states, intensity, and person-specific expression remains an important direction.

Interactive applications. Our experiments evaluate the revision of historical affective judgments rather than its effects on subsequent human–agent interaction. Longitudinal user studies are needed to examine how revised interpretations contribute to user modeling, affect-sensitive tutoring, and emotional support. Active clarification offers a complementary direction for exploring how users can confirm or refine an agent's interpretation, connecting retrospective affective revision to the ongoing development of shared understanding.

AI USE STATEMENT

Generative AI tools were used to generate controlled synthetic dataset components, support source-grounded dataset review and annotation, assist with some code implementation, and interpret diagnostic results. We reviewed all AI-assisted outputs against retained source data, run artifacts, or executable checks. The authors take responsibility for the final content, claims, and artifacts produced with AI assistance.

REFERENCES

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*, pp. 195–217. Springer International Publishing, Cham, 2016. ISBN 978-3-319-20451-2. doi: 10.1007/978-3-319-20451-2_13. URL https://doi.org/10.1007/978-3-319-20451-2_13.

Lisa Feldman Barrett and Elizabeth A. Kensinger. Context is routinely encoded during emotion perception. *Psychological Science*, 21(4):595–599, 2010. doi: 10.1177/0956797610363547. URL https://doi.org/10.1177/0956797610363547. PMID: 20424107.

Timothy W. Bickmore and Rosalind W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Comput.-Hum. Interact.*, 12(2):293–327, June 2005. ISSN 1073-0516. doi: 10.1145/1067860.1067867. URL https://doi.org/10.1145/1067860.1067867.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: A pre-announcement. In Steve Renals and Samy Bengio (eds.), *Machine Learning for Multimodal Interaction*, pp. 28–39, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32550-5.

Tiantian Chen, Jiaqi Lu, Ying Shen, and Lin Zhang. Es-memeval: Benchmarking conversational agents on personalized long-term emotional support. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 5810–5821, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792143. URL https://doi.org/10.1145/3774904.3792143.

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

Yuqi Chu, Lizi Liao, Jinggui Liang, Boyang Li, and Richang Hong. Beyond semantic similarity: Appraisal-guided chain-of-thought reasoning and retrieval for multimodal emotional support conversations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17104–17119, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.844. URL https://aclanthology.org/2026.findings-acl.844/.

Sidney K. D'Mello and Arthur C. Graesser. Malleability of students' perceptions of an affect-sensitive tutor and its influence on learning. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pp. 432–437. AAAI Press, 2012. URL https://cdn.aaai.org/ocs/4384/4384-21571-1-PB.pdf.

Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(79)90008-0. URL https://www.sciencedirect.com/science/article/pii/0004370279900080.

Xi Fang, Weijie Xu, Yuchong Zhang, Scott Nickleach, Stephanie Eckman, and Chandan K. Reddy. The personalization trap: How user memory alters emotional reasoning in LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 511–529, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-391-3. doi: 10.18653/v1/2026.acl-short.43. URL https://aclanthology.org/2026.acl-short.43/.

Yumeng Fu, Junjie Wu, Zhongjie Wang, Meishan Zhang, Lili Shan, Yulin Wu, and Bingquan Liu. LaERC-S: Improving LLM-based emotion recognition in conversation with speaker characteristics. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 6748–6761, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.451/.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 156259–156291, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fd1eff9dd295df50a41f2521942fa31d-Paper-Conference.pdf.

Dongjin Kang, Sunghwan Kim, Taeyoon Kwon, Seungjun Moon, Hyunsouk Cho, Youngjae Yu, Dongha Lee, and Jinyoung Yeo. Can large language models be good emotional supporter? mitigating preference bias on emotional support conversation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15232–15261, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.813. URL https://aclanthology.org/2024.acl-long.813/.

Jiayi Li, Yingfan Zhou, Pranav Narayanan Venkit, Halima Binte Islam, Sneha Arya, Shomir Wilson, and Sarah Rajtmajer. Can third parties read our emotions? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21478–21499, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1042. URL https://aclanthology.org/2025.acl-long.1042/.

Qiao Liang, Ying Shen, Yao Liu, Tiantian Chen, and Lin Zhang. Why do emotions change? appraisal-guided reasoning for emotion–cause triplet extraction in conversations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pp. 11750–11772, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.539. URL https://aclanthology.org/2026.acl-long.539/.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. Towards emotional support dialog systems. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3469–3483, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.269. URL https://aclanthology.org/2021.acl-long.269/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

David Melhart, Matthew Barthet, and Georgios N Yannakakis. Can large language models capture video game engagement? *arXiv preprint arXiv:2502.04379*, 2025.

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*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

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.

Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, September 2009. ISSN 1554-0669. doi: 10.1561/1500000019.

Richard Snodgrass and Ilsoo Ahn. Temporal databases. *Computer*, 19(9):35–42, September 1986. ISSN 0018-9162. doi: 10.1109/MC.1986.1663327. URL https://doi.org/10.1109/MC.1986.1663327.

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.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5180–5197, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.356. URL https://aclanthology.org/2022.acl-long.356/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement

learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Georgios N. Yannakakis and Julian Togelius. *Player Modeling*, pp. 315–335. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-83347-2. doi: 10.1007/978-3-031-83347-2_10. URL https://doi.org/10.1007/978-3-031-83347-2_10.

Biao Zhang, Paul Suganthan, Gaël Liu, Ilya Philippov, Sahil Dua, Ben Hora, Kat Black, Gus Martins, Omar Sanseviero, Shreya Pathak, Cassidy Hardin, Francesco Visin, Jiageng Zhang, Kathleen Kenealy, Qin Yin, Xiaodan Song, Olivier Lacombe, Armand Joulin, Tris Warkentin, and Adam Roberts. T5gemma 2: Seeing, reading, and understanding longer, 2025. URL https://arxiv.org/abs/2512.14856.

Jinming Zhang and Yunfei Long. MLD-EA: Check and complete narrative coherence by introducing emotions and actions. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 1892–1907, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.129/.

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

A Model and Hyperparameter Details

Models and inference. Fixed Initial, Full-history Reclassification, Mem0, and the two RABR variants share a T5Gemma-2-270M trigger. Except for Fixed Initial, which makes no downstream prediction call, these methods use Qwen3.8-27B-FP8 for evidence interpretation or final-state prediction. Downstream inference runs on NVIDIA A100 GPUs through vLLM with temperature 0, top-p 1, reasoning mode disabled, and a context limit of 131,072 tokens. The original downstream runs allow at most 4,096 output tokens; the graph-mode continuation for 114 retained LC retrieved-history failures uses a 16,384-token output cap and a 2,700-second client deadline. RABR (tuned) instead uses Qwen3.5-4B for its learned trigger, request, review, and updater stages. Each stage uses greedy decoding with reasoning mode disabled and at most 128 generated tokens.

