CREDIT WHERE CREDIT ISN'T DUE: PRICING REPUTATION IN MULTI-AGENT SHARED MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Shared memory enables multi-agent systems to reuse information across tasks and learn source reputation from task feedback. However, when a reader agent answers correctly from its own knowledge even though the facts it retrieved are false, the success rewards the writer of those facts and raises its reputation. Evidence accumulated while a writer was reliable also outlives a change in its behavior, so a writer that turns unreliable keeps its reputation. Our experiments show that when writers turn from correct to false facts, retrieval ranked by reputation learned from raw task outcomes returns a correct fact on only 20.4% of reads, less than half the rate of random selection. To address these problems, we propose Priced Reputation Learning (PRL) for shared knowledge graph memory. PRL credits the writers of a successful task only in proportion to the probability that the same reader would have failed with no memory. PRL further adds a forgetting mechanism so that reputation can follow writers that change. Across six memory conditions and twelve readers from five model families, PRL selects correct facts where reputation learned from raw outcomes serves almost none, and reaches the writer oracle to within reader variation.

1 INTRODUCTION

Multi-agent systems coordinate multiple large language model (LLM) agents to solve tasks that involve several steps or repeated interactions (Qian et al., 2025). In sequential workflows, information produced by one agent or during an earlier task may be useful to other agents later. Persistent shared memory allows agents to store and retrieve such information across task boundaries (Park et al., 2023; Shinn et al., 2023; Packer et al., 2023; Zhang et al., 2025). However, when multiple agents can write to the same memory, the stored information may become inconsistent or incorrect. Multiple writer agents may store different values for the same relation, leaving a reader agent to decide which version to use. Because each stored fact retains its writer provenance, a natural approach is to estimate writer reputation from downstream task feedback and use it to rank competing facts. After the reader completes a task, the writers whose facts were served receive credit according to the task outcome, and future retrieval favors writers with higher reputation. This creates a feedback loop in which source reputation determines what the reader sees, while the reader's outcome determines how source reputation changes.

Task success, however, is only an indirect measure of memory quality. A reader may retrieve a false fact but still answer correctly from its parametric knowledge (Figure 1a). If this successful outcome is credited to the served writer, the source of the false fact gains reputation and becomes more likely to be selected again. Past outcomes can also become stale when writer behavior changes (Figure 1b). A writer may contribute correct facts during an initial period and false facts later, while the positive evidence accumulated during the correct period remains in its reputation. The writer can therefore continue to outrank more reliable sources after its behavior changes. Reputation learning must therefore account for what the reader can solve without memory and for changes in writer behavior.

To address these problems, we propose **Priced Reputation Learning (PRL)** for shared knowledge graph memory. We organize the memory as a knowledge graph in which each stored fact is a subject, predicate, and object triple tagged with its writer. Competing values for the same relation are kept side by side rather than overwritten, and the choice among them is made at retrieval time

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: **Two failure modes of outcome-based writer reputation. (a)** The reader retrieves a false fact but answers correctly from its own knowledge, and the credited writer gains reputation. **(b)** A writer turns from correct to false contributions, but its earlier positive evidence keeps its reputation.

from writer reputation. PRL introduces a reader-conditioned credit-pricing scheme that evaluates a successful task against the same reader's closed-book performance. When the reader is likely to answer correctly without memory, the writers whose facts were served receive less positive credit; a failed task retains full negative credit. PRL combines this scheme with a forgetting rule that discounts accumulated evidence toward a common prior before each update, allowing reputation to follow changes in writer behavior. The resulting reputation scores are used to rank conflicting graph facts during retrieval. Figure 2 summarizes the resulting retrieval and reputation-update pipeline.

In summary, we make three contributions. First, we identify two failure modes of reputation learned from task outcomes in shared memory, a leak from reader knowledge and stale evidence after a writer changes, and give conditions under which each occurs. Second, we propose PRL, which prices credit against a per task closed-book baseline of the same reader and lets reputation evidence expire. Third, across six memory conditions and twelve readers, we show that PRL serves correct facts where unpriced credit serves almost none and matches a writer oracle to within reader variation, withholds the credit a writer farms on facts the reader can verify alone, and matches or exceeds existing source-selection methods wherever false versions outnumber correct ones.

