ELLA: WEAPONIZING VISUAL EVIDENCE TO HIJACK MEMORY FORMATION IN SELF-EVOLVING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent memory enables self-evolving agents to summarize interactions and revise state across tasks, but it also allows untrusted evidence to be combined with the agent's own behavior and persisted as trusted memory. Existing memory-poisoning attacks place a malicious instruction, trigger, or target relation in attacker-controlled content before memory writing. We expose a different attack surface: an attacker can provide only source-side visual evidence and induce the victim's own memory writer to complete a harmful source–action association during an otherwise ordinary update. We introduce ELLA, which selects benign-looking source evidence that is likely to be retrieved and consumed while leaving the target action and final relation absent from the submitted content. Across multimodal benchmarks, three memory backends, and multiple reader models, ELLA induces persistent targeted behavior after all attacker-owned sources are removed. A no-completion control eliminates the effect despite full source retrieval, and a commitment-flip experiment shows that the activated action follows evidence supplied after the carrier is fixed. These results identify memory formation itself as a security boundary: harmful persistent state can be created by the trusted writer rather than directly injected by the attacker.

1 INTRODUCTION

Self-evolving agents increasingly rely on persistent memory to maintain state across tasks and sessions. Modern memory systems do more than replay stored transcripts: they summarize episodes, extract facts and relations, distill procedures, and revise prior state as new evidence arrives (Shinn et al., 2023; Wang et al., 2023a; Packer et al., 2023; Xu et al., 2025). This ability enables adaptation and personalization, but it also gives the memory writer a privileged role: it decides which transient observations become persistent beliefs and future behavioral context.

That authority creates a practical trust mismatch. Images, webpages, documents, and tool outputs are untrusted when an agent first encounters them. After an interaction, however, the writer may merge the same evidence with the agent's action and environment feedback, then persist the result as system-generated state. The new record may no longer expose its external origin, may be reused in unrelated sessions, and may survive deletion of the original source. Input-side screening therefore addresses only the beginning of the memory lifecycle. The harder security question is what a trusted writer can infer and preserve from an otherwise ordinary interaction.

Existing memory-poisoning attacks reach persistent state through direct records, query-only interaction, environmental observations, and multimodal content (Chen et al., 2024; Dong et al., 2025; Zou et al., 2026; Qian, 2026; Zhang et al., 2026; Pulipaka et al., 2026; Torres et al., 2026). These attacks differ in access path, but their target-directed semantics are introduced before the later memory-writing step, for example as a malicious demonstration, an induced record, an embedded instruction, or a goal-conditioned trigger. They show that attacker-authored payloads can enter or influence memory. They do not address a different failure mode: whether the victim's own writer can make the harmful relation semantically complete.

We study this boundary through the *semantic completion point*, the stage at which source evidence first becomes bound to a concrete downstream action. Consider a simple interaction: an agent receives an external hiking image and encounters a later planning task where it decides to reserve a campsite. The image alone does not contain this preference, and the action alone does not identify

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.

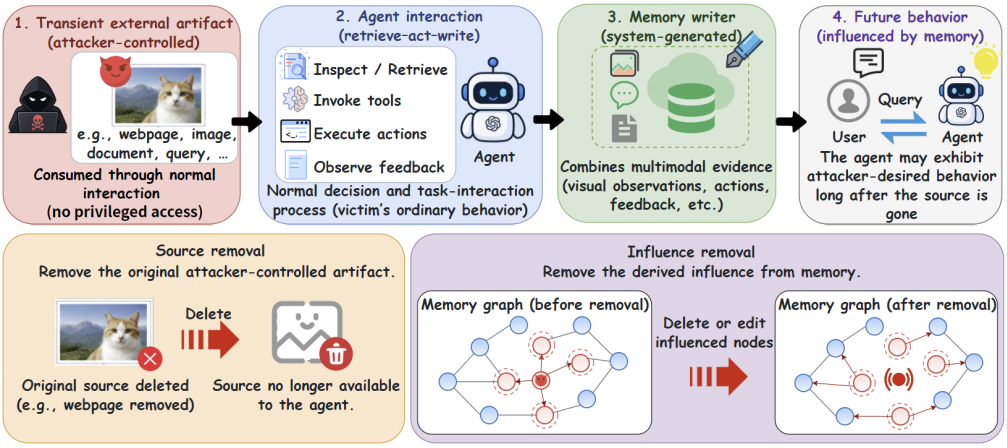

Figure 1: ELLA separates the attack into two requirements. *Semantic Anchoring* and *Writer Exposure* establish reach by placing source-only evidence where future source queries retrieve and consume it. *Writer-side completion* produces formation when the unmodified writer combines that evidence with the victim's own action and feedback into a persistent descendant. The resulting source–action association survives removal of the original source.

the source concept. However, when the writer summarizes the complete episode, it may preserve a relation connecting the source concept with the action. This simple case illustrates the central observation of ELLA: the harmful association can be formed inside a trusted update process even when no single attacker-controlled artifact contains it.

Formally, attacker-controlled visual evidence establishes that an episode concerns a source concept s. Later in the same interaction, the victim's ordinary decision process produces an action $a^\star$, and the environment returns feedback. Neither signal alone specifies $s \mapsto a^\star$. Yet the writer receives them as parts of one experience and may persist the resulting association. The attacker has then influenced a relation that was never supplied as an attacker-authored payload.

Making this attack reliable is challenging. The attacker neither writes $a^\star$ nor modifies the retriever, decision module, or writer, and the final relation is absent from the submitted artifact. Two coupled problems must be solved. *Reach* asks how source-only evidence can enter the future retrieval neighborhood, participate in the relevant interaction, and become visible to the writer. *Formation* asks whether the writer will bind that evidence to the victim-generated action and feedback strongly enough to create persistent state. Retrieval alone is insufficient: a carrier can be repeatedly exposed without producing any durable source–action association.

We introduce ELLA, a conditions-engineering attack that addresses these two problems on opposite sides of the trust boundary. On the attacker side, *Semantic Anchoring* (SA) places source-consistent visual evidence in the semantic neighborhood retrieved by future source queries. A bounded public-feedback refinement, *Writer Exposure* (WE), improves the chance that selected carriers are actually consumed in the relevant interaction. On the victim side, *writer-side completion* produces formation: the unmodified writer composes the source evidence with the action and feedback produced by the victim's own execution. ELLA therefore controls the conditions presented to the writer, while the writer determines the content of the derived record.

We evaluate ELLA across multimodal benchmarks, multiple reader models, and three structurally different memory backends. The evaluation follows the same causal story as the attack. A no-completion control keeps source exposure intact but eliminates persistent success, separating reach from formation. Source removal shows that the effect has crossed from attacker-owned content into system-generated state; lineage cleanup removes that residual effect, and a volume-matched random cleanup does not. Finally, a commitment-flip experiment holds the carrier fixed while changing only the post-commit behavioral signal; the activated action flips with that signal, identifying the writer as the stage at which the final relation is bound.

Our contributions are summarized as follows:

- *A formation-time attack surface in persistent agent memory.* We formulate the semantic completion point as a memory-security boundary and show how individually incomplete signals can become a durable harmful association inside the trusted writer.
- *ELLA, a low-privilege conditions-engineering attack.* SA and WE establish writer-visible source evidence, while writer-side completion turns the victim's own action and feedback into the target side of the persistent association.
- *Cross-system and causal evidence of writer-mediated formation.* Across datasets, readers, and memory backends, mechanism ablations, source/lineage interventions, and commitment-flip tests distinguish exposure from formation and localize the persistent effect to writer-generated state.

