# Caching for the Future: Scrub Jay Episodic Memory Principles for Agent Memory Systems

Anonymous ACL submission

## Abstract

LLM agents that persist across sessions accumulate stored memories whose validity varies enormously by content type, yet existing memory architectures treat all memories as equally persistent and systematically contaminate retrieved context with outdated facts. We show that per-memory, type-conditioned temporal decay, a property of western scrub jay episodic memory, can be operationalized as an auto-classified coefficient $\pi_i$ in an external LLM-agent memory store, yielding SCRUBJAY-MEM: each memory is encoded as a jointly-bound What–Where–When tuple with an estimated perishability $\pi_i$ and utility horizon $\tau_i$, retrieved by query-adaptive scoring, and revised retroactively at $O(1)$ LLM calls per update. We introduce the *Temporal Generalization Test* (TGT), a benchmark with held-out retention intervals and a Generalization Gap (GenGap) metric. On TGT, ScrubJay-MEM is the only retrieval-based system with substantially positive GenGap ($+0.108$); on MemoryAgentBench EventQA-64k it improves F1 by $+2.66$ over Mem0 and $+3.09$ over Qwen3-Embedding-4B under a llm backbone. A decay ablation collapses GenGap by $5.7\times$, establishing type-conditioned decay as necessary for the result. Gains narrow under stronger backbones and reverse on fact-consolidation tasks, scoping the contribution to temporal reasoning over perishable facts.

## 1 Introduction

A western scrub jay caching food in autumn does something current LLM agent memory systems cannot. Faced with a choice between stored worms and stored peanuts, the bird recovers worms when caches are fresh and peanuts when the worms have rotted (Clayton and Dickinson, 1998; Worsfold et al., 2025). It tracks not only what it cached and where, but also *when*, and uses this to infer perishability. The integrated What–Where–When (WWW) trace it forms is robust to time in a way that LLM agent memory architectures, despite recent sophistication (Packer et al., 2023; Xu et al., 2026; Chhikara et al., 2025), have not yet achieved. Recent reviews of this literature argue the computational implications of WWW memory remain underexplored (Salwiczek et al., 2010).

This matters because LLM agents now operate over horizons measured in weeks. A user's profession may remain valid for years; their meeting room for the morning is stale by afternoon; their current branch name may not survive lunch. As the memory store grows, the share of retrievals that surface outdated facts grows with it (Wu et al., 2025). The right memory at the wrong time is no better than the wrong memory.

The naive fix (discount all memories by age) does not work. Different memory types decay at fundamentally different rates. A uniform decay either discounts stable knowledge too aggressively or retains ephemeral facts too long. A binary short-term / long-term partition covers only two points on a continuum that contains at least four distinguishable rates (§3.2). Worse, perishability is not static: a memory initially classified as stable can become time-sensitive when context shifts. Solving the temporal memory problem therefore requires *per-memory*, *type-conditioned*, and *retroactively revisable* decay: properties no agent memory system currently combines.

The scrub jay's brain provides a template. We propose SCRUBJAY-MEM, an agent memory architecture in which each memory is stored as an Episodic Memory Unit: a jointly-bound tuple of semantic content, task context, timestamp, and an auto-classified perishability coefficient $\pi_i$. Retrieval combines all four signals through query-adaptive weights $[\alpha, \beta, \gamma, \delta]$. A Prospective Memory Buffer pre-loads anticipated memories before task execution, analogous to jays' future-oriented caching (Raby et al., 2007). Retroactive Contextual Integration revises decay parameters when new in-

formation arrives, mirroring how jays update cache values when ecological signals change (Clayton et al., 2001). Each component traces to a cognitive observation rather than an engineering choice.

We evaluate ScrubJay-MEM on two complementary benchmarks. On MemoryAgentBench EventQA-64k (Hu et al., 2025), the system achieves 61.58 F1, the strongest of the systems evaluated, including A-MEM, Mem0, and Contriever. We additionally release the *Temporal Generalization Test* (TGT), a controlled benchmark with held-out retention intervals, designed as the computational analog of Clayton and Dickinson's retention-interval experiment. On TGT, ScrubJay-MEM is the only retrieval-based system with substantially positive Generalization Gap ($+0.108$ vs. $\leq -0.022$ for all flat-retrieval baselines). Ablating type-conditioned decay collapses this gain by $5.7\times$, establishing decay as *necessary* for the GenGap result.

**Contributions.**

- **A biologically grounded memory architecture** (§3) translating the scrub jay's WWW system into computational form, with four mechanisms (EMU, type-conditioned perishability decay, Retroactive Contextual Integration, Prospective Memory Buffer) derived from a single cognitive model.

- **Two formal results** (Proposition 3.5, Proposition 3.6; proofs in Appendix E): per-update boundedness with WHAT-contraction for RCI, and expected sub-linear retrieval cost for the Prospective Memory Buffer.

- **The Temporal Generalization Test (TGT)** (§4.2; construction details in Appendix B), a controlled benchmark with held-out retention intervals, designed to measure whether memory systems generalise temporal decay knowledge to unseen intervals.

- **Type-conditioned decay as the necessary mechanism for TGT generalisation** (§4). Ablating perishability decay collapses GenGap by $5.7\times$ while leaving staleness-detection accuracy intact, evidence that the GenGap gain requires the decay component, even if independent contributions from the remaining three primitives are not isolated here.

## 2 Related Work

**LLM agent memory architectures.** External memory dominates long-horizon agent design. MemGPT (Packer et al., 2023) pages between main and archival stores; Mem0 (Chhikara et al., 2025) separates short- and long-term tiers; A-MEM (Xu et al., 2026) introduces Zettelkasten linking with LLM-driven evolution. LiCoMemory (Huang et al., 2025), Zep (Rasmussen et al., 2025), Synapse (Jiang et al., 2026), ReadAgent (Lee et al., 2024), and MemoryBank (Zhong et al., 2024) add graph linking, temporal entities, dynamic relations, gist compression, and consolidation loops respectively. Generative Agents (Park et al., 2023) score by importance, recency, and relevance; Hindsight (Latimer et al., 2025) reinterprets memories textually post-hoc. MemoryAgentBench (Hu et al., 2025) formalises memory evaluation across four competencies; we adopt its EventQA-64k as our primary benchmark. *ScrubJay-MEM departs from this landscape by assigning each memory an auto-classified perishability coefficient $\pi_i$ that governs type-conditioned exponential decay, and by routing retrieval through query-adaptive What–Where–When weights rather than fixed similarity. Temporal validity becomes an inspectable architectural parameter rather than an LLM-derived textual attribute.*

**Temporal decay and forgetting.** The Ebbinghaus curve (Ebbinghaus, 1913; Anderson et al., 2004; Brown et al., 2007) models decay as a global function of time. The Forgetting Transformer (Lin et al., 2025) adds forget gates inside attention; FadeMem (Wei et al., 2026) and MemoryBank (Zhong et al., 2024) apply uniform Ebbinghaus decay to external memory; Oblivion (Rana et al., 2026) and selective-forgetting variants (Ge et al., 2024; Wang et al., 2025; Mu et al., 2023; Zhang et al., 2023) make binary retain/discard decisions. *Our work differs along three axes: decay is **per-memory** rather than global, the rate is **type-conditioned** through an auto-classified $\pi_i$, and the rate is **retroactively revisable** through parameter-space updates that cost $O(1)$ LLM calls regardless of how many memories are affected.*

**Episodic memory and anticipation.** Episodic-memory architectures (Tulving, 1972; Pritzel et al.,

[1]Scrub jay photograph: Linda Tanner, "Western Scrub Jay," via Wikimedia Commons, licensed under CC BY 2.0. Source: https://en.wikipedia.org/wiki/File:Aphelocoma_californica_3.jpg.

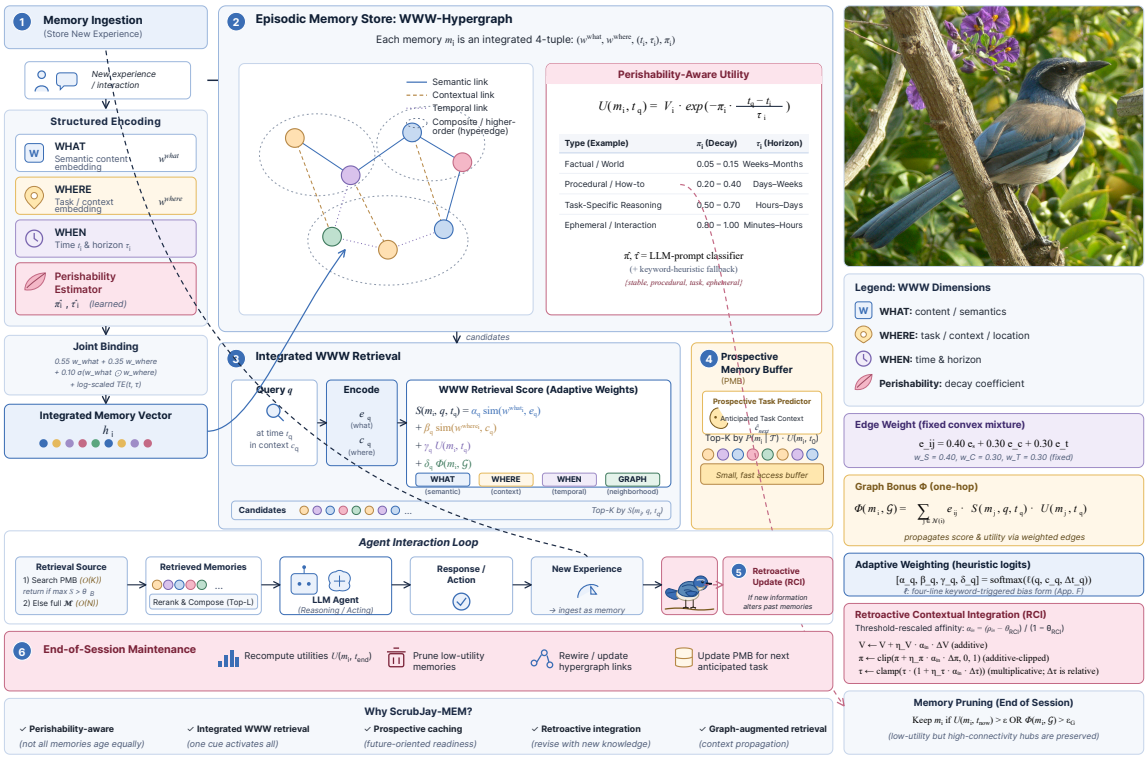


Figure 1: ScrubJay-MEM architecture (blueprint view).[1] The system encodes each experience as an integrated WWW (what–where–when) episodic memory with perishability $\pi$ and horizon $\tau$, organizes memories in a hypergraph, retrieves via a four-factor adaptive score, preloads likely-to-be-needed memories proactively (PMB), and retroactively updates utility and decay when new information arrives (RCI). *The figure depicts the full architectural blueprint with trainable primitives; the implementation evaluated in this paper substitutes parameter-free surrogates at three points (§3.1–3.3): a fixed-weight gated fusion in place of cross-attention binding, an LLM-prompt classifier with keyword fallback in place of the learned perishability MLP, and heuristic keyword-triggered logits in place of the learned adaptive-weighting head.*

2017; Ritter et al., 2021; Graves et al., 2016) predate LLM agents; WWW memory itself has appeared in robotics and virtual agents (Stachowicz and Kruijff, 2011; Brom and Lukavsky, 2009; Crystal, 2010) but not as an LLM-agent memory substrate. Recent behavioural work extends the corvid WWW literature with new controlled analyses (Worsfold et al., 2025). Prospective cognition (Szpunar et al., 2014; Schacter et al., 2007) and prefetching (Smith, 1982; Leviathan et al., 2023; Gao et al., 2023) provide antecedents for our Prospective Memory Buffer. *To our knowledge, ScrubJay-MEM is the first system to operationalise integrated WWW episodic memory specifically for LLM agents, jointly binding content, context, and temporal metadata into a single retrieval-ready embedding. The Prospective Memory Buffer further translates anticipatory caching into an expected sub- linear retrieval mechanism with bounded per-update guarantees (Proposition 3.5, Appendix E).*

## 3 ScrubJay-MEM

We describe ScrubJay-MEM in five parts: the core memory representation (§3.1–3.2), the retrieval mechanism (§3.3–3.4), the two active maintenance processes (§3.5–3.6), the encoding strategy (§3.7), and the pruning rule (§3.8). Figure 1 gives an overview.