2 Related Work

Memory in multi-agent systems. Persistent memory allows LLM agents to retain information beyond a single task and use past experience in later decisions. Retrieval and reflection support this reuse by making earlier observations and lessons available during subsequent tasks (Park et al., 2023; Shinn et al., 2023). As experience accumulates, agents also need to decide what to retain and make available within a limited context. MemGPT uses memory tiers to manage the context budget, while Mem0 extracts and consolidates information for later retrieval (Packer et al., 2023; Chhikara et al., 2025). In multi-agent systems, memory must also support access to experience produced by other agents. G-Memory organizes insights, queries, and interaction histories in a graph hierarchy for cross-task reuse and agent-specific retrieval (Zhang et al., 2025). When several agents contribute to shared memory, however, their contributions may conflict or differ in reliability. Source reliability provides one basis for deciding which contributions to retrieve.

Source reliability and reputation systems. Agreement between sources offers one form of evidence about their reliability. iP-TD estimates competence from agreement across reported values (Meir et al., 2023). Such agreement can be misleading when sources copy from one another. Feedback collected over repeated tasks provides another form of evidence. Reputation systems accumulate positive and negative feedback and discount earlier observations (Jøsang & Ismail, 2002), allowing scores to respond to changes in source behavior. In LLM coordination, historical contribu-

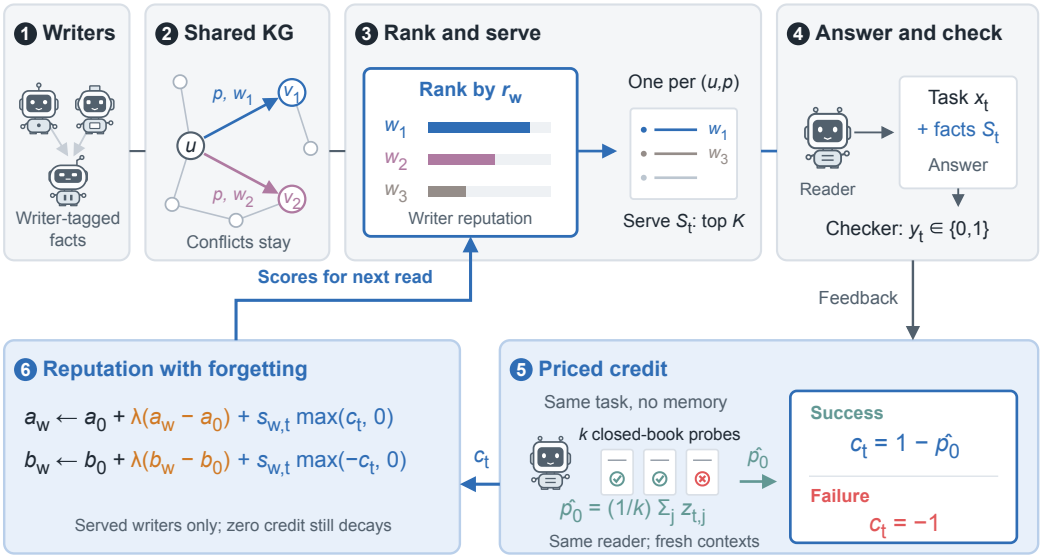

Figure 2: **Priced reputation learning.** Conflicting versions of a relation stay in memory with their writers; retrieval ranks them by writer reputation and serves one version per relation. The outcome is priced against closed-book probes of the same reader: a success adds $1 - \hat{p}_0$, a failure adds -1. Served writers have their evidence above the prior discounted (orange) before receiving credit in proportion to their served share (blue).

tion and correctness feedback guide output aggregation and peer selection, as in Credibility Scoring and Σ-Mem (Ebrahimi et al., 2025; Feng et al., 2026). Reliability assessment can also target stored information directly. TAME uses evaluator feedback on the contribution of retrieved experiences to guide their subsequent use (Cheng et al., 2026), whereas EquiMem calibrates shared-memory updates using retrieval queries and traversal paths (Meng et al., 2026). For methods that learn from task outcomes, the open question is how much credit a success should assign to the information it retrieved.