Shared trigger training. The T5Gemma-2-270M trigger generates the retrieval action and its visible evidence-turn basis together as one sequence. We fine-tune all parameters with a sequence-to-sequence objective in two stages. The first stage uses 1,684 LC request examples for training and 244 for development. The second stage initializes from the first stage's epoch-2 checkpoint and adds stop examples from the same current windows. After deduplication and removal of conflicting labels, the second-stage training set contains 2,188 examples (1,684 requests and 504 stops); its development set contains 332 (244 requests and 88 stops).

The second stage uses a fresh optimizer, a learning rate of 10^{-5}, three epochs, batch size one, eight gradient-accumulation steps, weight decay 0.01, and seed 20260912. It uses FP32 parameters, BF16 mixed precision, gradient checkpointing, and inputs of at most 8,192 tokens. Development balanced action accuracy selects the checkpoint, with evidence-basis F1 and then the earlier epoch as tie-breakers; the selected checkpoint is epoch 3. Evaluation keeps this checkpoint fixed. Trigger inference uses greedy decoding, at most 128 generated tokens, and a maximum source length of 8,192 tokens without truncation.

Initialization. LC uses annotated initial-affect labels, whereas SC and EC use shared initial estimates derived from source evidence available before the current observation. SC and EC retain unknown when those estimates do not identify a committed affective label.

Method inputs. For the five shared-trigger methods, trigger actions, compiled queries, and request evidence-turn bases are precomputed once per candidate and reused unchanged. Fixed Initial retains the frozen initial judgment. At a shared request-memory boundary, Full-history Reclassification reads all eligible history and directly predicts final affect. Mem0 (Chhikara et al., 2025) uses version 2.0.20, the *multi-qa-MiniLM-L6-cos-v1* embedding model, and a separate Qdrant vector store per trajectory. It ingests historical items before the corresponding current observation and retrieves at most ten memories per search.

RABR (Full history) reads all eligible history at a request, interprets target-relative evidence, and applies the deterministic updater to its own belief state. RABR (Retrieved history) uses the compiled trigger query to select the BM25 top three from the history available at initialization. Both variants retain every non-initialization turn visible at the checkpoint in the complete current observation, in source order, and use the same belief-update parameters below. The trigger's evidence-turn basis explains the request; it does not replace either retrieved history or the current observation. Each method processes its trajectory in order. Table 7 summarizes historical access and judgment updates across all six methods.

Belief-update parameters. RABR uses a uniform root profile and the five support levels $\{-2, -1, 0, 1, 2\}$ described in Section 3.4. The revision thresholds are $\delta = 0$ and $\gamma = 0$. Operational resolution uses hold thresholds $\alpha_{\text{hold}} = 0.125$ and $\beta_{\text{hold}} = 0$, and resume thresholds $\alpha_{\text{resume}} = 0.31875$ and $\beta_{\text{resume}} = 0.15$. Applications derived from the same source unit share a dependency group, and repeated records of that unit are treated as duplicate aliases. These settings define the reference configuration used in the method comparison.

Appendix H derives the evidence-reconstruction properties and the consequences of this reference configuration.

Table 7: Historical access and judgment-update configurations for four baselines and two RABR variants at a memory request.

Method	Historical access	Judgment update
Fixed Initial	None	Retain initial judgment
Full-history Reclassification	All eligible history	Direct final-affect reclassification
Mem0	Up to ten retrieved memories	Memory-grounded final-affect prediction
RABR (tuned)	BM25 top three first, then the remaining declared memory pool	Learned review and updater with method-local state
RABR (Full history)	All eligible history	Deterministic evidence-based belief update
RABR (Retrieved history)	BM25 top three from initialization history	Deterministic evidence-based belief update

Notes. Both RABR variants retain the complete current observation. RABR (Full history) and Full-history Reclassification receive the same eligible history at a request; the former interprets target-relative evidence and applies the versioned belief updater. RABR (tuned)'s request changes the order of the declared memory pool, which remains visible in full.

B RABR (TUNED) TRAJECTORY CONSTRUCTION AND TRAINING

Source cases and teacher rendering. We construct the supervised trajectories from the LC training and development splits, which contain 1,684 and 244 cases, respectively. Each case provides the current dialogue scene, a historical memory pool, an initial affect label, a delayed-evidence turn, and a recorded final label and keep/revise decision. The teacher is given the recorded endpoint and renders a path to that endpoint; the endpoint is supervision for trajectory synthesis rather than a model prediction. We use `deepseek-flash` with reasoning disabled to render trigger, request, review, and updater content. A Qwen3.5-4B calibrator and deterministic checks assess whether cited units exist, the compiled request retrieves the intended memory in the BM25 top three, the review supports the decision, and the updater agrees with the recorded endpoint. Failed checks are returned to the teacher for repair, for at most four attempts. The test split is excluded from rendering and training.

Packing and short-format projection. The rendered path is packed as an ordered exchange: system instructions and current scene, assistant trigger and request, an environment return containing retrieved memories, then assistant review and updater. The request's target memory identifier is not exposed in the input scene. Case-level consistency checks exclude conflicting or incomplete training trajectories, leaving 1,669 training cases; all 244 development cases remain separate. We project each accepted path into four supervised stage records, producing 6,676 training records and 976 development records. The final affect reading is determined by the updater and is not an additional training target.

Each stage is a short completion. Trigger outputs `T=REQUEST_MEMORY` or `T=ROOT_STOP`. Request outputs `Q=tNNNN,...|phrase1 ; phrase2`, pairing cited current-scene turns with short retrieval phrases. Review outputs `V=ASSESSMENT|CONTRIBUTION|mNNNN,...`, including the memory units on which its assessment rests. Updater outputs `U=REVISE|LABEL` or `U=KEEP|-` for LC; SC/EC permit a sorted, nonempty label set in the revision target. The `tNNNN` and `mNNNN` identifiers are case-local aliases for scene turns and memory items; the projection removes fields that do not affect the four decisions. Each stage record contains its preceding conversational context and its own completion, with loss masked on the context and applied only to the assistant completion.