### 3.1 Episodic Memory Unit (EMU)

Each memory $m_i$ is an integrated 4-tuple analogous to a single caching episode:

$$m_i = \left( \mathbf{w}_i^{\text{WHAT}},\ \mathbf{w}_i^{\text{WHERE}},\ (t_i,\ \tau_i),\ \pi_i \right), \qquad (1)$$

where $\mathbf{w}_i^{\text{WHAT}} \in \mathbb{R}^{d_1}$ encodes the *semantic content* of the memory (food type → information content), $\mathbf{w}_i^{\text{WHERE}} \in \mathbb{R}^{d_2}$ encodes the *task context* in which it

was created (cache site $\rightarrow$ conversation/task state), $t_i$ is the creation timestamp, $\tau_i \in \mathbb{R}_+$ is the estimated utility horizon, and $\pi_i \in (0, 1]$ is the *perishability coefficient* ($\pi \approx 1$ for ephemeral facts, $\pi \approx 0$ for stable knowledge).

The four fields are combined into a single binding vector $\mathbf{h}_i$ via a deterministic gated fusion that approximates integrated WWW retrieval at parameter-free cost. Let $\mathbf{g}_i = \sigma\big(\mathbf{w}_i^{\text{WHAT}} \odot \mathbf{w}_i^{\text{WHERE}}\big)$ be the elementwise sigmoid of the Hadamard product (a content-context coactivation gate). The binding is a fixed convex mixture

$$\mathbf{h}_i \;=\; 0.55\,\mathbf{w}_i^{\text{WHAT}} + 0.35\,\mathbf{w}_i^{\text{WHERE}} + 0.10\,\mathbf{g}_i, \quad (2)$$

followed by an L2 normalisation. Log-scaled temporal positional offsets $\log(t_i + 1)$ and $\log(\tau_i + 1)$ (each modulo a fixed period) are added to the first two coordinates of $\mathbf{h}_i$ to encode WHEN. The fusion weights and gate are fixed (parameter-free); replacing them with trained cross-attention is left to future work.

### 3.2 Perishability-Aware Utility

The temporal utility of $m_i$ at query time $t_q$ is

$$U(m_i, t_q) = V_i \cdot \exp\Big(-\pi_i \cdot \frac{t_q - t_i}{\tau_i}\Big), \quad (3)$$

with base value $V_i \in \mathbb{R}_+$ (learned). Critically, the decay rate is *type-conditioned*: stable world knowledge ($\pi \approx 0.05$) retains utility over long horizons while ephemeral interaction details ($\pi \approx 0.9$) decay rapidly (Table 3).

Perishability and horizon are estimated at storage time by an LLM-prompt classifier with a deterministic keyword-heuristic fallback. The LLM is asked to assign one of four labels {factual, procedural, task_specific, ephemeral} and to emit a numerical $\pi \in (0, 1]$ and $\tau \in \mathbb{R}_+$ as strict JSON; on parse failure or LLM unavailability the system falls back to a small keyword rule over the memory text and context (e.g., `today/immediate` $\rightarrow$ ephemeral with $\pi = 0.9$; `how to/steps/procedure` $\rightarrow$ procedural with $\pi = 0.3$). Estimates are clipped to $[\tau_{\min}, \tau_{\max}]$ to bound the utility horizon. We refer to this classifier compactly as

$$\hat{\pi}_i,\ \hat{\tau}_i = \phi\big(\text{text}_i,\ \text{ctx}_i,\ \mathbf{w}_i^{\text{WHAT}},\ \mathbf{w}_i^{\text{WHERE}}\big), \quad (4)$$

emphasising that the production of $(\hat{\pi}, \hat{\tau})$ in this v1 system is an LLM call plus a heuristic fallback rather than a trained neural network; the classifier prompt is reproduced verbatim in Appendix F. A trained-MLP variant is left to future work.

### 3.3 Integrated WWW Retrieval

Given query $\mathbf{q}$ at time $t_q$ with task context $\mathbf{c}_q$, the retrieval score for memory $m_i$ is

$$S(m_i, \mathbf{q}, t_q) = \alpha_q \underbrace{\text{sim}\big(\mathbf{w}_i^{\text{WHAT}}, \mathbf{e}_q\big)}_{\textbf{What}} + \beta_q \underbrace{\text{sim}\big(\mathbf{w}_i^{\text{WHERE}}, \mathbf{c}_q\big)}_{\textbf{Where}} + \gamma_q \underbrace{U(m_i, t_q)}_{\textbf{When}} + \delta_q \underbrace{\Phi(m_i, \mathcal{G})}_{\textbf{Graph}}. \quad (5)$$

where $\mathbf{e}_q = \text{Enc}(\mathbf{q})$. The weights $[\alpha_q, \beta_q, \gamma_q, \delta_q]$ are *query-adaptive*, produced by a heuristic scoring function that combines fixed base logits, embedding-norm scales, and keyword-triggered query biases:

$$\mathbf{w}_q = \begin{bmatrix}\alpha_q & \beta_q & \gamma_q & \delta_q\end{bmatrix}^\top = \text{softmax}\begin{bmatrix} 1.5 + b_q^{\text{WHAT}} + 0.20\,\|\mathbf{e}_q\| \\ 1.0 + b_q^{\text{WHERE}} + 0.10\,\|\mathbf{c}_q\| \\ 1.0 + b_q^{\text{WHEN}} + \min\big(2.0, \tfrac{1}{5}\log(\Delta t_q + 1)\big) \\ 1.0 + b_q^{\text{GRAPH}} \end{bmatrix}. \quad (6)$$

where the bias terms $b_q^X$ are keyword-triggered scalars that detect query intent (e.g. *latest/recent/today/stale* contribute $+1.5$ to $b_q^{\text{WHEN}}$; *context/project/session/workspace* contribute $+1.0$ to $b_q^{\text{WHERE}}$; *what/which/fact/detail/remember* contribute $+0.8$ to $b_q^{\text{WHAT}}$; *related/connected/neighbor/link* contribute $+0.8$ to $b_q^{\text{GRAPH}}$). Temporally urgent queries thus up-weight $\gamma$ (When) and context-sensitive queries up-weight $\beta$ (Where). The weighting is parameter-free in this v1 system; a learned weighting head $W_\theta$ that replaces the heuristic is left to future work. This contrasts with A-MEM's fixed-weight keyword/embedding lookup.

### 3.4 Episodic Hypergraph

We maintain a typed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over the memory store with three edge classes: semantic (WHAT–WHAT), contextual (WHERE–WHERE), and temporal co-occurrence (WHEN–WHEN), combined via a fixed convex mixture (in our experiments $w_S = 0.40$, $w_C = 0.30$, $w_T = 0.30$; see Appendix F):

$$e_{ij} = w_S\, e_{ij}^S + w_C\, e_{ij}^C + w_T\, e_{ij}^T, \quad (7)$$

where $e_{ij}^S$ and $e_{ij}^C$ are thresholded cosine similarities between the respective embeddings, and $e_{ij}^T = \exp(-\lambda_T |t_i - t_j|) \mathbb{1}[|t_i - t_j| < \Delta_T]$. The graph bonus for $m_i$ is computed by one-hop message passing:

$$\Phi(m_i, \mathcal{G}) = \sum_{j \in \mathcal{N}(i)} e_{ij}\, S(m_j, \mathbf{q}, t_q)\, U(m_j, t_q), \quad (8)$$

which propagates both semantic relevance *and* temporal utility from neighbors: links to decayed memories are automatically down-weighted. This extends A-MEM's semantic-only Zettelkasten graph.

### 3.5 Retroactive Contextual Integration (RCI)

When new information $\mathbf{n}$ arrives at time $t_n$ (e.g., a user preference change), RCI updates the *latent parameters* of all affected memories without a per-memory LLM call.

**Step 1: Identify affected memories.** Compute the cosine relevance $\rho_{in} = \mathrm{sim}(\mathbf{w}_i^{\text{WHAT}}, \mathbf{w}_n^{\text{WHAT}})$ and gate on the threshold $\theta_{\text{RCI}}$. Memories with $\rho_{in} \le \theta_{\text{RCI}}$ are skipped. For memories above the threshold, we rescale the similarity into a threshold-relative affinity $\alpha_{in} = (\rho_{in} - \theta_{\text{RCI}})/(1 - \theta_{\text{RCI}}) \in (0, 1]$ that drives the magnitude of all subsequent updates.

**Step 2: Update parameters.** A *single* LLM call parses $\mathbf{n}$ into bounded update deltas $(\Delta V(\mathbf{n}),\ \Delta\pi(\mathbf{n}),\ \Delta\tau(\mathbf{n}))$ and a textual summary; $\Delta\tau$ is interpreted as a relative ratio (so $\Delta\tau = +0.2$ denotes a $+20\%$ change in $\tau$). The deltas are applied to each affected memory via lightweight vector operations, modulated by the affinity $\alpha_{in}$ from Step 1:

$$V_i \leftarrow \max\big(0,\ V_i + \eta_V\, \alpha_{in}\, \Delta V(\mathbf{n})\big), \tag{9}$$

$$\pi_i \leftarrow \mathrm{clip}\big(\pi_i + \eta_\pi\, \alpha_{in}\, \Delta\pi(\mathbf{n}),\ 0,\ 1\big), \tag{10}$$

$$\tau_i \leftarrow \mathrm{clip}\Big(\tau_i \cdot \max\big(0.1,\ 1 + \eta_\tau\, \alpha_{in}\, \Delta\tau(\mathbf{n})\big),\ \tau_{\min},\ \tau_{\max}\Big). \tag{11}$$

The $V$ and $\pi$ updates are additive offsets (with $\pi$ clipped to $[0, 1]$); the $\tau$ update is multiplicative because $\Delta\tau$ is a relative ratio. All three are bounded per update by the LLM's fixed delta dynamic range and the affinity gate (Appendix E); the clip on $\tau$ keeps the horizon inside the operating interval $[\tau_{\min}, \tau_{\max}]$ regardless of repeated updates. Learning rates $(\eta_V,\ \eta_\pi,\ \eta_\tau)$ are reported in Appendix F.