Credit assignment for memory use. A reader may answer correctly from its parametric knowledge even when the retrieved information is incorrect, so task success alone does not establish the value of that information (Mallen et al., 2023; Neeman et al., 2023; Chen et al., 2026). One approach to measuring memory utility is to compare performance with and without memory. UCOB compares returns with and without skill guidance at the same task state to guide self-distillation and skill-memory updates (Tu et al., 2026). InfoMem measures utility through the increase in the ground-truth answer's per-token log-likelihood under the final memory, using this signal as a training reward for successful trajectories (Han et al., 2026). At the level of individual entries, MemAudit uses counterfactual removal to attribute harmful outputs to stored records during post-hoc auditing (Tan et al., 2026). PRL uses a task-level closed-book baseline to weight positive credit before updating persistent writer reputation for future retrieval. Evidence discounting limits how long accumulated credit affects subsequent source selection.

3 Priced Reputation Learning

Priced Reputation Learning (PRL) learns writer reputation from task outcomes while limiting two ways in which those outcomes mislead: a success that the reader would have achieved without memory, and evidence that outlives the behavior it was collected under. Here, *credit* is the signal assigned from a single task, while *reputation* is the persistent writer score that guides source selection. Section 3.1 fixes the interface, Sections 3.2 and 3.3 give the two mechanisms, and Section 3.4 states when each failure occurs. Figure 2 shows the complete pipeline.

3.1 WRITER REPUTATION IN SHARED MEMORY

We represent shared memory as a directed knowledge multigraph. Each factual edge $f = (u, p, v, w, \tau)$ records its subject u, predicate p, object v, writer w, and insertion index τ. Separate edges preserve repeated and conflicting contributions together with their provenance. Each writer carries positive and negative evidence counts a_w, b_w, initialized at the common prior $a_0 = b_0 = 2$, with reputation $r_w = a_w/(a_w + b_w)$.

For task x_t, retrieval collects facts matching subjects in the query and, for relation composition, follows their objects to a second graph depth. We rank candidates by writer reputation, retain the first version of each subject and predicate pair (u, p), and serve up to $K = 5$ surviving facts as reader context. Ties follow candidate order. Competing versions remain in memory, allowing later reputation updates to change which source is selected.

Let S_t be the served set and $y_t \in \{0, 1\}$ the task checker outcome for the reader answer. For nonempty S_t, writer w receives the share $s_{w,t} = |S_{w,t}|/|S_t|$ of task credit, where $S_{w,t}$ contains its served facts. Only these writers receive updates; writers outside S_t keep their counts, and no writer is updated when S_t is empty. The interface requires writer provenance, the reader, and a task checker, and uses no writer reliability labels.

3.2 PRICING POSITIVE CREDIT AGAINST THE READER BASELINE

A successful task should give less credit to its served writers when the reader is likely to succeed without memory. Let $p_0(x)$ denote the probability that the same reader answers task x correctly under a specified prompt without memory. At feedback time, we estimate this baseline with k closed-book probes, graded by the same task checker to obtain $z_{t,j} \in \{0, 1\}$. Each probe uses fresh system and task messages, with no retrieved facts, previous answer, feedback, or conversation history. PRL assigns task credit using

$$\widehat{p}_0(x_t) = \frac{1}{k}\sum_{j=1}^{k} z_{t,j}, \qquad c_t = \begin{cases} 1 - \widehat{p}_0(x_t), & y_t = 1, \\ -1, & y_t = 0. \end{cases} \tag{1}$$

Baseline estimates are cached by task identifier, and the main configuration uses $k = 1$. When all probes succeed, a successful task adds no positive credit; when all fail, it receives a full unit. Failures keep a full unit of negative credit, so a failed task still supplies corrective feedback when the closed-book probes also fail. The supplementary material relates this asymmetric rule to a symmetric outcome residual and derives its sampling variance.