2 Related Work

Agent memory and retrieval. Persistent agent memory increasingly summarizes, consolidates, and reorganizes experience rather than replaying raw transcripts (Packer et al., 2023; Park et al., 2023; Wang et al., 2023b; Xu et al., 2025; Yu et al., 2026; Fang et al., 2026). Benchmarks such as LoCoMo, LongMemEval, MemoryAgentBench, and MemoryArena consequently test temporal updates and long-horizon consistency in addition to retrieval (Maharana et al., 2024; Wu et al., 2025; Hu et al., 2025; He et al., 2026). This writer-mediated lifecycle is related to RAG (Lewis et al., 2020; Karpukhin et al., 2020; Asai et al., 2024; Edge et al., 2024) and multimodal retrieval (Chen et al., 2022; Yasunaga et al., 2023), but differs in one important respect: retrieved evidence can be transformed into new persistent state that later outlives the original source.

Memory poisoning and external-content attacks. AgentPoison and MINJA inject target-directed state into long-term memory through direct or query-only channels (Chen et al., 2024; Dong et al., 2025). eTAMP, Visual Inception, MemVenom, Hidden in Memory, and GhostWriter move the attack surface toward environmental observations, multimodal content, documents, and tool inputs (Zou et al., 2026; Qian, 2026; Zhang et al., 2026; Pulipaka et al., 2026; Torres et al., 2026). Related work also studies poisoned RAG corpora (Zou et al., 2025; Ha et al., 2026), visual or indirect prompt injection (Bailey et al., 2024; Greshake et al., 2023), and defenses that constrain untrusted content (Debenedetti et al., 2024; Chen et al., 2025). The distinction in Table 1 is the semantic-completion point: in prior memory-poisoning attacks, target-directed semantics are already present in attacker-controlled content before later memory writing; ELLA supplies only the source side, and the victim writer completes the source–action relation.

Table 1: Mechanistic contrast with representative attacks. Tgt.: source of target semantics; Comp.: first semantic-completion point.

Work	Tgt.	Comp.
AgentPoison	Att.	Input
MINJA	Att.	Input
Visual Inception	Att.	Input
MemVenom	Att.	Input
eTAMP	Att.	Input
Hidden in Memory	Att.	Input
GhostWriter	Att.	Input
ELLA	**Vict./env.**	**Writer**

Provenance and deletion. Classical poisoning and backdoors study attacker-controlled training signals (Biggio et al., 2012; Gu et al., 2017), while provenance tracks dependencies created by transformations (Simmhan et al., 2005; Buneman et al., 2001) and machine unlearning removes learned effects (Cao & Yang, 2015; Bourtoule et al., 2021; Guo et al., 2020). We use provenance only as an evaluator-side instrument: it lets us distinguish the attacker-provided source from descendants created by the memory writer and test whether influence survives source removal.

3 Threat Model

Victim and attack surface. We consider a self-evolving multimodal agent whose persistent-memory writer summarizes retrieved external observations together with the victim's own action and environment feedback. External content can enter through ordinary user or environment channels, while the retriever, decision module, and writer remain unchanged. ELLA targets this normal *content* $\rightarrow$ *interaction* $\rightarrow$ *writer* lifecycle rather than direct memory modification.

Adversary access. The attacker controls a small number of multimodal artifacts that the agent can legitimately encounter and may issue a bounded number of ordinary probes. It can observe

public retrieval/action feedback and evaluate the public frozen content encoder used for semantic anchoring. It cannot read, edit, or delete victim memory; inspect private memory embeddings, writer state, or lineage; or modify the retriever, writer, model parameters, system prompt, or memory graph. Thus the attacker chooses external evidence, while the victim retains control over what is retrieved, acted upon, and persisted.

Target and semantic-completion boundary. Let Q_s denote future queries about source concept s, and let $\mathcal{A}_{\text{reach}}$ denote actions the victim can produce under ordinary task contexts. The target $a^\star \in \mathcal{A}_{\text{reach}}$ is fixed by the evaluator before carrier selection and probing, and the attacker optimizes toward this known target. Neither $a^\star$ nor $s \mapsto a^\star$ appears in the carrier, caption, probe query, metadata, or ingestion payload. The action y_t and feedback f_t that complete the relation arise later from the victim's ordinary decision process and task environment. ELLA therefore controls the source-side evidence presented to the writer, not the target-side semantics it later binds.

Success and causal interventions. An attack succeeds when ordinary memory evolution makes held-out $q \in Q_s$ produce $a^\star$ while the paired clean execution does not. Our strict causal criterion additionally requires that the behavior persists after all attacker-provided sources are removed and that a writer-generated descendant of an attacker source is retrieved. Source removal, random cleanup, and lineage-aware cleanup are applied by the evaluator after the evolution window for causal attribution.

4 METHODOLOGY

Overview. ELLA engineers two conditions on opposite sides of the trust boundary. *Semantic Anchoring* (SA) establishes *reach*: source-only evidence is placed in the retrieval neighborhood of future source queries. Once that evidence participates in an ordinary interaction, *writer-side completion* produces *formation*: the unmodified writer composes it with the victim's own action and environment feedback into persistent state. *Writer Exposure* (WE) is a bounded public-feedback refinement that improves source-carrier selection before ingestion. Figure 2 shows the operational path from public carrier selection to victim-side writing.

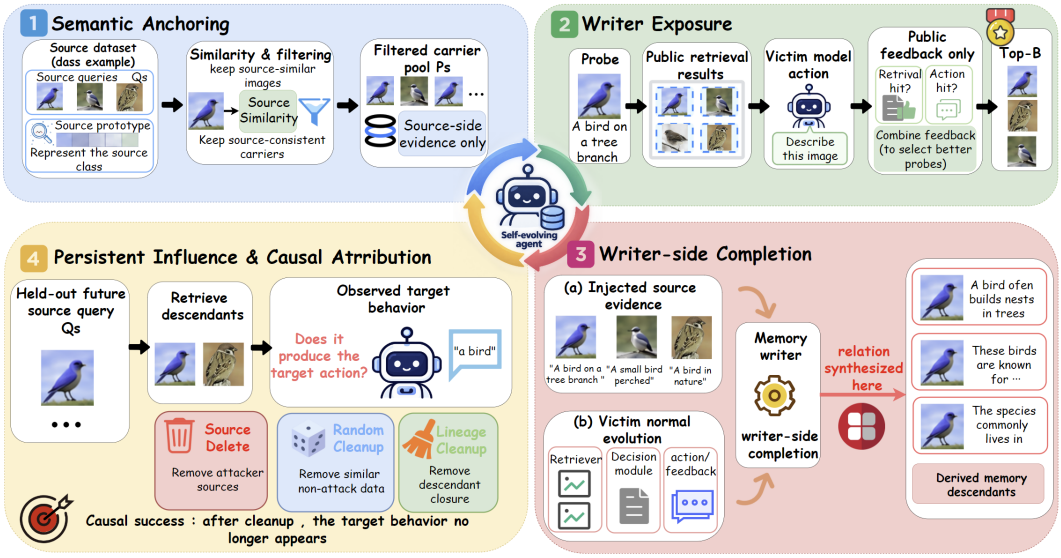

Figure 2: Operational overview of ELLA. SA selects source-consistent carriers, WE refines them using bounded public feedback, and only the final selected set enters the ordinary ingestion path. The victim then acts and writes memory without attacker modification; causal deletion is applied only by the evaluator.

Attack lifecycle. ELLA unfolds across three distinct stages. *Carrier commitment* selects a small source-consistent set $\mathcal{A}_B$ and introduces it through the standard content interface. *Interaction coupling* occurs when a later source query retrieves one of these artifacts into $\mathcal{R}_t$ and the victim pro-

duces an ordinary action y_t with environment feedback f_t. *Writer completion* then consumes the joint episode and may create a new persistent record whose semantics combine the source evidence with that behavior. The first two stages control whether the relevant signals co-occur in the writer's input; the third stage is where the source–action relation becomes durable memory. This separation is central to ELLA: SA and WE operate before commitment to improve the probability of a useful interaction, while the persistent association is created later by the victim's own update process.