**Step 3: Soft semantic fusion.** The WHAT embedding is moved toward the new evidence by a damped step using the same affinity gate, followed by L2 re-normalisation:

$$\mathbf{w}_i^{\text{WHAT}} \leftarrow \mathrm{L2}\Big(\mathbf{w}_i^{\text{WHAT}} + \eta_w\, \alpha_{in}\big(\mathbf{w}_n^{\text{WHAT}} - \mathbf{w}_i^{\text{WHAT}}\big)\Big). \tag{12}$$

Total per-event cost is $O(1)$ LLM calls (for delta parsing) plus $O(|\{i : \rho_{in} > \theta_{\text{RCI}}\}|)$ vector operations, compared with A-MEM's $O(N)$ LLM calls for $N$ affected memories.

[RCI Per-Update Boundedness and WHAT-Contraction] Let the parsed deltas be bounded: $|\Delta V(\mathbf{n})| \le \Delta V_{\max}$, $|\Delta\pi(\mathbf{n})| \le \Delta\pi_{\max}$, and $\Delta\tau(\mathbf{n}) \in [\Delta\tau_{\min}, \Delta\tau_{\max}]$ with $1 + \eta_\tau\, \Delta\tau_{\min} \ge 0.1$ (enforced by the clamp in Eq. 11). Then for any memory $m_i$ and any affinity $\alpha_{in} \in [0, 1]$: **(i)** the $V$-update satisfies $|\Delta V_i| \le \eta_V\, \Delta V_{\max}$; **(ii)** the $\pi$-update satisfies $|\Delta\pi_i| \le \eta_\pi\, \Delta\pi_{\max}$ inside the $[0, 1]$ clip; **(iii)** $\tau_i$ remains in $[\tau_{\min}, \tau_{\max}]$ after clipping; **(iv)** the WHAT-update (12) is, prior to renormalisation, a contraction toward $\mathbf{w}_n^{\text{WHAT}}$ with Lipschitz constant $(1 - \eta_w\, \alpha_{in})$; whenever $\eta_w\, \alpha_{in} < 1$ and $\mathbf{w}_n^{\text{WHAT}}$ is held fixed, repeated application converges geometrically. *Proof.* See Appendix E.

### 3.6 Prospective Memory Buffer (PMB)

Before executing task $\mathcal{T}$, ScrubJay-MEM pre-loads a buffer of $K$ memories most likely to be needed, inspired by the jay's anticipatory caching behavior.

**Anticipatory scoring.**

$$P(m_i \mid \mathcal{T}) = \sigma\big(\mathbf{v}_{\mathcal{T}}^\top \mathbf{w}_i^{\text{WHAT}} + \mathbf{u}_{\mathcal{T}}^\top \mathbf{w}_i^{\text{WHERE}} + b_{\mathcal{T}}\big), \tag{13}$$

where $(\mathbf{v}_{\mathcal{T}}, \mathbf{u}_{\mathcal{T}}, b_{\mathcal{T}})$ are light-weight task-encoder parameters.

**Buffer population.** $\mathcal{B}_{\mathcal{T}} = \text{top-K}\{P(m_i \mid \mathcal{T}) \cdot U(m_i, t_0)\}_{i=1}^{|\mathcal{M}|}$.

**Two-stage retrieval.** At query time the system first searches the buffer ($O(K)$); if the best score exceeds a confidence threshold $\theta_{\mathcal{B}}$, it is returned immediately. Otherwise, a fallback full-store search ($O(N)$) is performed.

[Sub-linear Retrieval] Let $P_{\text{hit}} = \Pr[\text{answer} \in \mathcal{B}_{\mathcal{T}}]$. Then expected retrieval cost is $K + (1 - P_{\text{hit}}) \cdot N$. As the task encoder improves, $P_{\text{hit}} \to 1$ and retrieval becomes $O(K)$.

### 3.7 Future-Oriented Encoding

Following evidence that prospective encoding improves later recall (Szpunar et al., 2014), each new experience $x_i$ is augmented at storage time with a brief prospective annotation:

$$\begin{aligned} \tilde{x}_i &= \big[x_i\,;\ \mathrm{LLM}_\theta(\text{“When useful?”} \,\|\, x_i)\big], \\ \mathbf{w}_i^{\text{WHAT}} &= \mathrm{Enc}(\tilde{x}_i). \end{aligned} \tag{14}$$

This biases the WHAT embedding toward anticipated retrieval scenarios, improving recall on future queries.

### 3.8 Value-Weighted Memory Consolidation

At session boundaries, memories whose utility has decayed below threshold $\epsilon$ are pruned, unless they serve as high-connectivity hubs in $\mathcal{G}$:

$$\mathcal{M}' = \{m_i \in \mathcal{M} : U(m_i, t_{\text{now}}) > \epsilon \text{ or } \Phi(m_i, \mathcal{G}) > \epsilon_G\}. \tag{15}$$

The second condition prevents fragmentation of the episodic hypergraph by retaining low-utility but structurally important nodes.

### 3.9 Full Pipeline Summary

The complete system operates in four phases. **(1) Storage**: new experiences are prospectively annotated (§3.7), encoded into an EMU (§3.1), assigned auto-estimated $(\hat{\pi}, \hat{\tau})$ (§3.2), and integrated into $\mathcal{G}$ (§3.4). **(2) Prospective loading**: the PMB pre-populates a buffer of $K$ anticipated memories for the upcoming task (§3.6). **(3) Retrieval**: incoming queries are scored against the buffer (then full store on miss) via adaptive WWW scoring (§3.3). **(4) Maintenance**: RCI updates affected memory parameters when new information arrives (§3.5); consolidation prunes low-utility memories at session end (§3.8).

## 4 Experiments

### 4.1 Temporal Reasoning on MemoryAgentBench (EventQA-64k)

We evaluate ScrubJay-MEM on the EventQA subset of MemoryAgentBench (Hu et al., 2025), which tests an agent's ability to retrieve and reason over time-ordered event sequences embedded in long narrative passages. EventQA is the most temporally demanding subset of MAB, making it the closest public analog to the kind of task ScrubJay-MEM's perishability mechanism is designed for.

**Setup.** The benchmark consists of 5 long-form contexts (median length 64K tokens) with 100 questions each, for a total of 500 queries. We use MAB's official scorer (`metrics_summarization`) which reports exact-match (EM), token-level F1, substring EM (sEM), and ROUGE-L F1, with no LLM-as-judge component. All systems share the same backbone (`llama3.1:8b`) and embedding model (`nomic-embed-text`, 768-dim), served locally via Ollama for reproducibility and to eliminate provider-specific confounds. Five baselines are evaluated: lexical (BM25), dense (Contriever (Izacard et al., 2022), Qwen3-Embedding-4B (Zhang et al., 2025)), and agentic-memory (Mem0 (Chhikara et al., 2025), A-MEM (Xu et al., 2026)). Numbers are reported with the same top-$k = 5$ retrieval depth and chunk size (4096) across systems.

Table 1: EventQA-64k results with the `llama3.1:8b` backbone. All systems use MAB's official scorer. **Bold**: best in column.

| System | EM | F1 | sEM | R-L |
|---|---|---|---|---|
| BM25 | 33.60 | 55.18 | 36.80 | 56.74 |
| A-MEM (Xu et al., 2026) | 37.00 | 56.46 | 41.20 | 58.20 |
| Contriever (Izacard et al., 2022) | 38.60 | 57.67 | 40.60 | 59.42 |
| Qwen3-Emb.-4B (Zhang et al., 2025) | 36.60 | 58.49 | 39.00 | 60.15 |
| Mem0 (Chhikara et al., 2025) | 36.00 | 58.92 | 37.40 | 60.84 |
| **ScrubJay-MEM** | **41.00** | **61.58** | **42.80** | **63.39** |

**Results.** ScrubJay-MEM achieves the highest F1 (**61.58**) and the highest exact-match (**41.00**) on EventQA-64k, outperforming the strongest agentic baseline (Mem0) by +2.66 F1 / +5.00 EM and the strongest embedding baseline (Qwen3-Embedding-4B) by +3.09 F1 / +4.40 EM (Table 1).

**Sensitivity to backbone capability.** A natural concern is whether ScrubJay-MEM's advantage persists with a stronger generation model. To probe this, we re-ran the full sweep with `qwen3:30b-instruct` (Table 11, appendix). With the stronger backbone, dense-retrieval baselines close the gap considerably (Contriever F1 78.68 vs. ScrubJay-MEM 72.82), suggesting that some of the temporal benefit our architecture confers can be matched by a sufficiently capable LLM reasoning over dense retrievals. We frame this honestly: *ScrubJay-MEM's contribution is largest in the resource-constrained regime that characterizes most deployed memory agents*, where the backbone LLM is not large enough to internally compensate for a temporally-naive retriever.

**Honest scoping: where ScrubJay-MEM does *not* help.** We also evaluated on MAB's Conflict-Resolution subsets (Factconsolidation-MH, Factconsolidation-SH). On these tasks, where factual updates supersede earlier statements but stale facts must remain visible for the consolidation step, our perishability mechanism is actively miscalibrated: type-conditioned decay suppresses the very memories the task requires. We report this in the appendix (Table 12) rather than the main results, but flag it explicitly: ScrubJay-MEM targets *temporal reasoning over perishable facts*, not generic fact retrieval. This delineation is a feature, not a limitation: the next subsection

(Section 4.2) quantifies the architectural property responsible for our win.

### 4.2 Temporal Generalization to Unseen Retention Intervals

EventQA establishes that ScrubJay-MEM helps on temporal benchmarks overall. This subsection isolates *why*: we introduce the Temporal Generalization Test (TGT), a controlled diagnostic that asks whether a memory system can generalize its temporal behavior to retention intervals it has never been calibrated on. The benchmark is designed as the computational analog of the scrub jay retention interval experiment of Clayton et al. (2003), which demonstrated that western scrub jays generalize cache-decay knowledge to intermediate, unseen retention intervals, a property we argue any biologically-plausible memory system should exhibit.

**Benchmark design.** TGT consists of 20 instances, each comprising 96 memories sampled across four perishability classes (Ephemeral, Session-Specific, Durable Preference, Stable Knowledge; 24 per class) and 66 queries issued at five retention intervals: $I_1$ (immediate, 0–1 sessions), $I_2$ (short, 3–5), $I_3$ (medium, 8–12, **unseen**), $I_4$ (long, 20–25), and $I_5$ (very long, 40–60, **unseen**). $I_3$ tests interpolation to an unseen interval between two seen ones; $I_5$ tests extrapolation beyond the calibration range. The full benchmark contains 1,320 queries.