Retrieval during a trajectory. The environment compiles the request's cited scene turns and grounding phrases into a BM25 query against the case's declared memory pool. It places the top three ranked memories first and appends the remaining declared memories before the review stage. Thus, the request determines evidence order, while review can inspect the whole declared pool. The short-format projection preserves the compiled query and top-three ranking of the packed trajectory; we checked this for all 1,913 train and development requests. At evaluation, the same retrieval interface connects the learned stages, and RABR (tuned) maintains its own committed affect label.

Full-chain optimization. We fine-tune Qwen3.5-4B with one LoRA adapter shared by all four stages. The full-chain run uses two NVIDIA A100 80GB GPUs on one node, BF16 computation, gradient checkpointing, and a 16,384-token sequence limit. All four records from a case contribute to one case-level objective. With $\ell_T, \ell_Q, \ell_V, \ell_U$ denoting mean completion-token losses for trigger, request, review, and update, respectively, the per-case loss is

$$\mathcal{L}_{\text{case}} = \ell_U + \tfrac{1}{3}(\ell_T + \ell_Q + \ell_V). \tag{7}$$

The two GPU ranks divide the stage records of a case and combine gradients for the same objective. We average case losses within each optimizer step. The run covers all 1,669 training cases in one epoch, with an effective batch of 16 cases and 105 optimizer steps. Table 8 gives the optimization settings.

Table 8: Training settings for RABR (tuned).

Setting	Value
Base model	Qwen3.5-4B
Adapter	LoRA rank 16, alpha 32, dropout 0; attention, MLP, and Gated-DeltaNet projections
Optimizer	AdamW; learning rate 10^{-5}; $\beta_1 = 0.9$, $\beta_2 = 0.95$; weight decay 0
Schedule	Cosine decay; 3 warmup steps for the full-chain objective and 1 for each trigger-action block
Precision and context	BF16; 16,384-token maximum; gradient checkpointing
Gradient clipping and seed	Maximum norm 1.0; seed 20260920
Full-chain supervision	2 A100 80GB GPUs; 1,669 cases; 1 epoch; effective case batch 16; 105 steps
trigger-action supervision	1 A100 80GB GPU; 334 paired LC cases (21 steps), 209 SC/EC cases (14 steps), and 440 SC cases (28 steps); 1 epoch per block; effective case batch 16
SC/EC downstream supervision	1 A100 80GB GPU; 101 cases, 303 stage records; 1 epoch; effective case batch 16; 7 steps

Trigger-action data and inference. The full-chain cases supervise requests at the delayed-evidence boundary. The same training procedure includes balanced stop supervision: we deterministically select 334 LC training cases across 39 source families and pair each request boundary with a pre-evidence stop boundary from the same case, yielding 334 `REQUEST_MEMORY` and 334 `ROOT_STOP` completions. SC and EC contribute 209 source-labeled trigger examples: 88 SC cases (55 requests and 33 stops) and 121 EC cases (62 requests and 59 stops). The SC trigger data also include all 440 action-identifiable SC training cases (274 requests and 166 stops); 54 SC cases without an identifiable action are excluded. Development and test cases are excluded from training. Each trigger data block is optimized for one epoch with an effective case batch of 16. At inference, LC uses its balanced trigger weights, and SC and EC use the SC/EC-trained trigger weights. A valid memory request activates request, review, and update; otherwise the current affect label is retained.

SC/EC downstream data. Source annotations provide request cues, allowed review decisions, and final Phase 1 affect label sets for SC and EC. Of 117 selected training requests, 101 yield complete request, review, and updater examples (48 SC and 53 EC cases), for 303 stage records. The updater emits `U=KEEP|-` or `U=REVISE|LABEL[,LABEL...]`, with unique labels in a fixed inventory order. We assign equal loss weight to the three downstream stages and train for one epoch at an effective case batch of 16 and learning rate 10^{-5}. SC and EC use the resulting downstream weights; LC uses the full-chain weights.

C RABRBench Construction

C.1 Shared Boundary and Package-Specific Targets

Every RABRBench package presents a current observation before historical material becomes accessible. SC and EC supervise the decision to request that material: a request is warranted when

the current observation makes eligible history relevant to explaining, confirming, or revising a past affective interpretation. A stop label means that retrieval is unnecessary at that boundary, and a not-identifiable label records an unresolved decision. Evidence-turn annotations identify the visible basis for a request without revealing the historical answer to the trigger.

LC supplies a referenced historical target among eligible memory items and supervises recovery of its final affective state. Each LC case has one source-declared delayed-evidence boundary labelled as a request-memory action; sequential scoring labels every other dialogue arrival as a stop action. The reference action is keep when the initial and final labels agree and revise otherwise; it is derived from the label pair. Target-ranking difficulty is reported separately from the final-affect and retrieval-decision scores.

Each case retains its construction provenance, supervision type, and source lineage. Evidence use is target-relative: the same observation can bear differently on different historical claims. Related variants and repeated views remain linked through their declared dependency units. The package-specific split rules below use those links while recording the known EC source overlap.

C.2 SC: Handcrafted Core and Filtered Expansion

The short-context package begins with 120 handcrafted examples designed to express the basic contrast between a new observation that warrants revisiting history and one that does not. These examples establish the task semantics and the compact case form. A larger expansion then varies the situation, activity, setting, affective interpretation, and evidence pattern while retaining the same two-observation family structure.

Expansion candidates pass through a staged pipeline. First, a family plan fixes the shared history and two independently written current observations. A renderer realizes the dialogue and declared memory. Family-level validation rejects cross-field inconsistencies or cases that expose historical-only information in the current scene. A blind review then assesses the final text, followed by a fresh action and evidence annotation. The retained set contains complete families that pass every stage. The final SC package contains 311 families and 622 cases, including the handcrafted core and the retained portion of the expansion.

At runtime, the root view exposes only the current scene. A memory request reveals the declared historical material through a deterministic post-request view. SC therefore tests the request decision under this information boundary.

C.3 EC: Reverse-Constructed Extended Histories

EC begins from source-preserved EvoMem dialogue histories distributed through MemoryAgentBench (Hu et al., 2026). Each original case provides a current observation for which no retrospective lookup is required. Reverse construction creates a matched warranted counterpart by first fixing the intended retrospective relation, then inserting one controlled earlier history unit and a later current cue that makes that unit relevant. The source dialogue remains unchanged, and the additions retain explicit lineage to their original case.

The five history identifiers are aggregate temporal containers, not five independent examples. Only the first source-preserving case in each history has no earlier unit; other cases contain prefixes of up to 48 units. Controlled counterparts contain the additional unit and therefore range from 1 to 49 historical units. Across the 1,012 retained cases, 871 contain 24–37 units. Original and controlled counterparts are alternative single-observation cases, not consecutive steps of an executed belief-revision trajectory.