3.3 LIMITING THE LIFETIME OF REPUTATION EVIDENCE

Pricing controls the new evidence entering reputation. Earlier positive evidence can still favor a writer after its contributions change from correct to false. To limit this persistence, PRL forgets a fraction of the evidence above the common prior whenever a writer receives feedback, and then adds positive or negative credit in proportion to the served share:

$$\begin{aligned} a_w &\leftarrow a_0 + \lambda(a_w - a_0) + s_{w,t}\max(c_t, 0), \\ b_w &\leftarrow b_0 + \lambda(b_w - b_0) + s_{w,t}\max(-c_t, 0). \end{aligned} \tag{2}$$

We use $\lambda = 0.9$, so an observation retains weight λ^j after j further updates to the same writer. Forgetting also applies when $c_t = 0$: a success with $\widehat{p}_0(x_t) = 1$ adds no new evidence and moves accumulated reputation toward the prior score of $1/2$. The update runs throughout the task stream and requires no signal indicating when a writer changes behavior. The supplementary material gives the weighted history and the evidence bound $1/(1 - \lambda)$.

3.4 LOCK-IN AND DETECTION DELAY

We state the two failure modes in a minimal setting. One subject and predicate pair has a false version from writer A and a correct version from writer B; each read serves exactly one of them, with no exploration, and the served writer takes the full share while the unserved writer is not updated. Let q be the probability that a task succeeds when the false version is served, and $\bar{p}_0$ the mean

closed-book baseline over the reads among them that succeed, so that the expected priced credit of a success is $q(1-\bar{p}_0)$ without assuming that success and baseline are independent across tasks. Let P and N be the expected positive and negative credit the false version receives per update. Unpriced credit gives $P = q$ and $N = 1 - q$; Equation 1 gives $P = q(1 - \bar{p}_0)$ and $N = 1 - q$. With $a_0 = b_0$ and any $\lambda \in (0, 1]$, the reputation of A computed from expected counts settles, under continued serving, above the prior $1/2$, the score of the unserved writer, if and only if $P > N$. Unpriced credit therefore locks the false version in whenever $q > 1/2$; priced credit only when $q > 1/(2 - \bar{p}_0)$, and never when the false version succeeds only where the reader would succeed alone ($q \leq \bar{p}_0$). A constant price c gives $P = qc$, so it locks the false version in whenever $c > (1 - q)/q$; the bound depends on q, a property of the reader and the memory condition rather than of the rule, so a constant that is safe in one condition can cross it in another, while $1 - \widehat{p}_0$ moves with q by construction. The condition does not involve λ, so forgetting cannot remove the leak.

Forgetting acts on the second failure mode. A writer that turns from correct to false enters the change with a lead of positive evidence, and without forgetting the number of updates needed to offset that lead grows linearly with it. With forgetting, the evidence above the prior stays below $1/(1 - \lambda)$, so the delay is bounded whatever the length of the correct phase: about ten updates at $\lambda = 0.9$ when half of the post-change tasks fail. On the recorded PopMix traces of Qwen3 8B (Qwen Team, 2025), reads that served only false versions have q between 0.59 and 0.64 and $\bar{p}_0$ between 0.84 and 0.86, above $1/2$ and below $1/(2 - \bar{p}_0)$, so unpriced credit is predicted to lock the false version in and priced credit is not; the unpriced arms serve correct facts on fewer than 1% of attacked reads and Ours on about 95%. On MQuAKE, none of the reads that served only false versions succeeds ($q = 0$), so no lock-in is predicted for either rule, and none occurs. Derivations and the cases the minimal setting omits are in the supplementary material.

4 Experiments

The experiments follow the two failure modes of Section 3.4 from symptom to cause: across memory conditions, then across readers with the memory held fixed, then along the honest phase, and finally with the reader's knowledge removed. Throughout, accuracy is read against unfiltered retrieval, which uses no writer signal, and the writer oracle, which bounds what selection can achieve.