**Metric: Generalization Gap.** We define the *Generalization Gap* as

$$\mathrm{GenGap} \;=\; \mathrm{Acc}(I_2) + \mathrm{Acc}(I_4) - 2 \cdot \mathrm{Acc}(I_3), \tag{16}$$

which measures how far the system's accuracy at the unseen interpolation interval falls below the linear interpolation of its accuracy at the two adjacent seen intervals. A system with continuous temporal modeling should have $\mathrm{GenGap} \approx 0$; a system that has memorized behavior only at the seen intervals will have $\mathrm{GenGap} \ll 0$. Positive GenGap indicates *robust* generalization—accuracy at the unseen interval at or above the seen-interval baseline. We additionally report the Temporal Generalization Score $\mathrm{TGS} = \frac{1}{5}\sum_k \mathrm{Acc}(I_k)$, and Combined / Factual / Staleness accuracy at the per-query level (definitions in Section B.7).

**Baselines.** We compare against a stratified set of nine baselines spanning the spectrum of memory designs: a random control (B0), an LLM-only condition with no retrieval (B1), lexical retrieval (B2: BM25), dense retrieval (B3: `nomic-embed-text` top-1), hybrid retrieval (B4: BM25 + dense via reciprocal rank fusion), a recency-prior baseline (B5: dense + global exponential recency), vanilla RAG (B6: dense retrieval + LLM answer generation), and RAG with explicit temporal prompting (B7: B6 with session-age annotations and type-perishability instructions in the prompt). The B5 baseline is included specifically to test whether a single global decay rate matches type-conditioned perishability; B7 tests whether a strong LLM with explicit age annotations can match an explicit decay model.

**Setup.** All TGT systems share `qwen3:30b-instruct` as the generation model (where applicable) and `nomic-embed-text` (768-dim) as the embedder; retrieval depth is top-$k = 10$ and we use a single seed (42). The benchmark itself is naturalised at construction time by `qwen3:30b-instruct` (Appendix B); systems consume the released benchmark unchanged. To avoid self-preference bias (Panickssery et al., 2024), we use `llama3.2:3b` as the answer-correctness judge, a different model family (Meta) from the Alibaba-trained generation backbone, chosen for cross-family separation rather than capacity. Each query is scored on (i) factual correctness against ground-truth memory content and (ii) temporal validity (staleness) judgment; combined accuracy requires both.

**Results.** Table 2 shows the full E3 TGT results.

We organize the discussion around three findings.

**Finding 1: ScrubJay-MEM is the only retrieval-based system that generalizes to unseen intervals.** Among all retrieval-based methods, only ScrubJay-MEM (full) achieves substantially positive GenGap ($+0.108$). Every flat-retrieval baseline (lexical, dense, hybrid, even hybrid with a recency prior) has *negative* GenGap, meaning their accuracy at the unseen $I_3$ interval is below the linear interpolation of the seen $I_2$ and $I_4$ accuracies. The next-best retrieval system on GenGap (hybrid RRF) sits at $-0.054$, giving ScrubJay-MEM a **16.2-percentage-point margin** on this metric.

**Finding 2: Type-conditioned decay, not flat recency, is the mechanism.** The decay ablation (ScrubJay-MEM (no decay)) collapses GenGap from $+0.108$ to $+0.019$ (a $5.7\times$ reduction)

Table 2: Temporal Generalization Test results. 20 instances, 1,320 queries, `qwen3:30b-instruct` generation + `llama3.2:3b` judge (cross-family). **Bold**: best in column among retrieval-based systems (excluding LLM-only and trivial controls). GenGap > 0 indicates robust generalization to the unseen interpolation interval. Tier categorises systems by per-query LLM usage: *retrieval-based* systems (including ours) produce the answer as the top-1 retrieved memory text without invoking the generator at query time; *RAG* systems retrieve then generate via the LLM. ScrubJay-MEM uses LLM calls at ingest and on RCI events (§3.2, 3.5, 3.7) but not at query time.

| System | Tier | Combined ↑ | Factual ↑ | Staleness ↑ | TGS ↑ | GenGap ↑ |
|---|---|---|---|---|---|---|
| *LLM-based and trivial controls* | | | | | | |
| Random (B0) | Trivial | 28.6 | 29.3 | 84.1 | 0.296 | +0.071 |
| LLM-only (B1) | LLM-only | 44.4 | 44.4 | 95.2 | 0.459 | +0.181 |
| RAG-vanilla (B6) | RAG | 62.7 | 67.1 | 91.5 | 0.633 | +0.039 |
| RAG-temporal (B7) | RAG | 67.2 | 71.5 | 90.7 | 0.678 | −0.018 |
| *Retrieval-based systems* | | | | | | |
| BM25 (B2) | Lexical | 35.2 | 40.2 | 77.7 | 0.353 | +0.039 |
| Dense top-1 (B3) | Dense | 37.0 | 41.7 | 73.6 | 0.372 | −0.022 |
| BM25⊕Dense RRF (B4) | Hybrid | **39.4** | **45.0** | 74.6 | **0.395** | −0.054 |
| Dense + recency (B5) | Temporal | 37.5 | 41.9 | 74.5 | 0.377 | −0.046 |
| ScrubJay-MEM (no decay) | Ours-abl. | 35.8 | 39.1 | 82.3 | 0.366 | +0.019 |
| **ScrubJay-MEM (full)** | **Ours** | 37.1 | 39.8 | **82.3** | 0.378 | **+0.108** |

while staleness accuracy is preserved (both variants at 82.3%). The flat-recency baseline (B5, GenGap −0.046) performs *worse* than the no-temporal-model baseline (B3, GenGap −0.022), directly refuting the conjecture that a single global decay rate is sufficient. Naive recency penalizes durable memories along with ephemeral ones; type-conditioned perishability decays only what should decay (Figure 2).

**Finding 3: LLM-based methods achieve higher combined accuracy but via a different mechanism.** RAG-temporal (B7) achieves the highest combined accuracy on E3 (67.2% vs. our 37.1%), driven by the LLM's ability to reason over session-age annotations at inference time. We do not contest this result; we contextualize it. B7's GenGap is −0.018, indicating that its temporal competence does not generalize as a learned property but must be re-derived at each query through prompt-level reasoning, at the cost of one LLM call per query. ScrubJay-MEM encodes temporal validity as architectural parameters $(\pi, \tau)$, enabling sub-linear retrieval (via the Prospective Memory Buffer, Section 3.6) and an inspectable decay curve. We view the two approaches as complementary rather than competing: prompting captures temporal reasoning at inference cost; perishability captures it at architectural cost. The two regimes occupy distinct Pareto frontiers in the joint (combined, staleness) plane (Figure 4): ScrubJay-MEM sits at the top of the retrieval frontier, matching LLM-grade staleness judgment without a per-query LLM call.

**A note on the LLM-only baseline.** LLM-ONLY (B1) has the highest GenGap of all systems (+0.181) but only 44.4% factual accuracy. This is an artifact of low-accuracy systems exhibiting no systematic per-interval bias: when answers are near-random with respect to memory content, they are also near-random with respect to retention interval, giving artificially positive GenGap. Among retrieval-grounded systems with factual accuracy above 35%, ScrubJay-MEM's GenGap is the highest by a substantial margin.

## 5 Conclusion

ScrubJay-MEM translates four mechanisms of western scrub jay episodic memory—integrated WWW encoding, a per-memory perishability coefficient $\pi_i$, retroactive contextual integration at $O(1)$ LLM calls per update, and a Prospective Memory Buffer—into an inspectable architecture for LLM-agent memory. Under `llama3.1:8b`, the system improves EventQA-64k F1 by +2.66 over Mem0 and +3.09 over Qwen3-Embedding-4B; on the Temporal Generalization Test introduced here, it is the only retrieval-based system with substantially positive GenGap (+0.108), and a NO-DECAY ablation collapses this gain by 5.7×, isolating type-conditioned decay as the responsible mechanism and operationalising a comparative-cognition primitive as an inspectable architectural parameter of LLM-agent memory.

## Limitations

**Scope.** Our gains concentrate on temporal reasoning over perishable facts. Under a stronger backbone (Section C), dense retrieval matches us on EventQA-64k, and on MemoryAgentBench Conflict-Resolution (Section D) flat retrieval is better-suited than type-conditioned decay; fact-consolidation requires that stale facts remain visible. ScrubJay-MEM is therefore best understood as a temporally-perishable fact retriever, complementary to flat retrievers for consolidation tasks and to prompt-level temporal reasoning for systems with abundant backbone capacity.

**TGT is a controlled diagnostic.** TGT is LLM-naturalised from deterministic templates with held-out retention intervals and a known validity schedule, properties that enable a mechanistic claim about generalisation but limit external validity to in-distribution conversational logs. TGT's four perishability classes share $\pi$-ranges with our internal taxonomy (Tables 3 and 6); the no-decay ablation, which collapses GenGap by $5.7\times$, controls for the concern that the benchmark and the architecture share categorical primitives. Reported numbers use `qwen3:30b-instruct` for generation and a cross-family `llama3.2:3b` judge to mitigate self-preference bias (Panickssery et al., 2024); the qualitative ranking we report is robust to per-query verdicts.

## Ethical Considerations

**Synthetic data, no real subjects.** TGT contains no real-user data: memory text is composed from a controlled vocabulary of fictional names and slot values (Appendix B.6) and naturalised by an LLM under a leakage-prevention protocol (Appendix B.5). Biological motivations cite prior captive-animal behavioural studies; we ran no animal experiments of our own.

**Misuse: silent forgetting of sensitive content.** ScrubJay-MEM selectively forgets information based on an auto-classified perishability coefficient $\pi_i$. A production deployment could silently drop user-relevant medical, financial, or safety-critical content mislabelled as ephemeral. We recommend treating $\pi_i$ as advisory and gating deletion of any sensitive memory class behind an explicit retention policy rather than the classifier alone.

## References


John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological review*, 111(4):1036.

Cyril Brom and J Lukavsky. 2009. Towards virtual characters with a full episodic memory ii: The episodic memory strikes back. In *Proc. empathic agents, AAMAS workshop*, pages 1–9.

Gordon DA Brown, Ian Neath, and Nick Chater. 2007. A temporal ratio model of memory. *Psychological review*, 114(3):539.

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

Nicola S Clayton, Timothy J Bussey, and Anthony Dickinson. 2003. Can animals recall the past and plan for the future? *Nature Reviews Neuroscience*, 4(8):685–691.

Nicola S Clayton and Anthony Dickinson. 1998. Episodic-like memory during cache recovery by scrub jays. *Nature*, 395(6699):272–274.

Nicola S Clayton, Kara Shirley Yu, and Anthony Dickinson. 2001. Scrub jays (aphelocoma coerulescens) form integrated memories of the multiple features of caching episodes. *Journal of Experimental Psychology: Animal Behavior Processes*, 27(1):17.

Sérgio PC Correia, Anthony Dickinson, and Nicola S Clayton. 2007. Western scrub-jays anticipate future needs independently of their current motivational state. *Current Biology*, 17(10):856–861.

Jonathon D Crystal. 2010. Episodic-like memory in animals. *Behavioural Brain Research*, 215(2):235–243.