C.4 LC: Long-Context Target and Final-State Cards

LC contains 2,411 cards in 64 source families. The surrounding material comes from AMI meeting transcripts, LoCoMo, and Multi-Session Chat (Carletta et al., 2006; Maharana et al., 2024; Xu et al., 2022). Each card exposes a current scene and retains 12–49 eligible memory items. Its sidecar records the affected historical target, the delayed-evidence turn, an initial affect label, and a final affect label. All eight Phase 1 labels occur in every split.

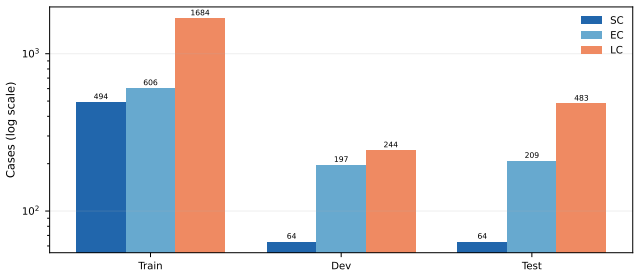

Figure 3: Dependency-aware train/development/test scale across the three RABRBENCH packages. Counts are cases. The vertical axis is logarithmic; labels above bars give exact counts.

Table 9: Dependency-aware RABRBENCH partitions for the active SC/EC/LC population. Counts are cases; split units identify the groups kept within one partition.

Package	Train	Dev	Test	Split unit
SC	494	64	64	Family
EC	606	197	209	Aggregate history
LC	1,684	244	483	Source family

Construction follows two routes with different evidentiary weight. Route A contains 172 cards whose target moment was discovered in source dialogue, sealed, and independently reviewed before an action was assigned. Route B contains 2,239 cards with a controlled target and delayed-evidence turn inserted into an otherwise source-preserved transcript window. Source text remains unchanged and generated units are explicitly marked. Only eight cards retain a naturally occurring later-evidence turn; LC is therefore a controlled long-context revision resource, not a prevalence sample of naturally occurring retrospective disclosure.

Every card has construction-time initial and final labels. The reference action is keep exactly when those labels agree and revise otherwise, so it is a coarser description of the label pair. The evaluation-gold overlay contains the 707 development and test cases that passed two independent machine reviews, including 467 of 483 test cases. The project owner additionally conducted a sample-based human review of the evaluation cases. The training split remains silver because it did not receive the second machine review.

C.5 DEPENDENCY-AWARE SPLITTING

SC keeps the two sibling observations from each family in one partition. EC assigns each complete aggregate history to a single partition: three histories to training and one each to development and test. Some source-overlap identifiers occur in more than one EC history, so its partitions are not strictly component-disjoint. LC assigns 39/12/13 source families to training/development/test, yielding 1,684/244/483 cards. These rules keep declared siblings and source families together. All LC cards also share one construction run, which remains a separate generator-level dependency.

Figure 3 visualizes the same dependency-aware partitions. The logarithmic axis keeps LC's larger controlled package from obscuring the smaller SC and EC splits; the printed values retain the package-specific case units defined in Table 9.

C.6 DATASET ANALYSIS

The three packages share an affect inventory but use different supervision units. SC and EC evaluate retrieval demand and may contain no semantic target, multiple targets, mixed affect, or unresolved references. Each LC card contains one historical target and one construction-time initial–final label pair. Table 11 reports reference coverage in the unit of each package before the label distributions are compared.

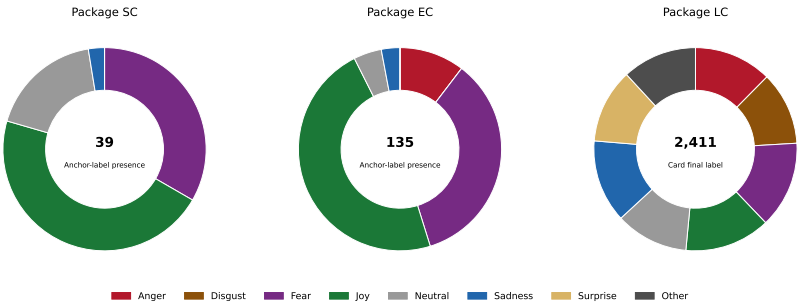

Figure 4: Affect composition across RABRBENCH as three aligned donut charts with one shared label mapping. SC and EC report within-package percentages of repeat-collapsed atomic label presences, with totals 39 and 135; LC reports the 2,411 card-level final labels. The charts preserve their different supervision units.

Figure 4 shows affect composition in three aligned charts with one label mapping. SC and EC count atomic label presences after repeated construction views are collapsed; a mixed state contributes once to each constituent label. LC counts card-level final labels, one per card. The percentages are calculated within each package's own unit.

Table 12 gives the exact counts underlying the three charts, retains zero-count inventory classes, and states the unit used by each package so that the visual percentages remain auditable.

The reviewed SC and EC references are dominated by *fear* and *joy* and contain no *disgust* or *surprise*. Across their held-out sets, 123 units have no target and 12 references remain unresolved. LC covers all eight labels, with full-package counts from 280 to 333. Its 1,245 keep and 1,166 revise cards are close in number by construction. Because the action is derived from whether the initial and final labels match, it adds no independent supervision beyond that label pair.

LC provenance and shortcut diagnostics. LC preserves source dialogue as background, while the later-evidence turn is model-generated in 2,403 of 2,411 cards and naturally occurring in eight. Evaluation-gold comprises 707 development and test cards that passed two independent machine reviews, including 467 of 483 test cards; the project owner additionally reviewed an evaluation sample. In the test split, the delayed-evidence turn contains a lexical marker of the final label in 86.3% of keep and 88.5% of revise cases. Such a cue could support final-label prediction without correctly locating the historical target, so its prevalence describes a potential shortcut rather than measured method behavior. Separately, a Jaccard-overlap retriever ranks the correct historical target first in 57.8% of cases and in the top three in 78.1%; this is a target-localization diagnostic.

Appendix I relates these cue and localization diagnostics to the evaluated history-access comparisons.

Table 11: Reviewed held-out supervision coverage for the active SC/EC/LC population. Groups are construction anchors for SC and EC and source families for LC. Counts retain each package's own supervision unit.

Package	Held-out units	Groups	No target	Scorable units	Scorable references	Unresolved references
SC	64	32	21	38	38	5
EC	209	1	102	100	100	7
LC	727	25	0	707	707	20

Table 10: The frozen evaluation population, and the denominator each score is computed over, for the active SC/EC/LC benchmark. Panel (a) counts cases; panel (b) counts the units each metric is averaged over.