4.1 Setup

We evaluate PRL in a MacNet agent workflow with shared knowledge graph memory (Qian et al., 2025; Yao et al., 2023). The evaluation combines claim verification constructed from PopQA, FEVER, TOFU, and MQuAKE (Mallen et al., 2023; Thorne et al., 2018; Maini et al., 2024; Zhong et al., 2023), in which the reader labels a claim SUPPORTS or REFUTES as in FEVER, with answer generation on PopQA and synthetic questions requiring two reasoning steps. Memory conditions cover false majorities, selective corruption, changing writers, and conflicts with reader knowledge. Writes stay close to deployment wherever the construction allows: in the three FEVER conditions, each writer's sentence passes through an LLM extractor instead of being inserted as a triple, so 7 to 9% of writes miss the intended fact and every read serves three to five facts from several writers; PopMix, TOFU, and MQuAKE insert triples to keep their corruption schedules exact. In PopMix, one reliable writer and two colluding writers contribute to each of 120 facts; the colluding writers switch from correct to matching false values after an initial honest fraction h, varied from 5 to 60%. All writes precede the scored claims, with feedback after each claim. Qwen3 8B is the main reader, and the PopMix comparison spans twelve readers from Qwen3, Gemma 3, Llama 3.1, GPT-OSS, and Gemini 3.5 Flash (Qwen Team, 2025; Gemma Team, 2025; Llama Team, 2024; OpenAI, 2025; Google DeepMind, 2026). We compare PRL (Ours) with methods that share the candidate pool and keep one version per subject and predicate under the same limit K. Unfiltered retrieval and random selection use no writer signal; content judgment asks the reader model to choose within each conflicting group. iP-TD scores sources by agreement across the store (Meir et al., 2023), TAME updates an item utility from task outcomes (Cheng et al., 2026), and Credibility Scoring updates a multiplicative writer score (Ebrahimi et al., 2025), each in an evaluator variant E and a zero-call variant N (Table 1). Unpriced credit is PRL with unit credit and no forgetting, which is the standard beta reputation update (Jøsang & Ismail, 2002) applied to task outcomes. The writer oracle uses the same retrieval and the same one version per relation, but ranks versions by the true reliability class

Table 1: Task accuracy (%) across six memory conditions with Qwen3 8B. For TAME and Credibility Scoring (CrS), E reproduces the contribution step with a model evaluator and N is the zero-call version. Served is the share of served facts that are correct on PopMix reads whose subject carries a false version. Unpriced credit is the beta reputation update on raw task outcomes. Bold and underline mark the best and second best value in each column, excluding the writer oracle.

Method	**PopQA**		**FEVER**			**TOFU**	**MQuAKE**
	PopMix accuracy	PopMix served	False majority	Selective corruption	Changing writer	Fixed corruption	Edited facts
Unfiltered retrieval	<u>84.6</u>	33.4	45.0	75.8	74.6	60.3	10.8
Random	72.5	32.4	50.8	85.8	80.4	52.8	7.5
Content judgment	83.3	<u>65.0</u>	61.7	83.3	80.8	88.1	7.5
iP-TD	67.1	0.3	45.4	82.5	**87.9**	17.8	0.0
TAME N	75.0	41.6	47.5	86.7	81.3	62.2	15.0
TAME E	72.9	40.8	50.8	<u>88.8</u>	84.6	65.0	11.7
CrS N	68.8	0.3	85.4	87.1	72.1	97.5	20.0
CrS E	66.3	0.3	63.8	73.8	74.6	89.7	**27.5**
Unpriced credit	66.3	0.3	**89.6**	71.7	70.4	**98.1**	<u>21.7</u>
Ours	**92.9**	**95.0**	<u>88.8</u>	**92.5**	<u>87.1</u>	<u>97.8</u>	**27.5**
Writer oracle	95.4	100.0	92.9	90.0	91.3	98.1	30.8

of their writers, which the data construction fixes. It serves only correct facts wherever a correct version is stored, so it bounds selection, not accuracy: a method that serves nearly the same facts can land above or below it. Intervals are entity bootstrap intervals, and method differences are tested with paired comparisons. The supplementary material provides dataset constructions, comparator details, implementation settings, seeds, and paired comparisons.

4.2 Comparison across memory conditions

Table 1 compares source selection methods across six memory conditions with a common Qwen3 8B reader, each built to isolate one failure mechanism; all methods in a column see the same memory contents and tasks, and the three FEVER conditions differ only in the corruption pattern. The data sources vary the closed-book baseline that pricing measures: PopQA facts are largely known to the reader, the TOFU biographies are fictitious, and the MQuAKE edits contradict what the reader knows. Ours has the best value in three of the six columns and the second best in the other three: it trails unpriced credit by less than a point under the false majority and on TOFU, and agreement under the changing writer, where paired redundancy guarantees a correct copy.