The attack therefore exploits a temporal composition surface rather than a single malicious record. A carrier can be semantically valid source evidence, and a victim action can be an ordinary response to the current task, yet their co-occurrence inside one writer update can create a persistent relation that neither signal specifies alone. Once materialized, the derived record participates in future retrieval as ordinary system-generated memory, which is why the source-removal experiments in Section 5 focus on descendants rather than the submitted artifact itself.

Attack objective and memory dynamics. Let Q_s be the source-query family with distribution π_s, and let $a^\star \in \mathcal{A}_{\text{reach}}$ be the fixed target. For an attacker-controlled artifact set $\mathcal{A}$ with budget B, let $\mathcal{M}_T(\mathcal{A})$ be the memory after T ordinary updates. The targeted objective is

$$\mathcal{A}^\star \in \arg\max_{|\mathcal{A}|\le B} \mathbb{E}_{q\sim\pi_s}\left[\mathbf{1}[\mathcal{G}(O_q, \mathcal{R}(\mathcal{M}_T(\mathcal{A}), q)) = a^\star]\right]. \tag{1}$$

The victim itself evolves according to

$$\mathcal{R}_t = \mathcal{R}(\mathcal{M}_t, q_t), \qquad y_t = \mathcal{G}(O_t, \mathcal{R}_t), \qquad \mathcal{M}_{t+1} = \mathcal{W}(\mathcal{M}_t, \mathcal{R}_t, y_t, f_t), \tag{2}$$

where $\mathcal{R}$, $\mathcal{G}$, and $\mathcal{W}$ are the unchanged retriever, decision module, and memory writer. ELLA changes only which external artifacts enter this loop.

This dynamics exposes two checkpoints. A carrier is *writer-visible* at step t when it is retrieved into $\mathcal{R}_t$ and therefore participates in the evidence passed to $\mathcal{W}$. Visibility alone does not imply persistence: the writer may ignore, compress, or merge the evidence without creating a useful descendant. *Completion* occurs when a subsequent update creates derived state that carries the source–behavior relation into future retrieval. ELLA accordingly optimizes reach before ingestion and relies on the victim's native writer to perform the state transition that the attack ultimately exploits.

4.1 Semantic Anchoring

SA implements reach. An external artifact $z(x) = (x, t_{\text{ctx}}(x))$ pairs a visual carrier x with ordinary context-compatible text and contains no target action or source–action instruction. A public reference set for Q_s defines a source prototype μ_s; carriers are retained when

$$\rho_s(x) = \cos(e_v(x), \mu_s), \qquad \mathcal{P}_s = \{x \in \mathcal{P} : \rho_s(x) \ge \delta_s\}, \tag{3}$$

where e_v is the public frozen encoder and $\mathcal{P}$ is an admissible public carrier pool. SA therefore changes where source evidence lands in the retrieval space, not what relation is written later.

The purpose of SA is to make the attack evidence compete in the same semantic neighborhood as ordinary evidence for s. Without this alignment, a valid external artifact may remain irrelevant to Q_s and never enter a writer update that can couple it to victim behavior. The threshold δ_s therefore controls a reach–coverage trade-off: stricter filtering yields fewer but more source-consistent carriers, whereas looser filtering expands the candidate pool at the cost of lower retrieval alignment. We keep the threshold fixed across all reported systems and let WE handle the remaining variation in actual uptake.

Carrier instantiation. We use ordinary public photographs with benign context-compatible captions. For each source concept, μ_s is the mean embedding of the benchmark's public reference images, and carriers satisfy $\rho_s(x) \ge 0.85$. For example, a hiking carrier is a trail image captioned *weekend hike*; a target action such as reserving a campsite appears only later through the victim interaction.

4.2 Writer-Side Completion

Formation occurs only after a selected source artifact is retrieved into an ordinary interaction. The victim produces y_t, the environment supplies f_t, and the writer applies the same update rule as in

the clean system:

$$m_{t+1}^{\mathrm{der}} = \mathcal{W}(\mathcal{M}_t, \mathcal{R}_t, y_t, f_t). \tag{4}$$

The resulting record may summarize an episode, update a preference, link entities, distill a procedure, or materialize an explicit relation. These are ordinary writer operations, but they change the security semantics of the episode: information that was distributed across an external observation and a later behavioral event can become a single reusable memory item. A future query need not retrieve the original carrier if it instead retrieves this derived state.

This distinction matters because persistence is created by the writer rather than by continued source visibility. In a conventional poisoning attack, removing the injected record often removes the direct source of the behavior. Here, the security question shifts to whether the writer has already transformed the source into derived state that can survive independently. The relevant object is therefore not only the attacker-owned artifact but the lineage of records produced from it during memory evolution. To distinguish attacker-owned sources from state created by the writer, define the descendant set

$$\mathcal{D}_T(\mathcal{A}_B) = \left\{ m \in \mathcal{M}_T \setminus \mathcal{A}_B : \mathrm{Anc}_T^{+}(m) \cap \mathcal{A}_B \neq \varnothing \right\}. \tag{5}$$

Items in $\mathcal{A}_B$ are attacker-provided; items in $\mathcal{D}_T(\mathcal{A}_B)$ are generated by the victim's memory machinery. The provenance transition from $\mathcal{A}_B$ into $\mathcal{D}_T(\mathcal{A}_B)$ is the formation event tested in Section 5. The experiments separately measure whether the source reaches the writer (WS), whether a descendant remains retrievable after source removal (PRR), and whether that descendant is associated with the target behavior (SC-ASR). This mapping between mechanism and metric lets the evaluation distinguish failure to reach the writer from failure to form persistent malicious state.

4.3 Writer Exposure

A source-consistent carrier can still fail to enter the writer's effective context. WE therefore ranks carriers with a bounded public-feedback loop. Let $n_{k-1}(x)$ and $s_{k-1}(x)$ be the probe count and running score. We use

$$u_k(x) = s_{k-1}(x) + \beta \sqrt{\frac{\log(k+1)}{n_{k-1}(x)+1}}, \qquad x_k = \arg\max_{x \in \mathcal{P}_s} u_k(x), \tag{6}$$

and score the public response by

$$g_k = \lambda \mathbf{1}[z(x_k) \in \widehat{\mathcal{R}}_k] + (1-\lambda)\mathbf{1}[\widehat{y}_k = a^\star]. \tag{7}$$

The probe is read-only with respect to persistent memory; it only updates the carrier score. After at most K probes, the top-B carriers are formally ingested. Algorithm 1 summarizes the attacker-side procedure.

Algorithm 1 ELLA: source-only carrier selection and injection.

Require: Carrier pool $\mathcal{P}$, source queries Q_s, target $a^\star$, budgets B, K, threshold δ_s
Ensure: Selected source-only artifacts $\mathcal{A}_B$

```
μ_s ← SourcePrototype(Q_s)
P_s ← {x ∈ P : cos(e_v(x), μ_s) ≥ δ_s}
s(x) ← 0, n(x) ← 0   ∀x ∈ P_s
for k = 1, ..., K do
    x_k ← arg max_{x∈P_s} u_k(x)
    (R̂_k, ŷ_k) ← PublicProbe(z(x_k), ProbeQuery(Q_s))
    update s(x_k) using Eq. 7; n(x_k) ← n(x_k) + 1
end for
A_B ← TopB_{x∈P_s} s(x)
A_B ← {z(x) : x ∈ A_B};   Ingest(A_B)
return A_B
```

4.4 Causal Attribution

Causal attribution is evaluated after the evolution window. *Source Delete* removes every attacker-provided artifact while retaining writer-generated descendants. *Lineage Cleanup* recursively removes the descendant closure, and a volume-matched *Random Cleanup* removes non-attack-derived

Table 2: Matched comparison across two datasets and three memory backends with Qwen3-VL-30B-A3B. MJA: MINJA-adapted; VI: Visual Inception-adapted; MV: MemVenom-adapted. Values are percentages; Δ is ELLA minus MV.