(a) Population

Package	Candidates	Assigned	Families	Excluded from the run
SC	64	36	19	28 (no shared trigger output)
EC	209	209	1	0
LC	483	483	13	0
Total	756	728	–	28

(b) Score denominators

Package	Trigger units scored	Trajectories with a trigger score	Revision targets	Trajectories with a revision target	Joint-scored trajectories
SC	36	36	22	22	35
EC	107	107	100	100	100
LC	8,218	483	467	467	467

Notes. A *trigger unit* is one annotated event in SC/EC and one scored dialogue arrival in LC (Appendix D). A *revision target* has a strict final-affect reference and one declared KEEP/REVISE action. *Joint-scored trajectories* have both a trigger score and a strict final-affect reference. Assigned trajectories without a revision target number 14 SC, 109 EC, and 16 LC; 102 of the EC cases are no-target controls. Scores exclude the 28 unassigned SC candidates.

Table 12: Counts underlying the composition panel of Figure 4. SC and EC counts are atomic label presences after collapsing repeated construction views; LC counts are single-label cards. Zero-count inventory labels remain explicit.

Package	Unit	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise	Other	Total
SC	Anchor-label presence	0	0	13	18	7	1	0	0	39
EC	Anchor-label presence	14	0	47	64	6	4	0	0	135
LC	Card final label	299	281	333	329	280	318	285	286	2,411

D THE LC SEQUENTIAL TRIGGER POPULATION

LC trigger labels are structural sequence labels over the full sequential dialogue population. Each trajectory's historical memory store initializes the available context before sequential evaluation; its turns carry no trigger decisions and are excluded from scoring. Every subsequent dialogue arrival is scored. The single source-declared delayed-evidence boundary is labelled as a request-memory action, with the annotated evidence turn as its basis. Every other arrival is labelled as a stop action with an empty basis.

The current-only trigger receives the visible dialogue prefix in source order at each boundary, without the historical initialization store. The prefix distinguishes arrivals within a trajectory; serializing the same whole-window input at every arrival would instead assign different actions to identical inputs. The initialized history remains available to downstream methods when a request is made.

The reconstruction covers all three frozen splits: 2,411 trajectories and 41,031 scored arrivals. The held-out test split contains the same 483 LC trajectory IDs used in the method comparison, expanded into 8,218 scored arrivals (Table 13).

Each request boundary has exactly one required evidence unit. The reconstructed population has no conflicting actions for identical serialized inputs and no malformed or unscorable arrivals. The stop population consists of all other non-initialization arrivals, without negative sampling or manual selection. Because there is only one request per trajectory, request recall and stop specificity are reported alongside the aggregate retrieval scores.

The frozen trigger was run once on each reconstructed dev and test input (12,324 calls: 4,106 dev and 8,218 test). Table 14 reports test results and test-only cost.

The per-arrival pass evaluates Trigger decisions across the LC dialogue sequence. The RABR (Retrieved history) downstream run replays each trajectory in source order and uses its retained shared Trigger decision at the designated current-event boundary. The 8,218 scored arrivals define the Trigger evaluation population; the downstream retrieval ledger in Appendix G has a different execution unit.

Table 13: Reconstructed LC sequential trigger population.

Item	Count
Trajectories	2,411
Source families	64
Initialization turns excluded	112,907
Non-initialization arrivals scored	41,031
REQUESTMEMORY boundaries	2,411
STOP boundaries	38,620
Malformed or unscorable	0
Duplicate-input action conflicts	0
Train trajectories / arrivals	1,684 / 28,707
Dev trajectories / arrivals	244 / 4,106
Test trajectories / arrivals	483 / 8,218

Table 14: Frozen trigger results and cost on the reconstructed LC test split (483 trajectories; 8,218 scored arrivals).

Measure	Value
Scored arrivals	8,218
RDA	0.838
RCO (request boundaries)	0.974
REQUESTMEMORY recall	0.998
STOP specificity	0.626
Balanced action accuracy	0.812
Action accuracy (event-level)	0.648
Malformed outputs	0
Physical model calls	8,218
Input tokens	2,876,928
Output tokens	64,061
Service seconds	1,989.27

Notes. RDA and RCO average event scores within trajectories and then source families. Recall, specificity, and action accuracy use test boundary counts; balanced action accuracy is the mean of recall and specificity. Cost is for the test split only.

E EVALUATION METRICS

Each metric uses a fixed subset with the required reference. A family links trajectories sharing source material or a construction anchor. Retrieval metrics and joint give each eligible family equal weight within SC, EC, or LC; affect and revision agreement are proportions over their respective fixed trajectory denominators. Unavailable predictions remain in those denominators and score zero. Scores are displayed to three decimal places, while relative improvements use unrounded values.

All six methods in Table 3 are evaluated on the same assigned held-out trajectory IDs within each package: 36 SC, 209 EC, and 483 LC. For each metric, its reference-eligible subset is also fixed across methods (Table 10); the trigger and final-affect components may use different units or execution passes without changing the underlying test trajectories.

Retrieval decision accuracy and retrieval-cue overlap. Retrieval decision accuracy (RDA) is one for a valid action matching the reference and zero otherwise. Retrieval-cue overlap (RCO) is

scored only at reference request boundaries. Let B_{it} and R_{it} denote the predicted and annotated sets of current dialogue turns justifying retrieval for trajectory i at step t. A valid, correct request receives $|B_{it} \cap R_{it}|/|B_{it} \cup R_{it}|$; a missed request, malformed output, empty predicted cue set, or cue outside the visible prefix receives zero. These cues identify current turns used to construct the query; the retriever subsequently returns historical records.

SC and EC compute RDA and RCO on their 36 and 107 target-bound test events, respectively. Separate no-target controls (21 SC and 102 EC) supply stop specificity; they do not enter those request metrics. LC scores RDA on all 8,218 non-initialization test arrivals and RCO on its 483 request boundaries. Each retrieval metric averages eligible event scores within a trajectory, then within a family, then equally across eligible families. Request recall is the fraction of reference requests predicted as requests; Stop specificity is the fraction of reference stops predicted as stops. Balanced action accuracy is their mean. The LC event-level action accuracy in Table 14 instead weights all arrivals equally.