The pattern across columns follows the baseline. Where the reader cannot answer alone, pricing has no success to discount, and every outcome rule performs alike. On TOFU, unpriced credit and Credibility Scoring N are both within a point of Ours. Where the reader can answer alone, on PopMix, the served column shows the leak directly. Over all reads, unpriced credit, iP-TD, and both Credibility Scoring variants serve correct facts on 20.4% of reads, less than half the 46.0% of random selection; on attacked reads they serve correct facts on 0.3%, an identical value because all four serve the same locked-in colluding version on every attacked read. Once the colluding writers turn, agreement and unpriced outcomes lock their versions in, as Section 3.4 predicts, while Ours serves correct facts on 95.0% of attacked reads.

Two comparators fail for reasons the table makes visible. iP-TD and majority selection rank versions by agreement among writers and use no task feedback; where the false version has more copies than the correct one, as in PopMix, TOFU, and MQuAKE, agreement is wrong by construction, and iP-TD reaches 67.1%, 17.8%, and 0.0% in those columns. Where paired redundancy guarantees a correct copy, as for the changing writer, agreement is right by construction and iP-TD leads. Content judgment shows the same weakness under a false majority, where it falls to 61.7%. The same ordering holds when the reader generates the answer instead of labelling a claim: on PopMix

Table 2: Task accuracy (%) across twelve readers on PopMix at $h = 20\%$. E and N as in Table 1. Unpriced credit assigns unit credit without forgetting; Ours prices positive credit with one online probe and uses $\lambda = 0.9$. Further variants appear in the supplementary material. Bold and underline mark the best and second best value in each column, excluding the writer oracle.

Method	**Qwen3**					**Llama 3.1**		**GPT-OSS**		**Gemma 3**		**Gemini 3.5**
	1.7B	4B	8B	14B	32B	8B	70B	20B	120B	4B	12B	Flash
Unfiltered retrieval	67.9	83.8	84.6	86.3	84.6	60.8	71.3	79.2	80.8	**40.8**	86.7	97.1
Random	60.0	72.5	72.5	74.6	74.6	52.9	72.9	77.5	75.0	29.6	71.7	83.8
Content judgment	71.7	85.8	83.3	89.2	87.5	55.8	87.5	84.2	87.5	29.6	85.8	**98.8**
iP-TD	55.0	61.7	67.1	65.4	64.6	46.7	63.3	67.5	78.3	23.8	57.5	76.3
TAME N	64.6	72.1	75.0	78.3	75.0	54.2	73.3	74.6	77.5	24.6	71.3	82.5
TAME E	58.8	70.4	72.9	76.3	74.2	51.3	71.3	75.0	80.4	24.6	70.8	85.0
CrS N	65.0	94.2	68.8	67.9	65.4	64.2	64.2	67.5	70.8	30.4	73.3	77.1
CrS E	57.1	92.5	66.3	65.4	68.3	67.9	77.5	67.5	82.9	32.5	64.2	75.0
Unpriced credit	57.1	60.8	66.3	65.8	70.0	55.8	66.7	**90.0**	73.3	30.8	56.3	75.8
Priced, no forgetting	65.8	**94.6**	86.7	90.0	82.5	68.3	87.5	86.3	82.1	27.1	85.8	97.1
Ours	**78.3**	94.2	**92.9**	**94.2**	**92.1**	**68.8**	**95.4**	85.8	**90.0**	27.5	**94.2**	97.9
Writer oracle	75.8	97.1	95.4	96.7	93.3	67.1	97.1	91.7	90.4	31.7	97.1	98.8

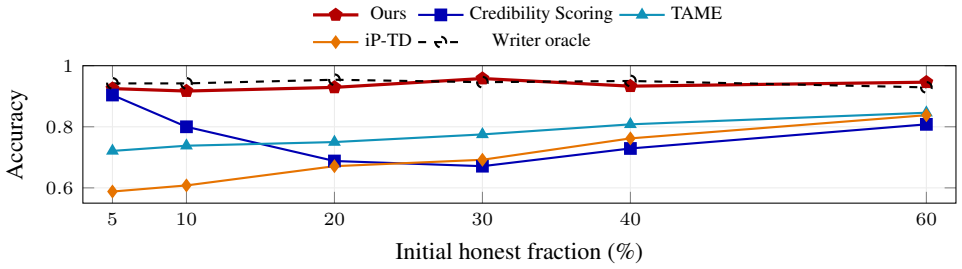

Figure 3: Accuracy as the initial honest fraction changes on PopMix with Qwen3 8B. Credibility Scoring and TAME use their N variants.

answer generation with exact match, unfiltered retrieval reaches 50.8%, unpriced credit 74.2%, Ours 85.8%, and the writer oracle 90.8%.