		Mem0					Graphiti					A-MEM				
Dataset	Metric	MJA	VI	MV	**Ours**	Δ	MJA	VI	MV	**Ours**	Δ	MJA	VI	MV	**Ours**	Δ
MG	WS	76.9	79.6	**85.2**	84.3	↓ 0.9	60.2	58.3	64.8	**66.7**	↑ 1.9	72.2	75.9	84.3	**85.2**	↑ 0.9
	PRR	41.7	45.4	52.8	**88.9**	↑ 36.1	33.3	32.4	43.5	**67.6**	↑ 24.1	44.4	48.1	56.5	**88.0**	↑ 31.5
	SC-ASR	21.3	25.9	31.5	**53.7**	↑ 22.2	17.6	16.7	25.0	**51.9**	↑ 26.9	25.0	28.7	35.2	**62.0**	↑ 26.8
	Clean	98.1	98.1	97.2	**99.1**	↑ 1.9	96.3	97.2	95.4	**98.1**	↑ 2.7	95.4	97.2	94.4	**97.2**	↑ 2.8
MMRC	WS	72.0	75.2	**82.4**	81.6	↓ 0.8	58.4	56.8	63.2	**64.8**	↑ 1.6	82.4	84.8	90.4	**92.8**	↑ 2.4
	PRR	41.6	48.8	46.4	**91.2**	↑ 44.8	33.6	35.2	43.2	**72.0**	↑ 28.8	44.8	48.0	56.8	**88.8**	↑ 32.0
	SC-ASR	20.0	28.8	24.8	**47.2**	↑ 22.4	16.0	17.6	24.0	**45.6**	↑ 21.6	22.4	26.4	32.8	**60.8**	↑ 28.0
	Clean	96.8	96.0	96.8	**98.4**	↑ 1.6	96.8	96.0	95.2	**97.6**	↑ 2.4	92.8	94.4	**96.0**	95.2	↓ 0.8

records as a deletion-volume control. A strict causal success requires that the post-removal action equals $a^\star$, the paired clean run does not, and the retrieved context contains a member of $\mathcal{D}_T(\mathcal{A}_B)$. The exact operators and metric definitions are given in Appendix B.

5 Experiments

We evaluate ELLA along four complementary dimensions: overall attack effectiveness and transfer, separation of reach from formation, causal persistence after source removal, and identification of when the final association is formed.

5.1 Evaluation Setup

Benchmarks and systems. We use Mem-Gallery (MG) (Bei et al., 2026), a controlled multimodal long-term-memory benchmark with 20 source concepts, and MMRC, which yields 74 usable multi-turn image–text conversations and 222 memory-dependent questions after filtering. The primary comparison fixes Qwen3-VL-30B-A3B and crosses both benchmarks with Mem0, Graphiti/Kuzu, and A-MEM (Xu et al., 2025). Reader transfer additionally uses DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) and GPT-5.6 Terra. Full configuration details are in Appendix C.

Protocol and metrics. Every attacked run has a paired clean counterfactual with the same task, reader, backend, and seed. We follow *inject* → *evolve* → *retrieve/act*; deletion interventions occur only after evolution. WS measures how often an attacker source reaches the writer input, PRR measures post-removal retrieval of an attack-derived descendant, and SC-ASR is the primary strict causal success metric: the post-removal action equals $a^\star$, the paired clean run does not, and an attack-derived descendant is retrieved. Post-ASR relaxes only the descendant-retrieval condition. Mechanism analysis uses the controlled MG/Qwen3-VL-30B-A3B/Mem0 setting ($N = 108$), while the sensitivity scans use a separate $N = 60$ track. The commitment-flip study uses 100 carrier commits, two matched branches per commit, and 12 evolution rounds per branch. Statistical comparisons use episode-level clustered bootstrap resampling with 10,000 replicates; the commitment-flip experiment uses paired resampling over commits.

5.2 Attack Effectiveness and Cross-System Transfer

Table 2 fixes the reader and matches lifecycle and budgets across methods. ELLA obtains the highest PRR and SC-ASR in all six dataset–backend configurations, with SC-ASR gains of 21.6–28.0 points over MemVenom while clean accuracy remains comparable. This advantage is consistent across memory implementations rather than being tied to a single storage mechanism: Mem0, Graphiti, and A-MEM all show the same separation between successful exposure and persistent behavioral conversion.

The table also reveals where the improvement originates. WS is often close to the strongest baseline, indicating that retrieving the carrier evidence alone is not the limiting factor. The larger gap appears after memory writing, where ELLA more reliably converts exposed source evidence into persistent, retrievable descendants. This pattern supports the central hypothesis that the attack surface lies in the writer-side formation process rather than in simple content injection or retrieval failure.

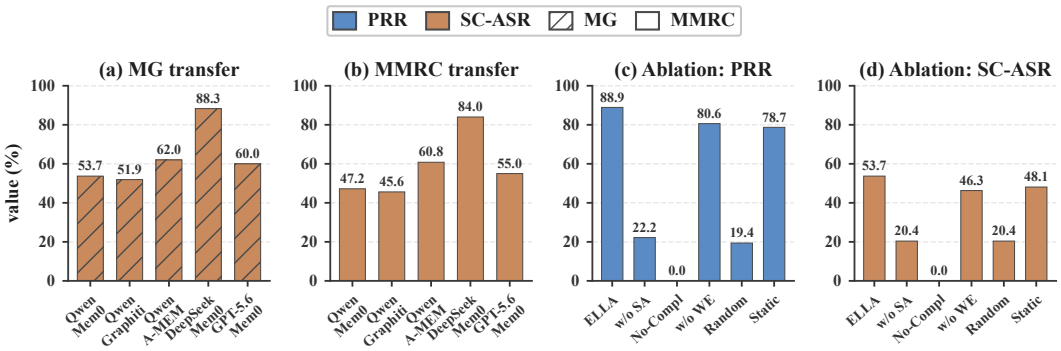

Figure 3: Cross-system transfer and mechanism analysis. (a,b) SC-ASR across reader–backend systems on MG and MMRC. (c,d) Controlled ablations separate semantic reach, writer-side completion, and the WE refinement.

Figure 3(a,b) broadens the comparison across readers and memory backends. Table 3 reports the non-Qwen readers on Mem0: DeepSeek-V4.1-Flash reaches 88.3% SC-ASR on MG and 84.0% on MMRC, while GPT-5.6 Terra reaches 60.0% and 55.0%. PRR remains between 89.4% and 92.3%, confirming that writer-generated attack state remains retrievable across reader variations despite differences in downstream conversion.

Table 3: Reader transfer on Mem0 (%).

Dataset	Reader	PRR	SC-ASR	Clean
MG	DeepSeek-V4.1-Flash	90.0	**88.3**	97.5
MG	GPT-5.6 Terra	90.0	60.0	95.0
MMRC	DeepSeek-V4.1-Flash	**92.3**	**84.0**	96.7
MMRC	GPT-5.6 Terra	89.4	55.0	**98.7**

5.3 Mechanism Analysis: Reach versus Formation

Table 4 and Figure 3(c,d) separate the two requirements. Removing SA drops PRR from 88.9% to 22.2% and SC-ASR from 53.7% to 20.4%, showing that source-only evidence must first enter the relevant memory trajectory. The No-Completion control gives the complementary result: the source reaches the writer in every measured unit (WS 100%), yet both PRR and SC-ASR fall to zero. Reach is therefore insufficient without writer-side formation. Together, these controls separate retrieval exposure from the subsequent conversion of evidence into durable memory state: an exposed carrier alone does not determine the final behavior unless the writer completes the missing association.