Hermann Ebbinghaus. 1913. *Memory*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model tells you what to discard: Adaptive kv cache compression for llms. In *International Conference on Learning Representations*, volume 2024, pages 22975–22988.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, and 1 others. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Hanqi Jiang, Junhao Chen, Yi Pan, Ling Chen, Weihang You, Yifan Zhou, Ruidong Zhang, Andrea Sikora, Lin Zhao, Yohannes Abate, and 1 others. 2026. Synapse: Empowering llm agents with episodic-semantic memory via spreading activation. *arXiv preprint arXiv:2601.02744*.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. 2025. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *International Conference on Machine Learning*, pages 26396–26415. PMLR.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Zhixuan Lin, Evgenii Nikishin, Xu He, and Aaron Courville. 2025. Forgetting transformer: Softmax attention with a forget gate. In *International Conference on Learning Representations*, volume 2025, pages 69704–69738.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. 2023. Memgpt: towards llms as operating systems.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

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

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. 2017. Neural episodic control. In *International conference on machine learning*, pages 2827–2836. PMLR.

Caroline R Raby, Dean M Alexis, Anthony Dickinson, and Nicola S Clayton. 2007. Planning for the future by western scrub-jays. *Nature*, 445(7130):919–921.

Ashish Rana, Chia-Chien Hung, Qumeng Sun, Julian Martin Kunkel, and Carolin Lawrence. 2026. Oblivion: Self-adaptive agentic memory control through decay-driven activation. *arXiv preprint arXiv:2604.00131*.

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

Samuel Ritter, Ryan Faulkner, Laurent Sartran, Adam Santoro, Matthew Botvinick, and David Raposo. 2021. Rapid task-solving in novel environments. In *International Conference on Learning Representations*.

Lucie H Salwiczek, Arii Watanabe, and Nicola S Clayton. 2010. Ten years of research into avian models of episodic-like memory and its implications for developmental and comparative cognition. *Behavioural Brain Research*, 215(2):221–234.

Daniel L Schacter, Donna Rose Addis, and Randy L Buckner. 2007. Remembering the past to imagine the future: the prospective brain. *Nature reviews neuroscience*, 8(9):657–661.

Alan Jay Smith. 1982. Cache memories. *ACM Computing Surveys (CSUR)*, 14(3):473–530.

Dennis Stachowicz and Geert-Jan M Kruijff. 2011. Episodic-like memory for cognitive robots. *IEEE Transactions on Autonomous Mental Development*, 4(1):1–16.

Karl K Szpunar, R Nathan Spreng, and Daniel L Schacter. 2014. A taxonomy of prospection: Introducing an organizational framework for future-oriented cognition. *Proceedings of the National Academy of Sciences*, 111(52):18414–18421.

Endel Tulving. 1972. Episodic and semantic memory. *Organization of memory*, 1(381-403):1.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2025. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Lei Wei, Xiao Peng, Xu Dong, Niantao Xie, and Bin Wang. 2026. Fademem: Biologically-inspired forgetting for efficient agent memory. *Preprint*, arXiv:2601.18642.

Ella Worsfold, Nicola S Clayton, and Lucy G Cheke. 2025. Revisiting episodic-like memory in scrub jays: Is there more we can still learn from what–where–when caching behaviour? *Learning & Behavior*, 53(1):65–79.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*.

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

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

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

# A Biological Background: Episodic Memory in Western Scrub Jays

Western scrub jays (*Aphelocoma californica*) are among the few non-human species that exhibit *episodic-like* memory: the ability to recall the *what*, *where*, and *when* of a past event as an integrated representation rather than three independent features (Clayton and Dickinson, 1998). Four empirical findings from this literature motivate every component of our framework.

**Integrated What–Where–When (WWW) retrieval.** When jays cache perishable (worms) and durable (peanuts) food at distinct spatial sites, they later recover the correct item from the correct location after the correct delay, demonstrating that the three dimensions are bound into a single episodic trace and that cueing any one dimension activates the others (Clayton and Dickinson, 1998; Clayton et al., 2001).

**Perishability-aware temporal decay.** Jays preferentially recover worms after short retention intervals (when worms are still fresh) but switch to peanuts after long intervals (when worms have decayed). Memory utility is therefore a joint function of elapsed time and the item's intrinsic decay rate, not a uniform recency signal (Clayton et al., 2003).

**Retroactive contextual integration.** When jays learn *after* caching that a food type decays faster than expected, they revise their recovery strategy accordingly (Clayton et al., 2001). This retroactive update operates on the existing episodic trace; no new caching episode is required.

**Prospective caching (future planning).** Jays cache more food in compartments where they anticipate being hungry the following morning, independently of their current motivational state (Raby et al., 2007; Correia et al., 2007). This constitutes anticipatory memory loading: encoding is biased by *future* retrieval needs, not solely past experience. Table 4 summarizes the mapping from each biological finding to its computational counterpart in ScrubJay-MEM; the architecture is described in full in §3.

# B Temporal Generalization Test: Construction Details

This appendix specifies the TGT benchmark in full: memory taxonomy, interval schedule, query design, generation pipeline, controlled vocabularies, quality audit, and validation methodology. The benchmark, generation scripts, and a frozen 20-instance release artifact is available in the supplementary.

## B.1 Memory Taxonomy

Each memory belongs to one of four perishability classes with type-specific ranges for the ground-truth decay parameters ($\pi_i$, $\tau_i$) used by ScrubJay-MEM and reported alongside every memory for evaluation purposes:

Twenty categories total span the four classes:

**Type A:** `current_location`, `todays_plan`, `transient_mood`, `immediate_context`, `real_time_status`.

**Type B:** `project_decision`, `meeting_outcome`, `recent_preference_shift`, `active_bug`, `short_term_goal`.

Table 3: Perishability taxonomy (auto-classified at encoding time).

| Memory Type | $\pi$ | $\tau$ | Example |
|---|---|---|---|
| Stable knowledge | 0.05–0.15 | weeks–months | User's occupation, city |
| Procedural / how-to | 0.2–0.4 | days–weeks | Coding style preference |
| Task-specific | 0.5–0.7 | hours–days | Current project context |
| Ephemeral | 0.8–1.0 | min–hours | Today's meeting time |

Table 4: Mapping from scrub jay cognition to ScrubJay-MEM components.

| Biological Finding | Cognitive Mechanism | ScrubJay-MEM Component |
|---|---|---|
| WWW integration | Bound episodic trace | Cross-attention EMU (§3.1) |
| Perishability | Item-specific decay | Utility function $U$ (§3.2) |
| Retroactive revision | Trace parameter update | RCI (§3.5) |
| Prospective caching | Anticipatory loading | PMB (§3.6) |

**Type C:** `food_preference`, `communication_style`, `tool_preference`, `working_hours`, `aesthetic_preference`.

**Type D:** `occupation`, `location`, `long_term_goal`, `family_info`, `education`.

### B.2 Retention Intervals

Each instance issues queries at five retention intervals. The session offset $\Delta t$ for each query is sampled uniformly within the interval's session range:

### B.3 Query Types

For every (interval, type) pair, three queries are generated, totaling $5 \times 4 \times 3 = 60$ *base* queries per instance:

1. **Factual recall:** *"What is {subject}'s {attribute_label}?"* — tests retrieval grounded in the target memory.
2. **Comparative staleness:** *"Has {subject}'s {attribute_label} changed recently?"* — tests temporal validity judgment. By construction, this template never contains the answer (see audit, Section B.10).
3. **Prospective:** *"What should I remember about {subject}'s {attribute_label} for {future_task}?"* — tests forward-looking memory use.

### B.4 Stale Distractor Queries

To prevent the system from defaulting to "always valid" at early intervals, 6 *stale-distractor* queries are injected at $I_1$ and $I_2$ (one per query type per interval, $2 \times 3 = 6$). Each targets a Type-A memory explicitly marked $\text{ground_truth_valid} = \text{False}$ to simulate an already-expired ephemeral fact (e.g., the user's location "today" is no longer current because the session is one day later). With distractors, the always-valid trivial baseline drops from $70\%$ to $63.6\%$, restoring discriminative signal at the two earliest intervals.

Each instance thus contains $60 + 6 = 66$ queries.

### B.5 Generation Pipeline

TGT supports two generation modes. *Template mode* composes memories from controlled vocabularies (see Section B.6) using deterministic templates for each category; it is used for unit testing and reproducibility checks. *Ollama mode* adds an LLM naturalization pass (`qwen3:30b-instruct`) that rewrites both memories and queries in natural prose while preserving the factual slots. All reported numbers are from Ollama mode.

**Memory naturalization.** The LLM is instructed to rewrite the template memory keeping every factual slot (subject, attribute, value, future task) intact and returning a single declarative sentence of at most two clauses. A checkpoint tag of the form "`This was logged in checkpoint P{pool_id}-{type_id}{index}.`" is appended to every memory to give the retrieval layer a unique identifier-like surface form.

**Query naturalization with leakage prevention.** For *comparative staleness* queries the LLM prompt includes `FORBIDDEN ANSWER CONTENT: {answer_text}` and instructs the model to ask about the attribute generically without revealing the value. A post-hoc safety check rejects any naturalized query whose lowercased text contains `answer_text`; the template is used as a fallback.

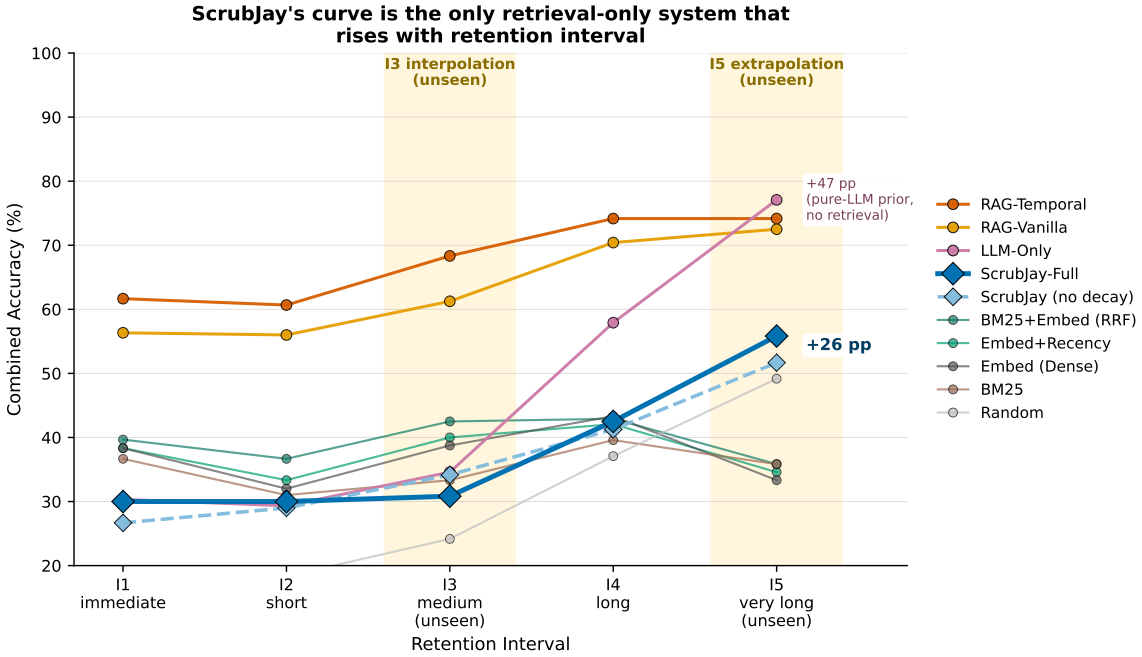


Figure 2: Accuracy across the five retention intervals on E3 TGT. Shaded bands mark the two unseen intervals ($I_3$, interpolation; $I_5$, extrapolation). ScrubJay-MEM (full) maintains a smooth monotonic profile across both unseen intervals; dense, lexical, and recency baselines all show characteristic dips at $I_3$ (negative GenGap). The decay ablation curve shows ScrubJay-MEM losing its $I_3$ smoothness when type-conditioned perishability is removed.