Affect and Joint. For each trajectory with a strict final reference, A_i is one when the final affective label set for the historical target exactly matches that reference and zero otherwise. Affect is the mean of A_i over its fixed evaluation subset. Joint measures retrieval-weighted final-affect correctness. Its per-step retrieval score is

$$g_{it} = \begin{cases} 1, & \text{valid, correct non-retrieval decision,} \\ \dfrac{|B_{it} \cap R_{it}|}{|B_{it} \cup R_{it}|}, & \text{valid, correct retrieval decision,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

A valid retrieval request has a nonempty cue set contained in the visible prefix. Let $\mathcal{T}_i$ contain the evaluated observations for trajectory i: annotated observations in SC/EC and scored dialogue arrivals after initialization in LC. We compute

$$G_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} g_{it}, \qquad J_i = G_i A_i. \tag{9}$$

Let $\mathcal{I}_f$ contain the trajectories in family f with both an evaluated retrieval score and a strict final reference, and let $\mathcal{F}$ contain families with nonempty $\mathcal{I}_f$. Within each package,

$$\text{Joint} = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \frac{1}{|\mathcal{I}_f|} \sum_{i \in \mathcal{I}_f} J_i. \tag{10}$$

Thus, a trajectory with correct final affect receives its mean retrieval score; one with incorrect final affect receives zero. Joint permits partial credit and is not the proportion of completely correct trajectories. The product is computed before aggregation, rather than from the separately reported metric averages.

For the five methods sharing the T5Gemma trigger, LC G_i is computed from the per-arrival Trigger evaluation in Appendix D, and A_i is computed from the corresponding method's downstream trajectory result. These components are matched by frozen trajectory identity before multiplication. RABR (tuned) instead computes both components from its own sequential LC execution. Joint has the same trajectory-level scoring definition for all six methods, with these method-specific execution scopes.

Revision. Where the source declares a single keep/revise action, the realized method action is keep when its final judged label set for the target equals the set it held at initialization and revise otherwise. Agreement is the share of those trajectories whose realized action matches the declared one, computed per package on a denominator fixed across methods, with a method that expresses no action scoring zero. Cases whose reference admits both actions, or declares none, lie outside the metric.

F LC Revision-Conditioned Affect Analysis

The LC test split contains 483 assigned trajectories, of which 467 have strict singleton final-affect references and a source-declared revision action. The latter partition into 246 keep and 221 revise

cases. The action agrees case by case with whether the reference final label equals the initial label. Table 15 expands the revise column of Table 4 by the reference final affective category. Each entry is the proportion of exact final-label matches among all cases in its row; method failures and unavailable predictions remain incorrect. Thus the denominator does not shrink to the cases on which a method chose to revise. The category macro-average weights the eight final-affect categories equally, whereas the micro total weights trajectories equally. Cases share 13 LC source families, so the breakdown is descriptive.

Table 15: Exact final-affect accuracy on LC test cases requiring revision, grouped by reference final-affect category. Scores use three decimal places.

Final-affect category	Cases	RABR (Full history)	RABR (Retrieved history)	RABR (tuned)
ANGER	28	0.857	0.607	0.679
DISGUST	25	0.640	0.640	0.800
FEAR	33	0.667	0.576	0.727
JOY	24	0.583	0.667	0.417
NEUTRAL	24	0.875	0.542	0.542
OTHER	26	0.385	0.346	0.000
SADNESS	33	0.939	0.758	0.879
SURPRISE	28	0.857	0.643	0.929
Micro total	221	0.733	0.602	0.638
Category macro	8	0.725	0.597	0.621

G RESULT SCOPE AND EXECUTION FAILURES

Table 3 compares four baselines (Fixed Initial, Full-history Reclassification, Mem0, and RABR (tuned)) with two RABR variants. The three original baselines and both variants share frozen T5Gemma trigger decisions; RABR (tuned) uses its own learned trigger. The two RABR variants share the deterministic updater and package-specific score denominators. At a request, RABR (Full history) reads all eligible history, whereas RABR (Retrieved history) reads at most the three highest-ranked initialization-history items returned by BM25.

RABR (Retrieved history) assigned 36 SC, 209 EC, and 483 LC trajectories. All 728 reached a terminal state. A continuation completed 114 of the 115 previously failed LC trajectories, leaving one failed LC trajectory and one failed SC trajectory; EC had no failures. The 114 replacements were joined by frozen trajectory ID, with the reference population unchanged. Failures remain in the fixed scoring population and receive zero when a strict final reference exists. The combined BM25 ledger records 604 executed retrievals using the frozen compiled queries; every returned item belongs to initialization history and each retrieval returns at most three items.

The comparison ledger uses the retained per-trajectory Trigger cache at the designated current-event boundary. Its 604 retrievals comprise 490 operations in the original corrected-method terminals and 114 in the successful LC continuation terminals. Provider-level retries were disabled; the continuation was a separate execution of previously failed LC trajectories. Its 114 outcomes replace the corresponding failed terminals by frozen trajectory ID, so each trajectory contributes at most one selected terminal to the reported ledger. The 604 operations count executed BM25 retrievals in this comparison, whereas Appendix D counts LC Trigger decisions across all scored arrivals.

The token column in Table 3 reports means per completed downstream call, excluding failed calls rather than imputing zero. Input tokens are provider prompt tokens. Output tokens for Full-history Reclassification and Mem0 are provider completions for the final readout; Mem0's native memory-service usage is unavailable. For the two Qwen-based RABR variants, output tokens instead measure the accepted `interpretations_by_target[target]` array: compact sorted-key JSON tokenized with the pinned Qwen tokenizer without special tokens. This derived payload count excludes discarded output and the rest of the updater state. RABR (Full history) uses a 10% completed-call sample (3 SC, 15 EC, 48 LC) for that estimate; RABR (Retrieved history) uses all 34 SC, 89 EC, and 481 LC completed calls. The two output-token definitions do not measure the same generation boundary.

The downstream-call means describe the comparison ledger's completed calls; Appendix D separately reports the per-arrival LC Trigger inference cost.

H Analytical Properties of the Reference Updater

The following properties concern the deterministic updater defined in Sections 3.4–3.6. They condition on fixed target links, admitted applications, dependency assignments, and support scores. They characterize aggregation and state transitions, rather than the semantic correctness of the assigned scores, and do not apply automatically to the learned updater in RABR (tuned).