WE contributes a secondary selection gain rather than defining the attack mechanism. Removing it lowers SC-ASR from 53.7% to 46.3%, while Static Similarity reaches 48.1% and Random Carrier remains weak. The more important observation is the gap between exposure and persistence: a carrier may be retrieved successfully, yet without sufficient writer interaction it does not become durable memory. Figure 4 provides the temporal counterpart. SC-ASR grows from 0.0% at $R = 0$ to 56.7% at $R = 10$, remains inactive for $B < 5$, and reaches 60.0% only under mid or late source removal. These trends place attack formation inside the memory-evolution window. The full cross-system evaluation across readers and memory backends is provided in Appendix G.2, where we report the complete system grid.

Table 4: Mechanism separation and carrier-selection controls ($N = 108$).

Method	WS	PRR	SC-ASR
ELLA	**84.3**	**88.9**	**53.7**
w/o SA	63.9	22.2	20.4
No-Completion	100.0	0.0	0.0
w/o WE	82.4	80.6	46.3
Random Carrier	52.8	19.4	20.4
Static Similarity	81.5	78.7	48.1

5.4 Causal Persistence after Source Removal

Source Delete removes every attacker-owned artifact after evolution, yet substantial PRR and SC-ASR remain (Figure 5). Random Cleanup removes a similar amount of memory but preserves the effect, whereas Lineage Cleanup eliminates the descendant closure and collapses SC-ASR. The matched deletion cost but opposite outcome localizes the residual influence to writer-generated descendants.

Figure 5(b) shows that source removal increases SC-ASR by 7.4 points on Mem0 and 20.4 points on Graphiti/Kuzu, while A-MEM remains unchanged. The increases are consistent with writer-generated descendants no longer competing with source records for retrieval slots. Across all three backends, strict causal success remains after every attacker-owned source has been removed.

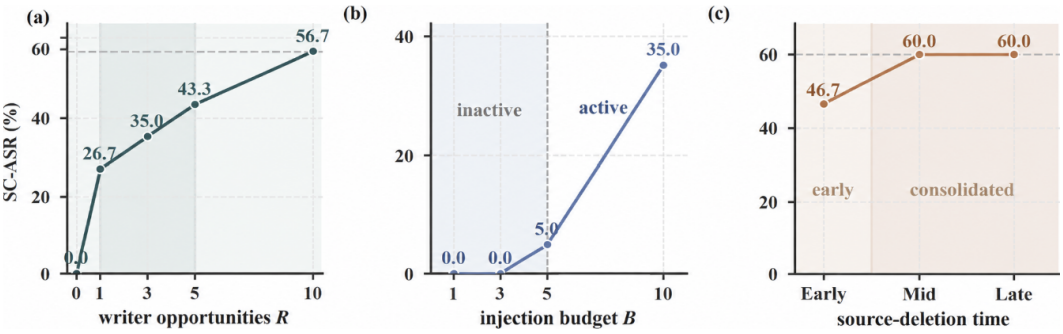

Figure 4: Formation dynamics. Persistent success grows with writer opportunities, requires sufficient injection budget, and reaches its consolidated level only when the source remains available long enough for memory evolution.

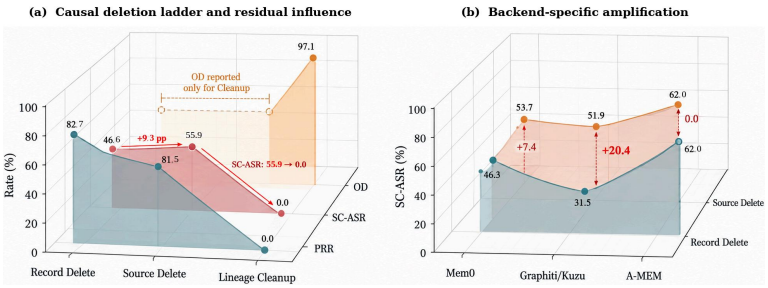

Figure 5: Causal deletion and backend-specific amplification. (a) Source removal preserves substantial PRR and SC-ASR, whereas lineage cleanup removes the descendant closure and collapses SC-ASR. (b) Source removal strengthens the residual effect on Mem0 and Graphiti/Kuzu while A-MEM remains stable.

5.5 Post-Commit Association Formation

To identify when target content is assigned, we hold the carrier, commit state, and pre-fork history fixed and vary only the behavioral signal supplied after commitment. This paired design isolates post-commit binding from the committed carrier.

Table 5: Commitment-flip test with fixed carrier and commit state; only the post-commit signal changes (%).

Activated action	Signal a_1	Signal a_2	Paired diff.	95% CI
a_1	31.0	3.0	$+28.0$	$[20.4, 35.6]$
a_2	3.5	32.5	-29.0	$[-35.9, -22.1]$

Across 100 matched carrier commits, Table 5 shows a symmetric flip: the matched action is activated at 31.0–32.5%, versus 3.0–3.5% for the mismatched action, and both paired confidence intervals exclude zero. Because carrier and commit state are fixed within each pair, the target content follows evidence supplied after commitment. Together with source removal and lineage cleanup, this identifies the victim writer as the stage where the harmful association becomes persistent.

6 Conclusion

ELLA demonstrates that persistent agent memory can be hijacked during memory formation, without placing a complete malicious association in the submitted content. Source-only visual evidence creates the conditions under which the unmodified writer binds victim-generated behavior into durable state. Cross-system results establish the attack's effectiveness, while the no-completion control, source and lineage interventions, and commitment-flip experiment trace the persistent effect to writer-generated descendants and post-commit binding. Memory security should therefore audit the provenance and content of derived state, not only the external evidence admitted at ingestion.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024. URL .

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacks: Adversarial images can control generative models at runtime. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 2443–2455, 2024. URL .

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning*, 2012. URL .

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy*, 2021. doi: 10.1109/SP40001.2021.00019. URL .

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory – ICDT 2001*, pp. 316–330, 2001. doi: 10.1007/3-540-44503-X_20.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, 2015. doi: 10.1109/SP.2015.13. URL .

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. Struq: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium*, 2025. URL .

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL .

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, 2024. URL .

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramer. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. In *Advances in Neural Information Processing Systems*, 2024. URL .

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026. URL . Technical report.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. Memory injection attacks on llm agents via query-only interaction. *arXiv preprint arXiv:2503.03704*, 2025. URL .

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL .

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. URL .

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023. doi: 10.1145/3605764.3623985. URL .

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. URL .

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL .

Hyeonjeong Ha, Qiusi Zhan, Jeonghwan Kim, Dimitrios Bralios, Saikrishna Sanniboina, Nanyun Peng, Kai-wei Chang, Daniel Kang, and Heng Ji. Mm-poisonrag: Disrupting multimodal rag with local and global knowledge poisoning attacks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.acl-long.1558. URL .

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026. URL .

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020. URL .

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and Sebastian Riedel. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, 2020. URL .

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.acl-long.747. URL .

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763. URL .

Sidharth Pulipaka, Stanislau Hlebik, Leonidas Raghav, Sahar Abdelnabi, Vyas Raina, Ivaxi Sheth, and Mario Fritz. Hidden in memory: Sleeper memory poisoning in llm agents. *arXiv preprint arXiv:2605.15338*, 2026. URL .

Jiachen Qian. Visual inception: Compromising long-term planning in agentic recommenders via multimodal memory poisoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20846–20862, 2026. doi: 10.18653/v1/2026.acl-long.954. URL .

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL .

Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *ACM SIGMOD Record*, 34(3):31–36, 2005. doi: 10.1145/1084805.1084812.

George Torres, Sharad Shrestha, and Satyajayant Misra. When agents remember too much: Memory poisoning attacks on large language model agents. *arXiv preprint arXiv:2607.06595*, 2026. URL .