4.3 Comparison across readers

Table 2 compares twelve readers on PopMix at $h = 20\%$. Because absolute accuracy is set by the reader, each column is read against its own unfiltered and oracle rows, and two patterns hold across them. Ours matches the writer oracle: on eleven of the twelve readers, its accuracy lies between 4.2% below and 2.5% above the oracle, and no paired difference is significant; for Qwen3 8B, served correctness shows that selection itself is close to the oracle (Table 1). Unpriced credit, in contrast, falls below unfiltered retrieval for eleven of the twelve readers, and the loss grows with the accuracy of the reader without selection (Figure 5a). This is the ordering the leak predicts: readers that answer most tasks alone supply the most successes that reward false sources. Model scale is not the variable, since Llama 3.1 70B has a lower unfiltered accuracy than Qwen3 4B and a smaller loss. The exception is GPT-OSS 20B, where unpriced credit is the best rule; Ours is 5.9% below the oracle, a significant gap, and 4.2% below unpriced credit, which is not significant. Its trace shows no lock-in: correct facts are served on 99.8% of reads, because the colluding versions were dislodged when their writers turned and never returned. Priced credit without forgetting improves on unpriced credit for ten of the twelve readers, and adding forgetting improves further on ten.

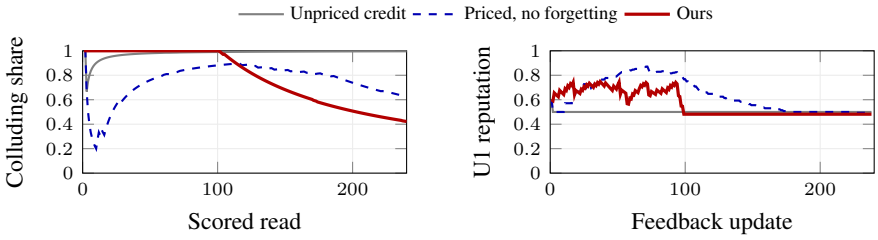

Figure 4: Retrieval dynamics at $h = 40\%$ on PopMix. Left: cumulative fraction of served facts contributed by the two colluding writers. Right: reputation of the colluding writer U1 after successive feedback updates.

Table 3: Component ablation on PopMix, Qwen3 8B, under two honest fractions and under reputation farming (Section 4.6). Ours combines priced credit with one online probe and forgetting.

Update	**Honest fraction**		**Farming**
	$h = 40\%$	$h = 60\%$	
Unpriced, no forgetting	71.7	82.5	72.5
Priced, no forgetting	81.7	87.9	96.3
Unpriced with forgetting	83.3	86.3	71.3
Ours	**93.3**	**94.6**	**98.3**

4.4 Duration of honest behavior

Figure 3 varies the initial honest fraction, which sets how much positive evidence the colluding writers hold when their versions turn false. The rules move in opposite directions. Credibility Scoring N, which keeps all evidence, falls from 90.4% at $h = 5\%$ to 67.1% at $h = 30\%$. The longer the honest phase, the larger the lead that later failures must offset; this is the linear delay of Section 3.4. iP-TD and TAME rise with h instead, because a longer honest phase leaves more correct copies for agreement to find. Ours stays between 91.7% and 95.8% across the sweep; iP-TD and TAME remain significantly lower at every fraction under the paired entity analysis and Credibility Scoring N at every fraction except $h = 5\%$, while the difference from the writer oracle is not significant at any fraction. The supplementary material reports the paired comparisons.