H.1 Evidence Reconstruction and Retraction

For a fixed target h, let $\mathcal{E} = \{E_g\}_{g \in G}$ contain its active, unique applications partitioned into nonempty dependency groups. With $n_g = |E_g|$, Equation 3 can be written as

$$\boldsymbol{\mu}_g = \frac{1}{n_g} \sum_{e \in E_g} \mathbf{s}_{e,h}, \qquad F_h(\mathcal{E}) = \mathrm{softmax}\left(\mathbf{b}_h + \sum_{g \in G} \boldsymbol{\mu}_g \right). \tag{11}$$

Property 1 (support under replay). Reconstructing from the same $\mathcal{E}$ returns the same $F_h(\mathcal{E})$. Permuting the order of the group sum or presenting an incorporated application as a noncontributing duplicate alias leaves the mathematical support unchanged.

Derivation. Each group mean depends only on its fixed applications and scores. An existing identifier or duplicate alias adds no new application, so the group means, their sum, and its softmax remain the same. Finite-precision implementations may differ by rounding. This property does not cover changes to the text seen by the reasoning agent before scoring, nor does it make the complete commitment and resolution trajectory order-independent.

Property 2 (support after retraction). Suppose one application e is removed from group g, with all other applications, scores, and group assignments fixed. If $\mathbf{z}_h$ and $\mathbf{z}'_h$ are the logits before and after retraction, then

$$\mathbf{z}'_h - \mathbf{z}_h = \begin{cases} \dfrac{\boldsymbol{\mu}_g - \mathbf{s}_{e,h}}{n_g - 1}, & n_g > 1, \\ -\boldsymbol{\mu}_g, & n_g = 1, \end{cases} \qquad \mathbf{p}'_h = \mathrm{softmax}(\mathbf{z}'_h). \tag{12}$$

Derivation. For $n_g > 1$, the remaining mean is $(n_g \boldsymbol{\mu}_g - \mathbf{s}_{e,h})/(n_g - 1)$; subtracting the old mean yields the first case. For $n_g = 1$, the empty group contributes zero, and all other groups cancel in the difference. Recomputing Equation 3 from the remaining active applications therefore gives this support. Subtracting only $\mathbf{s}_{e,h}/n_g$ would miss the changed group denominator.

Removing all active applications returns $\mathrm{softmax}(\mathbf{b}_h)$, uniform for the reference root. Restoring an earlier evidence configuration restores its support vector, but its commitment and operational resolution also depend on their transition history. Stored earlier belief versions remain unchanged.

H.2 Revision and Availability in the Reference Configuration

Property 3 (zero-threshold revision). For an existing commitment $c \in \mathcal{H}$, setting $\delta = \gamma = 0$ in Equation 4 yields

$$\textsc{Revise} \iff \ell_v \neq \varnothing \ \wedge \ \ell_v \neq c. \tag{13}$$

Derivation. The anchor distance is nonnegative. If ℓ_v uniquely leads and differs from c, then $p^{(v)}_{h,\ell_v} - p^{(v)}_{h,c} > 0$, so both threshold inequalities hold. Conversely, Equation 4 explicitly requires a unique leader different from c. In this reference configuration, revision follows a change in the unique leader of reconstructed support; positive-threshold filtering is outside the evaluated configuration. An exact tie retains the commitment and suspends its use. Initial commitment remains subject to its separate resume conditions.

Operational resolution. With an existing commitment and zero revision thresholds, the resulting commitment equals any unique leader. Equation 5 therefore becomes

$$r_h^{(v)} = \textsc{Active} \quad \Longleftrightarrow \quad \ell_v \neq \varnothing, \quad P_v \geq \alpha_\rho, \quad M_v \geq \beta_\rho. \tag{14}$$

The resume thresholds are no lower than the hold thresholds, so resuming use requires at least as much peak support and margin as maintaining use. For the eight-category reference inventory, a unique leader has $P_v > 1/8$ and $M_v > 0$; the hold values $\alpha_{\text{hold}} = 0.125$ and $\beta_{\text{hold}} = 0$ add no further filter after an active resolution. Initial commitment and resumption still require $P_v \geq 0.31875$ and $M_v \geq 0.15$. These are properties of the stated rules, not claims of calibrated probabilities or correct affect labels.

H.3 Historical-Query Consistency

Property 4 (persistence of as-of answers). Assume immutable belief versions and evidence manifests, append-only storage with version order monotone in transaction time, and incorporation of evidence only after it becomes available. For a fixed information time T, appending updates stored after T leaves $B_h(\tau; T)$ in Equation 6 unchanged.

Derivation. New versions have storage times greater than T and cannot enter $\{v : T_v \leq T\}$. Its selected version and stored contents remain fixed. Thus an as-of query preserves the interpretation recorded by T while a current query can reflect later evidence. The property concerns stored belief history, independent of the semantic correctness of either interpretation.

I Interpreting the Role of Historical Evidence

Task scope. Retrospective affective belief revision concerns an earlier person–episode target. SC and EC annotate whether a current observation warrants consulting eligible history; LC scores final-affect judgments after historical material becomes accessible. This task definition allows some final labels to be inferable from current cues alone. Any assessment of additional historical access must also account for the complete current observation and the initial judgment or target information already supplied to a method.

Affective cues and target grounding. An explicit delayed disclosure can make an affect category easy to recognize while leaving its relation to a historical target to be resolved. Appendix C.6 reports final-label lexical markers in 86.3% of LC test Keep cases and 88.5% of Revise cases. These are cue-presence rates, not current-only prediction accuracies. The Jaccard target-ranking diagnostic measures lexical localization; Affect scores the final label for the referenced target, and Joint combines it with retrieval-decision quality. Neither score separately requires a correct target-ranking output or evidence-attribution trace.

What the existing comparisons establish. The two deterministic RABR variants retain the same complete current observation, shared trigger, updater, and LC test trajectories, while differing in historical access. On the 467 strict LC references, their Affect scores are 0.872 for full history and 0.812 for retrieved history (Table 3). This difference is concentrated in the reference Revise cases (0.733 versus 0.602), rather than Keep cases (0.996 versus 1.000; Table 4). In the separate pooled, matched SC/EC/LC retrieval-depth comparison, Affect is highest at three rather than five items for both evaluated rankers (Table 5).

These results describe retrospective judgments under the evaluated history-access configurations. Measured final-affect accuracy differs between those configurations, particularly on LC cases requiring a different affective category. The existing comparisons do not isolate additional retrieval against a current-observation-only version of the same predictor; the lexical-cue and target-ranking diagnostics remain separate from that question.

J Post-hoc Analysis of Retrospective Revision

This analysis uses retained per-trajectory outputs and the frozen test references, without new model inference. Its primary LC comparison joins the same 467 strict test trajectories by ID for every

method; 246 require KEEP and 221 require REVISE. The gold action partitions the reference cases, while the outcome below remains exact final-affect accuracy. Failed or unavailable final predictions remain incorrect in the fixed denominator. For a comparator B, let G count cases where RABR (Full history) is Affect-correct and B is not, and let L count the reverse; their paired difference is $(G - L)/n$.