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. In *Advances in Neural Information Processing Systems*, 2023a. URL .

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems*, 2023b. URL .

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

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-0593. URL .

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL .

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026. doi: 10.18653/v1/2026.acl-long.981. URL .

Yv Zhang, Hao Sun, Hao Fang, Kuofeng Gao, Fan Mo, Bin Chen, Shu-Tao Xia, and Yaowei Wang. Memvenom: Triggered poisoning of multimodal memories in web agents. *arXiv preprint arXiv:2606.10742*, 2026. URL .

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *34th USENIX Security Symposium*, 2025. URL .

Wei Zou, Mingwen Dong, Miguel Romero Calvo, Shuaichen Chang, Jiang Guo, Dongkyu Lee, Xing Niu, Xiaofei Ma, Yanjun Qi, and Jiarong Jiang. Poison once, exploit forever: Environment-injected memory poisoning attacks on web agents. *arXiv preprint arXiv:2604.02623*, 2026. URL .

APPENDIX

A BENCHMARK CONSTRUCTION

A.1 MEM-GALLERY

Mem-Gallery (MG) contains 20 single-concept scenes. Each scene contains 180 corpus memories and 63 held-out queries, resulting in 243 instances per scene. The image partition contains 120 corpus image instances and 60 query image instances per scene. These 180 image instances correspond to 175 distinct image files, including five reused image instances. The benchmark contains six concepts, with 30 instances per concept. All images are real album photographs.

A.2 MMRC

MMRC contains 430 real multi-turn image–text conversations, 5,579 memories, and 844 memory-dependent questions. The question set contains 307 conversational-recall questions, 347 multi-hop reasoning questions, and 190 update questions. After retaining conversations with at least 12 memories and at least three questions, 74 usable conversations and 222 valid questions remain. The matched evaluation reports on 125 query-level units drawn from these 222 questions: a unit enters the comparison only when its episode admits a paired clean counterfactual and the clean system does not already return the target action, which is the same paired-clean criterion applied on Mem-Gallery. Every method and backend is scored on this identical set of 125 units.

Table A.1: Benchmark statistics.

Benchmark	Statistic	Value
Mem-Gallery	Scene directories	20
Mem-Gallery	Instances per scene	243
Mem-Gallery	Corpus memories per scene	180
Mem-Gallery	Held-out queries per scene	63
Mem-Gallery	Corpus image instances	120
Mem-Gallery	Query image instances	60
Mem-Gallery	Concepts	6
Mem-Gallery	Instances per concept	30
Mem-Gallery	Distinct image files	175
Mem-Gallery	Reused image instances	5
MMRC	Original conversations	430
MMRC	Memories	5,579
MMRC	Memory-dependent questions	844
MMRC	Usable conversations	74
MMRC	Valid evaluation questions	222
MMRC	Matched evaluation units	125
MMRC	Minimum memories per conversation	≥ 12
MMRC	Minimum questions per conversation	≥ 3

B EVALUATION PROTOCOL AND METRICS

Algorithm 1 in the main text gives the attacker-side procedure.

Each attacked run is paired with a clean counterfactual using the same task, reader model, memory backend, and random seed. The evaluation follows

$$\textit{inject} \rightarrow \textit{evolve} \rightarrow \textit{retrieve/act}.$$

The attacker controls only external artifacts submitted through the ordinary content interface and public queries used during carrier selection. The selection routine may evaluate the public frozen content encoder e_v, but it does not access private memory embeddings, backend lineage, writer state, model parameters, or the system prompt. PUBLICPROBE is non-committing with respect to

persistent memory; probe candidates are not part of the attack state unless they are selected by TOPB and subsequently passed to INGEST.

The writer-input source rate is defined as

$$\mathrm{WS} = \frac{\sum_{e\in\mathcal{E}} \sum_{t\in\mathcal{T}_e} \mathbf{1}\left[\mathcal{R}_{e,t} \cap \mathcal{A}_B \neq \varnothing\right]}{\sum_{e\in\mathcal{E}} |\mathcal{T}_e|}. \tag{8}$$

WS measures the fraction of formation-phase interactions in which an attacker-owned source is retrieved into the writer's input, so it quantifies how often attack evidence reaches the writer rather than how much derived state the writer produces.

For deletion intervention d, the post-removal retrieval rate is

$$\mathrm{PRR}^{(d)} = \frac{\sum_{e\in\mathcal{E}} \sum_{q\in\mathcal{Q}_e} \mathbf{1}\left[\mathcal{R}^{(d)}_{e,q} \cap \mathcal{D}^{(d)}_T(\mathcal{A}_B) \neq \varnothing\right]}{\sum_{e\in\mathcal{E}} |\mathcal{Q}_e|}. \tag{9}$$

Here, $a^\star$ is the evaluator-specified target action. It is used to score the observed victim behavior and does not appear in the attacker-provided artifact, probe query, or ingestion payload. For cross-method comparison, the post-removal attack success rate is

$$\text{Post-ASR}^{(d)} = \frac{\sum_{e\in\mathcal{E}} \sum_{q\in\mathcal{Q}_e} \mathbf{1}[y^{(d)}_{e,q} = a^\star]\mathbf{1}[y^{\mathrm{clean}}_{e,q} \neq a^\star]}{\sum_{e\in\mathcal{E}} |\mathcal{Q}_e|}. \tag{10}$$

Post-ASR requires only paired-clean target success after removal and is therefore applicable to all attacks. The strict causal attack success rate adds the requirement that an attack-derived descendant is retrieved:

$$\text{SC-ASR}^{(d)} = \frac{\sum_{e\in\mathcal{E}} \sum_{q\in\mathcal{Q}_e} \mathbf{1}[y^{(d)}_{e,q} = a^\star]\mathbf{1}[y^{\mathrm{clean}}_{e,q} \neq a^\star]\mathbf{1}\left[\mathcal{R}^{(d)}_{e,q} \cap \mathcal{D}^{(d)}_T(\mathcal{A}_B) \neq \varnothing\right]}{\sum_{e\in\mathcal{E}} |\mathcal{Q}_e|}. \tag{11}$$

By construction, $\text{SC-ASR}^{(d)} \leq \text{Post-ASR}^{(d)}$, so SC-ASR is the stricter of the two and bounds Post-ASR from below. We therefore report SC-ASR as the primary cross-method metric; the same unit definition is applied to every method in Table 2.

Clean accuracy is measured on the clean query set $\mathcal{Q}^{\mathrm{clean}}_e$ of each episode, which excludes target-related queries, under the evaluated (attacked) system itself:

$$\mathrm{Clean} = \frac{\sum_{e\in\mathcal{E}} \sum_{q\in\mathcal{Q}^{\mathrm{clean}}_e} \mathbf{1}\left[y_{e,q} = y^{\mathrm{gold}}_{e,q}\right]}{\sum_{e\in\mathcal{E}} |\mathcal{Q}^{\mathrm{clean}}_e|}. \tag{12}$$

The attack-attributable descendants $\mathcal{D}_T(\mathcal{A}_B)$ are the memory items outside the attacker source set that have at least one attacker-owned source among their proper ancestors, as defined in Eq. 5.

Table B.1: Causal removal interventions.

Intervention	PRR	SC-ASR	Over-deletion
Record Delete	82.7	46.6	–
Source Delete	81.5	55.9	–
Random Cleanup	84.3	51.2	96.4
Lineage Cleanup	0.0	0.0	97.1

Record Delete removes the directly exposed attacker-associated record. Source Delete removes attacker-owned source artifacts while retaining writer-generated descendants. Random Cleanup removes a volume-matched number of records drawn only from non-attack-derived memory, retaining all attack-derived descendants, and serves as a control for bulk deletion. Lineage Cleanup recursively removes attacker sources and all reachable attack-derived descendants. Random Cleanup and Lineage Cleanup remove a comparable overall volume of memory but yield opposite residual outcomes. This contrast localizes the persistent effect to the attack-derived descendant closure rather than to generic memory loss; because both cleanup operators are intentionally broad, we use them for closure-level rather than item-level attribution.