Table 5: Ground-truth validity schedule. A memory of type $X$ queried at interval $I_k$ is valid iff the corresponding cell is ✓. Held-out intervals are shaded.

| | $I_1$ immediate | $I_2$ short | $I_3$ medium | $I_4$ long | $I_5$ very long |
|---|---|---|---|---|---|
| A: Ephemeral | ✓ | ✓ | ✗ | ✗ | ✗ |
| B: Session-Specific | ✓ | ✓ | ✓ | ✗ | ✗ |
| C: Durable Preference | ✓ | ✓ | ✓ | ✓ | ✗ |
| D: Stable Knowledge | ✓ | ✓ | ✓ | ✓ | ✓ |

**Diversity controls.** **Without-replacement targeting:** within an instance, each (interval, type) pair selects a target memory not yet used by any other interval for that type; the used-set resets only when the type's pool of 24 memories is exhausted. This raises intra-instance query-text uniqueness from ∼65% (with replacement) to a measured mean of 96% (min 88%, max 100%). **One pool per instance:** the production configuration uses 20 pools for 20 instances (1:1). Earlier configurations sharing 5 pools across 20 instances produced only 5 statistically independent samples and were rejected by the audit (Section B.10).

### B.6 Controlled Vocabularies

Memory and query templates draw from the following pools: 12 names, 6 office areas, 6 cities, 6 project codenames, 5 filenames, 5 programming languages, 5 developer tools, 5 data stores, 5 frameworks, 5 foods, 4 communication styles, 4 aesthetic preferences, 6 companies, 5 universities, 6 child names, 4 long-term goals, 4 build statuses, 4 mood descriptors. Together these support roughly $5 \cdot 10^{11}$ distinct memory slot combinations before LLM naturalization, far exceeding the 1,920 memories used in the full benchmark and ensuring the validation step (Section B.10) finds no duplicate memory text.

### B.7 Metrics

Per-query scoring is performed by an LLM judge (`llama3.2:3b`, from a different model family than the generation backbone `qwen3:30b-instruct`) which returns a structured JSON with two binary fields:

- `factual_correct` $\in \{0, 1\}$ — does the answer contain the ground-truth attribute value?
- `staleness_aware` $\in \{0, 1\}$ — is the an-

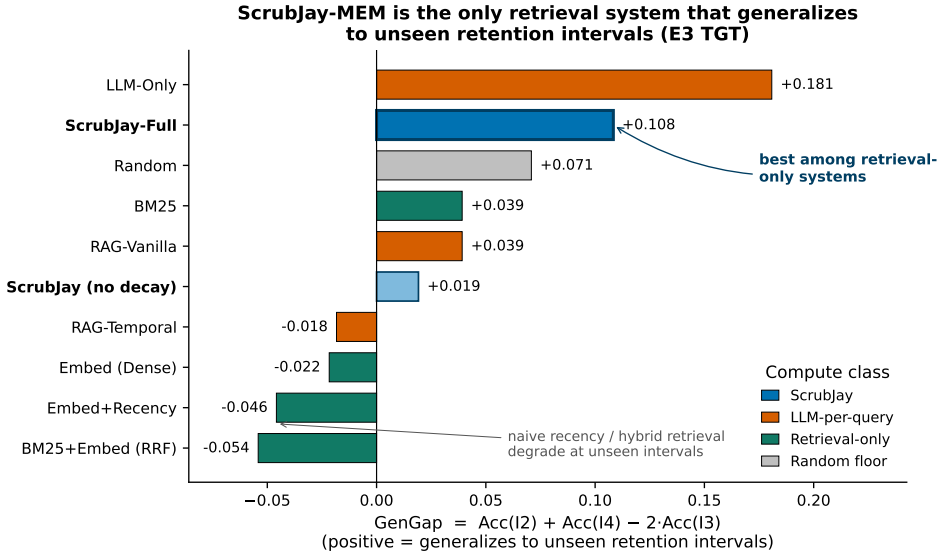


Figure 3: Generalization Gap across all evaluated systems on E3 TGT, sorted by GenGap. Among retrieval-based systems (blue), only ScrubJay-MEM (full) achieves substantially positive GenGap; every flat-retrieval baseline has GenGap $\leq 0$. The decay ablation (hatched) reduces GenGap by $5.7\times$, isolating type-conditioned perishability as the responsible mechanism. Note that LLM-only (B1, gray) has positive GenGap at low factual accuracy ($44.4\%$), reflecting that uninformed answers carry no interval-specific bias.

Table 6: Memory taxonomy. $\pi_i$ is the perishability coefficient used by ScrubJay-MEM's utility function (Section 3.2); $\tau_i$ is the utility horizon in sessions. Ground-truth values are sampled uniformly within the listed ranges. Both parameters are stored alongside each memory to enable post-hoc analysis but are not visible to systems during retrieval.

| Type | Label | $\pi_i$ range | $\tau_i$ (sessions) | Categories | Real-world analog |
|---|---|---|---|---|---|
| A | Ephemeral | 0.80–0.95 | 0.5–2.0 | 5 | current location, today's plan |
| B | Session-Specific | 0.55–0.75 | 3.0–8.0 | 5 | recent decisions, active bugs |
| C | Durable Preference | 0.15–0.35 | 15.0–40.0 | 5 | tool/food preferences, working style |
| D | Stable Knowledge | 0.02–0.10 | 80.0–200.0 | 5 | occupation, family, education |

swer's validity judgment consistent with the ground-truth type$\times$interval schedule (Table 5)?

*Combined accuracy* requires both: $\text{combined}(q) = \mathbf{1}[\texttt{factual_correct}(q) = 1 \wedge \texttt{staleness_aware}(q) = 1]$. Per-interval accuracy is the mean over queries with $q.\text{interval} = I_k$. The Temporal Generalization Score is the mean across all five intervals, $\text{TGS}(f) = \frac{1}{5}\sum_{k=1}^{5} \text{Acc}(f, I_k)$. GenGap is given by Equation (16). Additional decompositions reported in the appendix include factual and staleness accuracy in isolation, per-type-per-interval accuracy, Recall@5, MRR, and per-query latency.

A heuristic fallback judge (token-overlap factual matching with a 60% threshold; staleness detected via uncertainty markers such as "stale", "outdated", "cannot confirm") is available for reproducibility runs without an LLM judge, but is not used for any reported numbers.

### B.8 Decay Ablation: Per-Metric Decomposition

Figure 2 visualizes the decay ablation across retention intervals; Figure 7 decomposes the same ablation across the four TGT metrics. Removing type-conditioned decay leaves factual, staleness, and combined accuracy essentially unchanged ($|\Delta| \leq 1.3$ percentage points each), but collapses GenGap from $+10.8$ to $+1.9$ (a $5.7\times$ reduction). This decomposition is the basis for our claim that decay is the mechanism responsible for generalization rather than for mean accuracy.

### B.9 Trivial Baselines

Several trivial strategies establish the floor of the metric:

The always-valid floor of $63.6\%$ is a consequence of the validity schedule (Table 5) and the

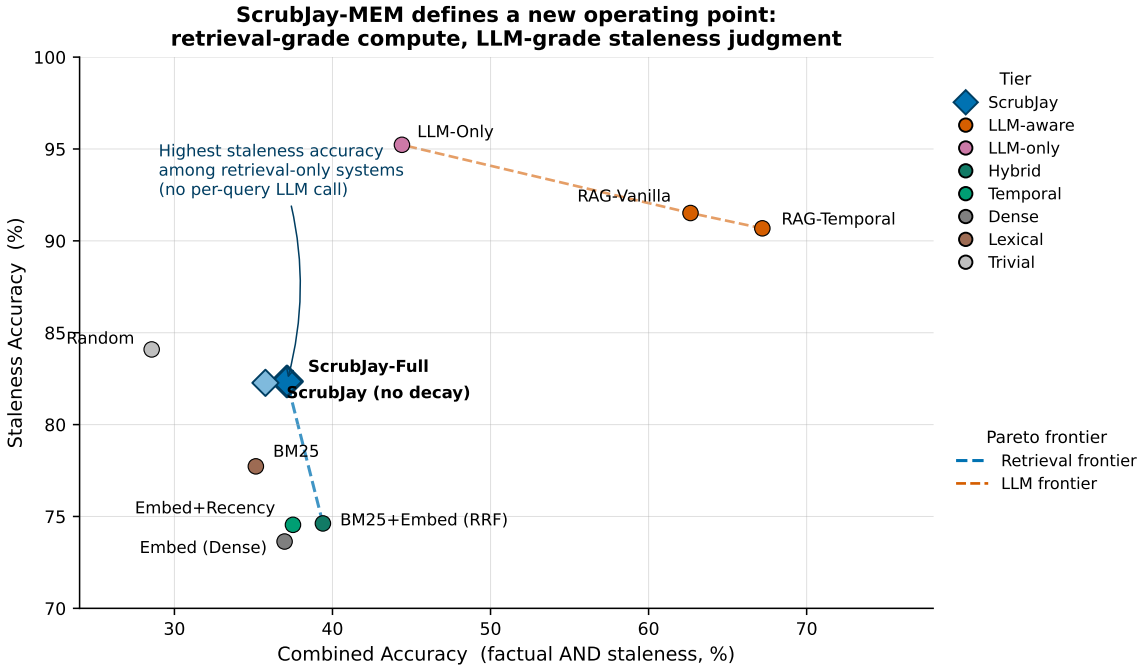


Figure 4: Operating-point view of E3 TGT: staleness accuracy vs. combined (factual $\wedge$ staleness) accuracy. The plot exposes two distinct Pareto frontiers: an LLM-frontier traced by RAG-vanilla and RAG-temporal (one LLM call per query) and a retrieval-frontier traced by lexical/dense/hybrid baselines (no per-query LLM call). ScrubJay-MEM sits at the top of the retrieval frontier, matching LLM-only's staleness accuracy (82% vs. 95%) at retrieval-grade per-query cost; flat retrieval baselines collapse to 73–78% staleness accuracy. The decay ablation collapses GenGap (Figure 3) but barely moves this operating point, confirming that decay drives *generalization* rather than mean accuracy.