J.1 PAIRED LC OUTCOMES

Table 16: Paired exact final-affect outcomes on fixed LC strict-reference subsets. The RABR column is RABR (Full history); both full-history methods receive the same eligible history and share the trigger, but differ in their downstream processing. G and L are casewise Affect gains and losses for RABR relative to the named comparator.

Comparator	Gold action	n	RABR correct	Comparator correct	G	L	Difference
Full-history Reclassification	All	467	407	347	90	30	$+0.128$
	KEEP	246	245	188	57	0	$+0.232$
	REVISE	221	162	159	33	30	$+0.014$
Mem0	All	467	407	338	100	31	$+0.148$
	KEEP	246	245	170	75	0	$+0.305$
	REVISE	221	162	168	25	31	-0.027

The overall LC Affect gain over direct full-history reclassification consists of 57 net KEEP cases and three net REVISE cases. On the latter subset, both methods also exchange many individual successes ($G = 33$, $L = 30$); Mem0 selects six more exact final labels than RABR (Full history). Thus the measured aggregate advantage is chiefly in retaining an already correct historical affect label, while the paired REVISE results indicate different successes rather than a uniform replacement-label advantage. These comparisons describe output behavior under the evaluated pipelines; they do not attribute the differences to an individual component.

J.2 CURRENT-TEXT LEXICAL CUES

We apply the fixed surface-marker list of Appendix C.6 to the delayed-evidence turn of each of the 467 strict trajectories. Checking the complete current-scene dialogue gives the same hit/no-hit membership in this subset; prompt label lists and reference annotations are excluded. Table 17 retains all four gold-action and marker strata. A missing marker does not imply an absence of semantic or paraphrased affect cues.

Table 17: LC strict-reference Affect by gold action and presence of a surface marker for the final label in the delayed-evidence turn. Cells are exact matches out of the fixed stratum size; G/L compares RABR (Full history) with Full-history Reclassification.

Gold action	Marker	n	Reclassification	Mem0	RABR full	G/L
KEEP	Present	213	168	146	213	$45/0$
KEEP	Absent	33	20	24	32	$12/0$
REVISE	Present	195	141	146	140	$28/29$
REVISE	Absent	26	18	22	22	$5/1$

Among the 26 REVISE cases without a matching surface marker anywhere in the current scene, RABR (Full history) gains five and loses one exact final label relative to direct reclassification. This small subset shows that its observed gains are not confined to text that literally contains the final affect name. It does not establish historical necessity: the current scene may convey affect through other words or events, and the methods still differ in how they process their available history.

J.3 SOURCE-FAMILY STABILITY

Table 18 lists paired Affect differences for all 13 LC source families; five AMI families contain 440 of 467 strict cases. A dash denotes an empty action subset. Small-family rates are descriptive, not independent replications.

Table 18: Per-family RABR (Full history) minus Full-history Reclassification Affect differences on fixed LC strict-reference subsets.

Source family	All n	All diff.	Keep n	Keep diff.	Revise n	Revise diff.
`ami-series-es2005`	86	$+0.186$	47	$+0.298$	39	$+0.051$
`ami-series-es2013`	95	$+0.126$	49	$+0.265$	46	-0.022
`ami-series-is1004`	89	$+0.146$	49	$+0.204$	40	$+0.075$
`ami-series-is1008`	95	$+0.074$	48	$+0.125$	47	$+0.021$
`ami-series-ts3010`	75	$+0.133$	41	$+0.268$	34	-0.029
`locomo-conv26`	11	-0.091	5	0.000	6	-0.167
`msc-test-test_10`	2	0.000	1	0.000	1	0.000
`msc-test-test_11`	3	$+0.333$	2	0.000	1	$+1.000$
`msc-test-test_22`	2	$+0.500$	1	$+1.000$	1	0.000
`msc-test-test_23`	2	0.000	1	$+1.000$	1	-1.000
`msc-test-test_31`	2	0.000	0	—	2	0.000
`msc-test-test_44`	3	$+0.333$	1	0.000	2	$+0.500$
`msc-test-test_7`	2	0.000	1	$+1.000$	1	-1.000

Leaving out one family at a time gives overall paired differences from $+0.115$ to $+0.142$, and REVISE-only differences from 0.000 to $+0.023$. The shared LC construction run and unequal family sizes remain part of the evidence scope.

J.4 MATCHED RETRIEVAL-DEPTH OUTCOMES

The retrieval ablation is a separate pooled sample. Within each ranker, $k = 3$ and $k = 5$ pair the same 146 assigned IDs, trigger decisions, and compiled queries. Affect uses the same 119 strict SC/EC/LC references at both depths, and Revision the same 111 strict EC/LC references (Table 19). One SC terminal failed at both depths per ranker and remains incorrect in the Affect denominator, outside the correct-to-wrong count. Retrieval was called on 123 trajectories and omitted on 23 at both depths. All 123 called pairs have matching queries and nested returned source IDs; 116 add exactly two distinct records.

Table 19: Paired outcomes at $k = 3 \rightarrow 5$ within each ranker. Affect and Revision have fixed denominators of 119 and 111. G/L gives casewise gains/losses at $k = 5$; "target at 3" counts Affect losses with the reference target in the actual $k = 3$ results.

Ranker	Affect correct	Affect G/L	Revision correct	Revision G/L	Target at 3
BM25	$80 \rightarrow 77$	$6/9$	$95 \rightarrow 95$	$4/4$	$6/9$
Dense cosine	$88 \rightarrow 82$	$3/9$	$98 \rightarrow 100$	$4/2$	$7/9$

In eight of nine BM25 Affect losses and seven of nine dense-cosine losses, the scored KEEP/REVISE action remains correct at both depths. The reference target was already among the actual $k = 3$ returned IDs in six and seven losses, respectively. At $k = 5$, the explicit final readout is UNKNOWN or empty in eight and seven losses; the other one and two have a concrete wrong label. None is a failed terminal. Thus a correct action can coexist with an unresolved final affect readout, even with the target already retrieved. The traces do not establish which new record, if any, caused the change.

All nine Affect losses per ranker come from `ami-series-es2005`. Excluding it leaves a two-case net Affect gain for BM25 and no net change for dense cosine among 33 strict references. The pooled decrease is concentrated in this family, within the shared construction run.