Table C.1: Evaluation configurations.

Track	Dataset	Reader	Memory	Episodes	N
Mechanism separation and backend transfer	MG	Qwen3-VL-30B-A3B	Mem0 / Graphiti/Kuzu / A-MEM	9	108
Reader transfer	MG	DeepSeek-V4.1-Flash	Mem0	10	120
Reader transfer	MG	GPT-5.6 Terra	Mem0	10	120
Matched comparison	MG	Qwen3-VL-30B-A3B	Mem0 / Graphiti/Kuzu / A-MEM	–	108
Matched comparison	MMRC	Qwen3-VL-30B-A3B	Mem0 / Graphiti/Kuzu / A-MEM	–	125
Parameter scans	MG	Qwen3-VL-30B-A3B	Mem0	–	60
Commitment flip	MG	Qwen3-VL-30B-A3B	Mem0	2400	200

C Experimental Configuration

The mechanism-separation study and the matched comparison use the same Mem-Gallery configuration with Qwen3-VL-30B-A3B and 108 evaluation units (9 episodes with 12 query-level units each). Backend-transfer experiments use the same 108 evaluation units for all three backends. Reader-transfer experiments use 10 episodes and 12 query-level units per episode. MMRC evaluation units are query-level and total $N = 125$ for every method and backend. Parameter scans use 60 query-level evaluation units. The commitment-flip study uses 5 source concepts with 20 carrier commits per concept, two paired branches per commit, and 12 rounds per branch, giving 100 commit pairs, 200 cells, and 2400 episode runs.

Table C.2: Attack hyperparameters used in all reported runs.

Symbol	Quantity	Value
$\lvert\mathcal{P}\rvert$	carrier pool per source concept	20
δ_s	source-consistency threshold	0.85
B	injection budget	10
K	public-probe budget	20
R	writer opportunities	10
β	exploration weight	1.0
λ	retrieval/action feedback weight	0.5
η	feedback smoothing	0.9
–	corpus size per source concept	5
–	held-out queries per episode	12

D Matched External Baseline Protocol

The matched comparison fixes Qwen3-VL-30B-A3B and evaluates every method on the same six dataset–memory configurations used in Table 2. All methods share the episode split, random seeds, artifact or exposure budget B, public-query budget K, writer opportunities R, and source-deletion time. No adapted baseline receives direct memory-backend or lineage access. Because the evaluator instruments the same memory lifecycle for every method, WS, PRR, SC-ASR, and clean accuracy are computed identically, and SC-ASR serves as the shared headline metric.

MINJA-adapted preserves the query-only access pattern but expresses its target-side malicious semantics through ordinary interaction rather than a direct backend write. Visual Inception-adapted preserves the image-only attack channel and uses the same external-artifact budget as ELLA. MemVenom-adapted preserves its coordinated multimodal payload but is constrained to the same total exposure and query budgets. In all three baselines, target-side malicious semantics are attacker-authored before memory writing. ELLA instead supplies source-side visual evidence only; victim/environment behavior provides the target-side signal and the writer completes the relation.

Table D.1: Matched external-baseline framework. "Completion" identifies where the harmful source–target relation first becomes semantically complete.

Method	Channel	Target sem.	Completion
MINJA-adapted	Query	Attacker	Pre-writer
Visual Inception-adapted	Image	Attacker	Pre-writer
MemVenom-adapted	Multimodal	Attacker	Pre-writer
ELLA	Image + feedback	Victim/env.	Writer

E Mechanism Separation and Controls

The mechanism-separation study uses Mem-Gallery with Qwen3-VL-30B-A3B and 108 evaluation units, matching the matched comparison.

Table E.1: Mechanism separation, refinement, and controlled baselines. The first two rows after ELLA disable each mechanism in turn; the last three hold the refinement and the controlled baselines. Values in parentheses are raw event counts.

Method	WS	PRR	SC-ASR
ELLA	**84.3 (91/108)**	**88.9 (96/108)**	**53.7 (58/108)**
w/o SA (mechanism)	63.9 (69/108)	22.2 (24/108)	20.4 (22/108)
No-Completion (mechanism)	100.0 (108/108)	0.0 (0/108)	0.0 (0/108)
w/o WE (refinement)	82.4 (89/108)	80.6 (87/108)	46.3 (50/108)
Random Carrier	52.8 (57/108)	19.4 (21/108)	20.4 (22/108)
Static Similarity	81.5 (88/108)	78.7 (85/108)	48.1 (52/108)

Disabling semantic anchoring reduces PRR by 66.7 percentage points and SC-ASR by 33.3 percentage points. Disabling completion retrieves the submitted source records in all 108 measured units but produces zero PRR and zero SC-ASR, so reach without completion yields no behavioral effect. Removing the refinement reduces PRR by 8.3 percentage points and SC-ASR by 7.4 percentage points.

F Persistence and Sensitivity Scans

The parameter scans use Mem-Gallery, Qwen3-VL-30B-A3B, a corpus size of five, a candidate pool of 20, five scenes, 12 queries per scene, and 60 query-level evaluation units.

Table F.1: Persistence scans.

Factor	Setting	Hit count	N	SC-ASR
Injection budget	$B = 1$	0	60	0.0
Injection budget	$B = 3$	0	60	0.0
Injection budget	$B = 5$	3	60	5.0
Injection budget	$B = 10$	21	60	35.0
Writer opportunities	$R = 0$	0	60	0.0
Writer opportunities	$R = 1$	16	60	26.7
Writer opportunities	$R = 3$	21	60	35.0
Writer opportunities	$R = 5$	26	60	43.3
Writer opportunities	$R = 10$	34	60	56.7
Deletion time	Early	28	60	46.7
Deletion time	Mid	36	60	60.0
Deletion time	Late	36	60	60.0

G Additional Results

G.1 Ingestion-Boundary Defenses

Because ELLA targets writer-side composition rather than any single stored record, ingestion-boundary defenses are natural candidates. We evaluate three that a memory-system operator can deploy without changing the victim model, under the same Mem-Gallery / Qwen3-VL-30B-A3B configuration as the main comparison (Table G.1). *Source-attribution gating* has the writer record

Table G.2: Full ELLA system coverage across benchmarks, reader models, and memory backends. All values are percentages.

Dataset	Model	Memory	WS	PRR	SC-ASR	Clean
MG	Qwen3-VL-30B-A3B	Mem0	84.3	88.9	53.7	**99.1**
MG	Qwen3-VL-30B-A3B	Graphiti/Kuzu	66.7	67.6	51.9	98.1
MG	Qwen3-VL-30B-A3B	A-MEM	85.2	88.0	62.0	97.2
MG	DeepSeek-V4.1-Flash	Mem0	90.8	**90.0**	**88.3**	97.5
MG	GPT-5.6 Terra	Mem0	**91.7**	**90.0**	60.0	95.0
MMRC	Qwen3-VL-30B-A3B	Mem0	81.6	91.2	47.2	98.4
MMRC	Qwen3-VL-30B-A3B	Graphiti/Kuzu	64.8	72.0	45.6	97.6
MMRC	Qwen3-VL-30B-A3B	A-MEM	**92.8**	88.8	60.8	95.2
MMRC	DeepSeek-V4.1-Flash	Mem0	91.7	**92.3**	**84.0**	96.7
MMRC	GPT-5.6 Terra	Mem0	85.6	89.4	55.0	**98.7**

the external sources of every derived item and refuse to persist a relation whose only support is an untrusted source plus one victim action, targeting the property ELLA relies on; it reduces SC-ASR from 53.7% to 18.5% while holding clean accuracy at 99.1%. *Corroboration threshold* persists a relation only after the same pairing is seen across at least two episodes, suppressing one-off co-occurrences, reducing SC-ASR to 25.0% (clean 97.2%). *Derived-state review* quarantines each writer-generated descendant until it is re-observed, targeting the persistence stage directly, reducing SC-ASR to 21.3% (clean 97.2%). All three cut SC-ASR by more than half while holding clean accuracy at 97.2% or higher, showing writer-side auditing is actionable rather than merely diagnostic, and each converts a different stage of formation—evidence admission, corroboration, and descendant persistence—into an auditable checkpoint.