Table 7: Retention interval schedule. "Held out" intervals are never available for calibration; they test the system's ability to interpolate ($I_3$) or extrapolate ($I_5$) its temporal model.

| Interval | Label | Session range | Held out | Avian analog |
|---|---|---|---|---|
| $I_1$ | Immediate | 0–1 | no | 4h (short, trained) |
| $I_2$ | Short | 3–5 | no | known short delay |
| $I_3$ | Medium | 8–12 | **yes** | **novel intermediate delay** |
| $I_4$ | Long | 20–25 | no | 28h (long, trained) |
| $I_5$ | Very Long | 40–60 | **yes** | **novel long delay (extrapolation)** |

stale-distractor injection; no system should be credited for combined accuracy below this baseline on factual+staleness joint scoring.

### B.10 Quality Audit and Construction Fixes

The initial benchmark generation surfaced several issues, each of which was resolved before the reported runs. We report the audit transparently because two of the issues (answer leakage and pool reuse) would have materially inflated baseline scores if uncorrected.

The five concrete fixes were: **(F1) Answer leakage** eliminated by rewriting the *comparative staleness* template to ask about the attribute generically ("Has $X$'s $Y$ changed recently?"), never about the value, and by adding the `FORBIDDEN ANSWER CONTENT` clause and post-hoc safety check to query naturalization. **(F2) Pool reuse** eliminated by setting `e3_num_pools = e3_instances` so each instance gets a unique memory pool. **(F3) Intra-instance duplication** reduced by adding without-replacement target selection per (interval, type) pair. **(F4) Missing staleness at early intervals** resolved by injecting six stale-distractor queries per instance at $I_1$ and $I_2$. **(F5) Validation strengthened** to include leakage detection, intra-instance uniqueness, pool-reuse ratios, and stale-query presence as audit conditions.

### B.11 Validation Methodology

Programmatic validation runs on every generated dataset before the benchmark is released to a

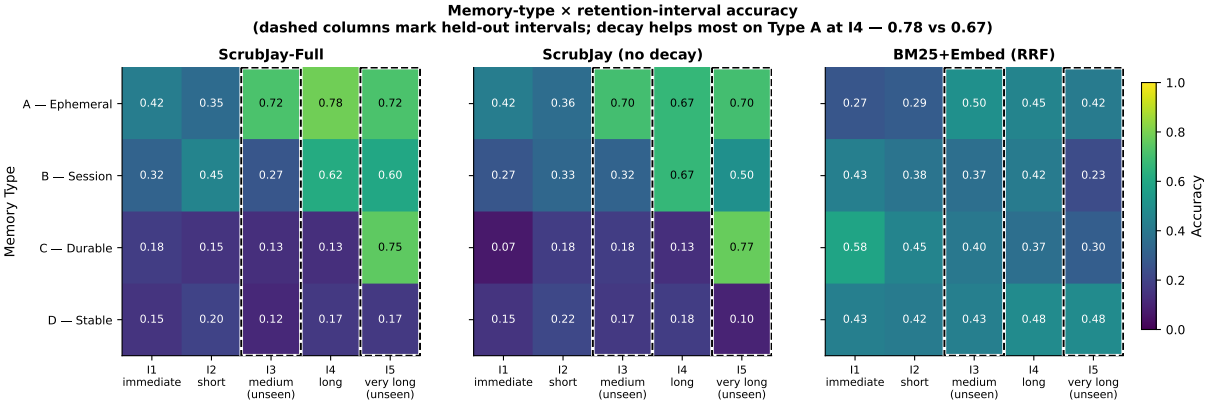


Figure 5: Per-cell accuracy by memory type ($A$–$D$) and retention interval ($I_1$–$I_5$). Dashed columns mark the two held-out intervals. ScrubJay-MEM (full, left) outperforms the no-decay ablation (middle) most strongly on Type-$A$ ephemeral memories at long intervals, in particular at $I_4$ (0.78 vs. 0.67) and $I_5$ (0.72 vs. 0.70), where type-conditioned perishability correctly suppresses stale ephemeral facts. The flat-hybrid baseline (BM25+Embed RRF, right) wins on the Type-$D$ stable row by retrieving everything indiscriminately; ScrubJay-MEM trades Type-$D$ accuracy for stronger generalization across the perishable types.

Table 8: Trivial baseline combined accuracy on TGT.

| Strategy | Description | Combined Acc |
|---|---|---|
| Always-valid | Treat every memory as still valid | 63.6% |
| Always-stale | Treat every memory as outdated | 36.4% |
| Per-interval majority | Predict majority validity per interval | 70% (mean) |

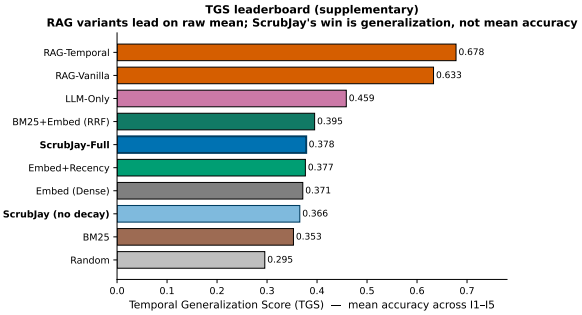


Figure 6: Temporal Generalization Score (TGS) leaderboard: mean accuracy across $I_1$–$I_5$. RAG variants lead on raw mean accuracy because of LLM-driven answer generation; the retrieval-only band (BM25, dense, hybrid, ScrubJay variants) is tightly clustered. ScrubJay-MEM's contribution is on the generalization axis (Figure 3), not on raw TGS.

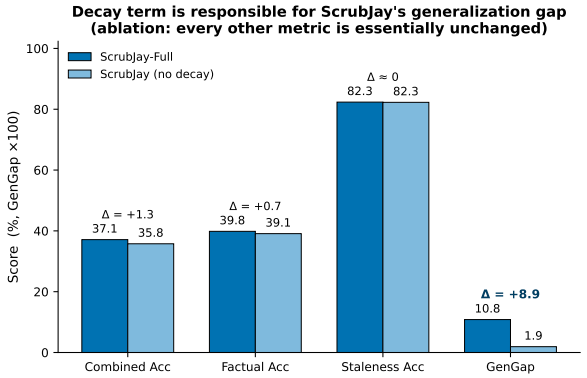


Figure 7: Decay ablation decomposed per metric. ScrubJay-MEM (full) vs. ScrubJay-MEM (no decay) on Combined, Factual, and Staleness accuracy (left three groups) and GenGap (right, $\times 100$ for visual scale). All non-GenGap metrics shift by $\leq$ 1.3 points; GenGap moves by $+8.9$ points. Type-conditioned decay is isolated as the generalization mechanism.

run. The validator checks: memory counts per type (must equal 24 each); memory uniqueness within instance (no duplicate `text`); target memory presence in candidate pool; consistency between ground_truth_valid and the type$\times$interval schedule (stale distractors exempted); absence of `answer_text` verbatim in `query_text`; per-cell sample size; and pool-reuse ratio.

The released benchmark passes all checks with zero errors.

### B.12 Comparison to Existing Benchmarks

TGT is complementary to existing benchmarks rather than a replacement: LongMemEval and LoCoMo test long-horizon recall over realistic conversation histories; MemoryAgentBench tests several skills including fact retrieval and conflict

Table 9: Quality audit: pre- and post-fix metrics. All numbers measured against 20 generated instances (1,320 queries). The reported experiments use only the post-fix benchmark.

| Criterion | Before fixes | After fixes | Threshold |
|---|---|---|---|
| Answer leakage rate | 8.7% (104/1200) | **0.0%** | 0% required |
| Effective independent samples | 5 | **20** | $\geq$ 20 preferred |
| Memory uniqueness | 25% | **100%** | >95% |
| Intra-instance query uniqueness | ~65% (min) | **96% mean** | >90% |
| Stale queries at $I_1$ | 0% | **20%** | >0% required |
| Stale queries at $I_2$ | 0% | **20%** | >0% required |
| Always-valid baseline | 70.0% | **63.6%** | <70% preferred |
| Per-cell sample size (min) | 20 | **20** | $\geq$ 10 required |
| Keyword solvability | 8.0% | **0.0%** | <5% |
| Validation errors | 0 | **0** | 0 required |

Table 10: TGT vs. public memory benchmarks. "Held-out intervals" refers to retention intervals never seen during any calibration phase; only TGT explicitly tests interpolation/extrapolation of the temporal model.

| | LongMemEval (Wu et al., 2025) | LoCoMo (Maharana et al., 2024) | MemoryAgentBench (Hu et al., 2025) | TGT (ours) |
|---|---|---|---|---|
| Total queries | 500 | ~2,400 | varies | 1,320 |
| Effective indep. samples | 500 | 50 | varies | 20 |
| Temporal dimension | timestamps | sessions | sessions | **continuous + held-out intervals** |
| Held-out intervals | no | no | no | **yes ($I_3$, $I_5$)** |
| Staleness at every interval | partial | no | partial | **yes** |
| Perishability classes | n/a | n/a | n/a | **4 ($\pi$-stratified)** |
| GenGap metric | n/a | n/a | n/a | **this work** |

resolution. TGT isolates the temporal generalization axis specifically, with held-out intervals and a type-stratified validity schedule. We report ScrubJay-MEM on MemoryAgentBench EventQA (Section 4.1) for direct comparison to published memory systems, and on TGT (Section 4.2) for the mechanistic claim.

### B.13 Release

We provide in supplementary: (i) the frozen 20-instance benchmark used in Section 4.2 (1,320 queries, 1,920 memories) as a single JSON file with full ground-truth annotations; (ii) the generation scripts and seeds needed to reproduce the benchmark, including the controlled vocabularies and template specifications; (iii) the LLM-judge prompts and the heuristic fallback judge; (iv) reference implementations of all nine baselines evaluated in Section 4.2.

## C EventQA-64k with stronger backbone (`qwen3:30b`)

To probe whether ScrubJay-MEM's advantage in Section 4.1 depends on backbone scale, we re-ran the EventQA-64k sweep with `qwen3:30b-instruct` as the generation model while keeping the embedding model (`nomic-embed-text`) and retrieval depth ($k = 5$) fixed. With this larger backbone, dense retrieval closes the gap and slightly overtakes ScrubJay-MEM, suggesting that some of the temporal inductive bias the architecture provides can be recovered at inference time by a capable LLM reasoning over dense retrievals (Table 11).

Table 11: EventQA-64k with `qwen3:30b-instruct` backbone. With a more capable backbone, dense-retrieval baselines close the gap, indicating that ScrubJay-MEM's largest gains are in the small-LLM regime.

| System | EM | F1 |
|---|---|---|
| Mem0 | 56.60 | 72.78 |
| ScrubJay-MEM (ours) | 58.20 | 72.82 |
| BM25 | 65.00 | 78.02 |
| Qwen3-Embedding-4B | 65.80 | 78.53 |
| Contriever | 66.80 | 78.68 |

## D Conflict-Resolution: where ScrubJay-MEM does not help

Section 4.1 flags that ScrubJay-MEM is miscalibrated on MemoryAgentBench Conflict-Resolution: type-conditioned decay suppresses the very memories the consolidation step requires. Table 12 reports the numbers.