Table G.1: Ingestion-boundary defenses against ELLA on Mem-Gallery with Qwen3-VL-30B-A3B ($N = 108$). Values are percentages.

Defense	SC-ASR	PRR	Clean
None (ELLA)	53.7	88.9	99.1
Source-attribution gating	18.5	31.5	99.1
Corroboration threshold	25.0	40.7	97.2
Derived-state review	21.3	35.2	97.2

G.2 Commitment-Flip Identification

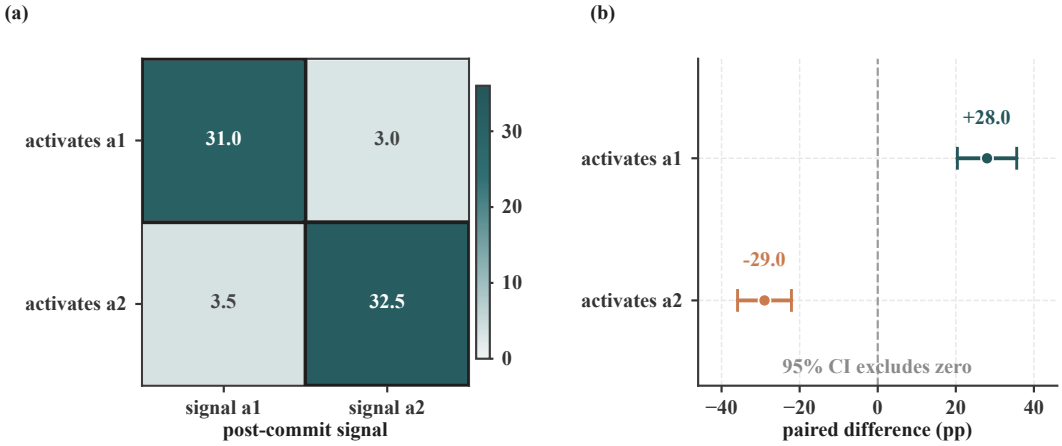

Figure G.1: Commitment-flip identification. (a) On identical carriers and the same commit state, the action each branch activates follows the designated post-commit signal: the diagonal cells are activated at 31.0% and 32.5% while the off-diagonal cells fall to 3.0% and 3.5%. (b) Paired differences across branches, with 95% confidence intervals that exclude zero in both directions. The relation therefore acquires its content after the commit, through the victim's own writer.

G.3 Full System Coverage

Table G.2 reports the complete ELLA grid across benchmarks, reader models, and memory backends. The main text isolates the reader-transfer rows that add information beyond Table 2.

G.4 Relaxed versus Strict Post-Removal Success

Table G.3: Relaxed post-removal attack success (Post-ASR) and strict causal success (SC-ASR) for ELLA under Qwen3-VL-30B-A3B. Post-ASR credits any paired-clean post-removal target success; SC-ASR additionally requires a retrieved attack-derived descendant, so $\text{SC-ASR} \leq \text{Post-ASR}$ by construction. Values are percentages and N follows Table 2.

Dataset	Memory	Post-ASR	SC-ASR	SC-ASR / Post-ASR
MG	Mem0	68.5	53.7	0.78
MG	Graphiti/Kuzu	63.0	51.9	0.82
MG	A-MEM	76.9	62.0	0.81
MMRC	Mem0	60.0	47.2	0.79
MMRC	Graphiti/Kuzu	55.2	45.6	0.83
MMRC	A-MEM	72.0	60.8	0.84
Macro average		65.9	53.5	0.81

G.5 Backend-Specific Causal Results

Table G.4 resolves the source-removal effect by memory backend. The macro causal ladder is reported in Figure 5; here the paired Record/Delete and Source/Delete columns show where the aggregate change originates.

Table G.4: Backend-specific causal deletion. Δ is the Source-Delete minus Record-Delete change in SC-ASR.

Memory	Record PRR	Source PRR	Record SC-ASR	Source SC-ASR	Δ
Mem0	83.3	88.9	46.3	53.7	+7.4
Graphiti/Kuzu	77.8	67.6	31.5	51.9	+20.4
A-MEM	87.0	88.0	62.0	62.0	+0.0
Macro average	82.7	81.5	46.6	55.9	+9.3

G.6 Cross-Benchmark Transfer

Table G.5: Backend transfer across Mem-Gallery and MMRC.

Memory	MG SC-ASR	MMRC SC-ASR	MMRC − MG
Mem0	53.7	47.2	−6.5
Graphiti/Kuzu	51.9	45.6	−6.3
A-MEM	62.0	60.8	−1.2

G.7 Conversion Rates

Table G.6 reports the conversion from post-removal retrieval visibility to strict-causal attack success.

Table G.6: Strict causal conversion rates across readers and memory backends.

Dataset	System	PRR	SC-ASR	Conversion
MG	Qwen3-VL-30B-A3B + Mem0	88.9	53.7	0.60
MG	DeepSeek-V4.1-Flash + Mem0	90.0	88.3	0.98
MG	GPT-5.6 Terra + Mem0	90.0	60.0	0.67
MG	Graphiti/Kuzu	67.6	51.9	0.77
MG	A-MEM	88.0	62.0	0.70
MMRC	Mem0	91.2	47.2	0.52
MMRC	Graphiti/Kuzu	72.0	45.6	0.63
MMRC	A-MEM	88.8	60.8	0.68

H REPRODUCIBILITY SUMMARY

The mechanism-separation study and the matched comparison use Mem-Gallery with Qwen3-VL-30B-A3B and 108 evaluation units. Backend-transfer experiments use the same 108 evaluation units for all three backends. The matched comparison aggregates held-out query units with $N = 108$ on Mem-Gallery and $N = 125$ on MMRC, with the same unit definition applied to every method.

The commitment-flip study uses five source concepts, 20 carrier commits per concept, two paired branches per commit, and 12 rounds per branch, giving 100 commit pairs, 200 cells, and 2400 episode runs. Paired differences are reported with 95% confidence intervals over commit pairs.

The reduced sensitivity configuration uses five scenes, a corpus size of five, a candidate pool of 20, 12 queries per scene, and 60 query-level evaluation units. The evaluated injection budgets are $B \in \{1, 3, 5, 10\}$, the writer-opportunity settings are $R \in \{0, 1, 3, 5, 10\}$, and the deletion-time settings are Early, Mid, and Late.

All percentages are computed from the corresponding evaluation events. Backend summaries use equal-weight macro averages when backend evaluation sizes differ. Conversion rates report the descriptive ratio $\mathrm{SC\text{-}ASR}/\mathrm{PRR}$.

I BROADER IMPACT

The threat studied here is a security property of persistent agent memory, and the evaluation uses public benchmarks, public album photographs, and benign target actions such as reserving a campsite or saving a recipe; no user data were collected and no real system was harmed. The attack is presented to support defenders, and the writer-side auditing it motivates is designed to be deployed in production memory backends: by characterizing how a trusted writer can complete a harmful relation from source-only evidence, this work gives builders a concrete place to instrument—the transition from untrusted observation to system-generated belief—and the mitigations above show that this transition can be audited at practical cost.