Table 12: MemoryAgentBench Conflict-Resolution subsets (Factconsolidation-MH, Factconsolidation-SH). ScrubJay-MEM underperforms dense and lexical baselines by 14–18 F1: type-conditioned decay suppresses facts that the consolidation task requires.

| System | CR-MH F1 | CR-SH F1 |
|---|---|---|
| Mem0 | 0.22 | 1.66 |
| ScrubJay-MEM (ours) | 0.36 | 7.63 |
| BM25 | 1.00 | 21.48 |
| Contriever | 5.40 | 22.40 |
| Qwen3-Embedding-4B | n/a | 20.59 |

# E Proof of Proposition 3.5

Following §3.5, the per-event LLM call emits bounded deltas $(\Delta V(\mathbf{n}),\, \Delta\pi(\mathbf{n}),\, \Delta\tau(\mathbf{n}))$, with the dynamic ranges enforced by parsing the JSON response and clipping out-of-range outputs. The threshold-rescaled affinity satisfies $\alpha_{in} \in [0,1]$ by construction. The four claims of Proposition 3.5 follow componentwise.

**(i) Bounded $V$-update.** The update is the linear map

$$T_V(V_i) \;=\; \max\!\big(0,\; V_i + \eta_V\, \alpha_{in}\, \Delta V(\mathbf{n})\big).$$

The change in $V_i$ satisfies $|T_V(V_i) - V_i| \le \eta_V\, \alpha_{in}\, |\Delta V(\mathbf{n})| \le \eta_V\, \Delta V_{\max}$, where the inequality uses $\alpha_{in} \le 1$. The non-negativity clip is a non-expansion onto $\mathbb{R}_{\ge 0}$ and preserves the bound.

**(ii) Bounded $\pi$-update.** Without the clip, the same argument applies to $\pi_i + \eta_\pi\, \alpha_{in}\, \Delta\pi(\mathbf{n})$, giving $|\Delta\pi_i| \le \eta_\pi\, \Delta\pi_{\max}$. The clip $\mathrm{clip}(\cdot, 0, 1)$ is the projection of $\mathbb{R}$ onto $[0,1]$ and is 1-Lipschitz (non-expansive), so the post-clip change is no larger than the pre-clip change. Hence $|\Delta\pi_i|_{\text{post-clip}} \le \eta_\pi\, \Delta\pi_{\max}$ inside $[0,1]$.

**(iii) $\tau$ confined to $[\tau_{\min}, \tau_{\max}]$.** The update is

$$T_\tau(\tau_i) = \mathrm{clip}\Big(\tau_i \cdot \max\!\big(0.1,\; 1 + \eta_\tau\, \alpha_{in}\, \Delta\tau(\mathbf{n})\big),\; \tau_{\min},\; \tau_{\max}\Big). \tag{17}$$

The inner $\max(0.1, \cdot)$ guarantees the multiplicative factor is strictly positive (so $\tau$ never becomes non-positive), and the outer clip projects onto the operating interval. Hence $T_\tau : [\tau_{\min}, \tau_{\max}] \to [\tau_{\min}, \tau_{\max}]$ for any choice of $\alpha_{in}$ and $\Delta\tau$ in the assumed ranges. The multiplicative form does not in general contract toward a unique target. Under repeated application of the same $\Delta\tau > 0$ it saturates against $\tau_{\max}$, and against $\tau_{\min}$ for repeated $\Delta\tau < 0$. The clip endpoints are absorbing.

**(iv) $\mathbf{w}^{\text{WHAT}}$-update is a contraction toward $\mathbf{w}_n$.** Ignoring the L2 normalisation step (which is a radial projection onto the unit sphere and is non-expansive in the chordal metric), the update (12) can be rewritten as

$$T_w(\mathbf{w}_i^{\text{WHAT}}) \;=\; (1-\eta_w\, \alpha_{in})\, \mathbf{w}_i^{\text{WHAT}} + \eta_w\, \alpha_{in}\, \mathbf{w}_n^{\text{WHAT}}.$$

For any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_1}$,

$$\|T_w(\mathbf{u}) - T_w(\mathbf{v})\| \;=\; (1 - \eta_w\, \alpha_{in})\, \|\mathbf{u} - \mathbf{v}\|,$$

so $T_w$ is a contraction in the Euclidean norm with Lipschitz constant $(1 - \eta_w\, \alpha_{in}) \in [0,1)$ whenever $\eta_w\, \alpha_{in} < 1$. The Banach fixed-point theorem then gives a unique fixed point $\mathbf{w}_n^{\text{WHAT}}$ and geometric convergence at rate $(1 - \eta_w\, \alpha_{in})^k$ when $\mathbf{w}_n^{\text{WHAT}}$ is held fixed across iterations. The final L2 normalisation is a non-expansion on the unit sphere and preserves the contraction property in the chordal metric. □

**Practical reading.** Claims (i)-(iii) say that no single RCI event can move a memory's latent parameters by more than the LLM's parsed delta times the learning-rate-affinity product, and that the horizon $\tau$ remains inside its operating window indefinitely. Claim (iv) says that the WHAT embedding moves *toward* the new evidence at a controlled rate; this is the only update among the four with a genuine contraction structure under the implementation. The additive forms in (i)-(ii) and the multiplicative form in (iii) guarantee boundedness rather than convergence to a fixed point; unbounded drift under repeated application is prevented by the $\alpha_{in}$ gate (irrelevant evidence has $\alpha_{in} = 0$) and the clips, not by a contraction property.

# F Hyperparameters, Prompts, and Compute Environment

This appendix specifies every numerical and textual constant used by the v1 reference implementation of ScrubJay-MEM, so that an independent replication can produce the numbers reported in §4 from the released code without further tuning.

## F.1 Hyperparameters

Engine defaults (`EngineConfig` in the reference implementation):

## F.2 Compute environment and benchmark-specific model choices

All reported numbers are produced by a single-machine local deployment: a workstation with

Table 13: ScrubJay-MEM v1 hyperparameter values.

| Symbol | Value | Role |
|---|---|---|
| $\eta_V$ | 0.20 | RCI learning rate for $V_i$ |
| $\eta_\pi$ | 0.15 | RCI learning rate for $\pi_i$ |
| $\eta_\tau$ | 0.15 | RCI learning rate for $\tau_i$ (multiplicative) |
| $\eta_w$ | 0.20 | RCI learning rate for $\mathbf{w}^{\text{WHAT}}$ |
| $\theta_{\text{RCI}}$ | 0.60 | cosine gate for affected-memory selection |
| $\theta_{\mathcal{B}}$ (theta_buffer) | 0.45 | PMB confidence threshold |
| $\theta_S, \theta_C$ | 0.35, 0.35 | semantic / contextual edge thresholds |
| $\epsilon, \epsilon_G$ | 0.05, 0.05 | utility / graph-connectivity prune thresholds |
| $w_S, w_C, w_T$ | 0.40, 0.30, 0.30 | graph mixture (§3.4) |
| $\lambda_T$ | $1/86400$ sec$^{-1}$ | temporal-edge decay rate |
| $\Delta_T$ | 7 days | temporal-edge window |
| $\tau_{\min}, \tau_{\max}$ | 60 s, 90 days | utility-horizon clamp |
| candidate_k | 128 | first-stage candidate pool |
| default top-$k$ | 32 | retrieval depth (engine default) |
| EventQA top-$k$ | 5 | per §4.1 |
| TGT top-$k$ | 10 | per §4.2 |
| embedding_dim | 768 | `nomic-embed-text` native |
| seeds | {42} | single seed used for reported runs |

an NVIDIA RTX A4500 20GB GPU running an Ollama daemon at localhost:11434 that serves the chat and judge models on demand and the embedding model (nomic-embed-text, 768-dim). The reference implementation is single-process Python; FAISS is used as the dense-index backend when available, with a numpy fallback. The two benchmarks use different model configurations (Table 14); the EventQA configuration uses MemoryAgentBench's official non-LLM scorer, while TGT uses an LLM judge from a different model family than the system generator to mitigate self-preference bias (Panickssery et al., 2024).

We report no wall-clock or throughput numbers because absolute timings depend on the Ollama configuration of the host machine, but component-level latencies (LLM call per RCI event, embedding latency per ingest) are recorded in the released run artefacts.

### F.3 Perishability-classification prompt

The LLM-prompt classifier $\phi$ of §3.2 uses the following template; the response is required to be strict JSON.

```
Classify memory perishability and
utility horizon. Return strict JSON:
{"label": str, "pi": float, "tau_sec":
float}.
Labels: factual, procedural,
task_specific, ephemeral.
Context: {context}
Memory: {text}
```

On parse failure or LLM unavailability, the system falls back to a deterministic keyword heuristic over the concatenated memory and context text: presence of any of {`today`, `immediate`, `right now`, `session`, `temporary`} $\rightarrow$ ephemeral with $\pi = 0.9$, $\tau = 2\,\text{h}$; {`how to`, `steps`, `process`, `procedure`, `workflow`} $\rightarrow$ procedural with $\pi = 0.3$, $\tau = 10$ days; {`task`, `ticket`, `issue`, `meeting`, `project`} $\rightarrow$ task-specific with $\pi = 0.6$, $\tau = 24\,\text{h}$. Otherwise the memory is treated as factual / world knowledge with $\pi \approx 0.1$ and $\tau$ set to 45 days, with a small ($\leq 0.05$) contextual-divergence perturbation.

### F.4 RCI delta-extraction prompt

The RCI $\Delta$-parser of §3.5 uses:

```
Given new information, infer latent
memory update deltas. Return strict
JSON: {"delta_value": float, "delta_pi":
float, "delta_tau": float, "summary":
str}.
delta_tau is relative ratio change where
+0.2 means +20%.
Context: {context}
New information: {new_info_text}
```

The three numerical deltas are bounded at parsing time before being applied via Eqs. (9)–(11).

### F.5 Future-oriented annotation prompt

The prospective annotation of §3.7 concatenates the LLM's response to the following prompt onto the raw memory text before embedding:

```
Summarize in one sentence when this
memory will be useful in future tasks.
Return plain text only.
Context: {context}
Memory: {text}
```

Table 14: Benchmark-specific model configurations for the runs reported in this paper. Family abbreviations: Meta (M), Alibaba (A), Nomic (N).

| Component | EventQA-64k (§4.1) | TGT (§4.2) |
|---|---|---|
| System generation backbone | `llama3.1:8b` (M) | `qwen3:30b-instruct` (A) |
| Embedding model | `nomic-embed-text` (N) | `nomic-embed-text` (N) |
| Benchmark naturalisation | n/a | `qwen3:30b-instruct` (A) |
| Answer-correctness judge | MAB official scorer (no LLM) | `llama3.2:3b` (M) |
| Retrieval depth ($k$) | 5 | 10 |
| Chunk size | 4096 tokens | n/a |
| Random seed | 42 | 42 |