4.5 Credit assignment and reputation dynamics

Table 3 separates the effects of pricing and forgetting. At $h = 40\%$, unpriced credit without forgetting yields 71.7% and the complete update 93.3%; adding either component alone recovers about half of the gap, and the two together close it. Figure 4 shows the associated retrieval dynamics: unpriced credit keeps serving the colluding writers, priced credit alone switches away from them only late, as the evidence from their honest phase persists, and the full update switches early. Random exploration in place of forgetting adds less than one point at $h = 40\%$ (supplementary material), so the gain comes from discounting old evidence rather than from exposing more versions. The ordering replicates across three seeds. Neither hyperparameter is delicate: three online probes instead of one give 85.4% against 86.7% without forgetting at $h = 20\%$, and across $\lambda \in \{0.7, 0.8, 0.9, 0.95\}$ Ours stays between 88.7% and 94.6% at $h = 20\%$, with no paired difference from $\lambda = 0.9$ significant. A constant price can replace the probe only inside the bound of Section 3.4, which moves with the reader: the constant best on Qwen3 8B lies outside it on Gemini 3.5 Flash (bound 0.44) and falls below the per-task price there until recalibrated. The supplementary material reports these runs.

The methods also differ in cost. Ranking, forgetting, and count updates are arithmetic and add no model calls. Ours adds k closed-book probes the first time a task is credited, cached by task identifier; with $k = 1$ this matches the one judge call per task of Credibility Scoring E and is fewer than TAME E and content judgment, which query per served fact and per conflicting group.

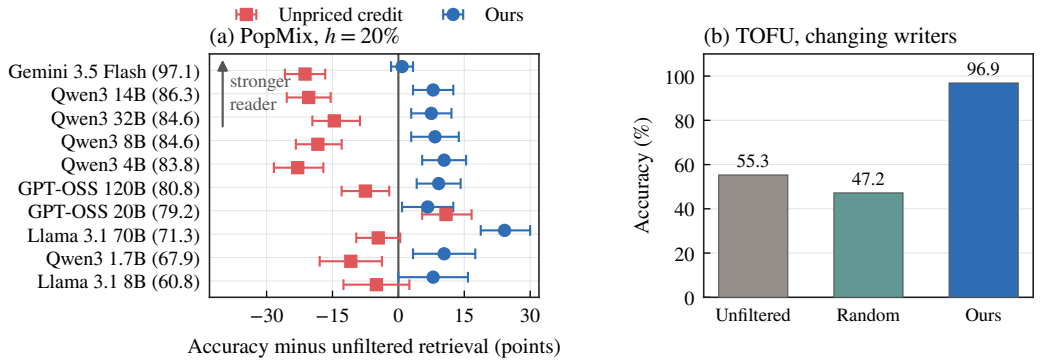

Figure 5: Reader knowledge and source selection. (a) Accuracy of unpriced credit and of Ours relative to unfiltered retrieval for ten readers on PopMix at $h = 20\%$, with paired 95% intervals. (b) Accuracy on TOFU with changing writers for Qwen3 8B, where the reader knows none of the facts and Ours raises accuracy from 55.3% with unfiltered retrieval to 96.9%.

4.6 Reader knowledge and the value of selection

Figure 5a shows the leak reader by reader; Ours stays above unfiltered retrieval on every reader except Gemma 3 4B, discussed below. To see where the leak comes from, we replace every PopMix subject with an invented name and keep the relations and objects, so the reader knows none of the facts. Its online probes show that it then answers SUPPORTS to nearly every claim it cannot verify. A false fact under a SUPPORTS claim therefore still succeeds, and unpriced credit still locks the false versions in: it serves correct facts on 20.0% of reads and reaches 56.7%, below 75.0% for unfiltered retrieval. Ours serves correct facts on 97.5% of reads and reaches 92.9%, the largest gain over unpriced credit in any condition with Qwen3 8B, so the baseline that pricing measures covers success without memory from any source, factual knowledge or a default label. At the other end of the range (Figure 5b), on TOFU with changing writers the reader knows none of the biographies and has no useful default, so selection by outcome credit raises accuracy from 55.3% with unfiltered retrieval to about 97% whether or not it is priced.

The asymmetry behind the invented-name result can be exploited on purpose, by **reputation farming**: a writer that is correct where the reader can verify a claim alone and false where it cannot collects credit from successes the reader would have had anyway. The colluding writers are correct on the 60 PopMix subjects with the highest closed-book baseline and false on 60 subjects with invented names, with the scored claims of both halves interleaved, so most of the successes they earn are ones the reader would have had alone. Unpriced credit counts those successes in full and reaches 72.5%, below 84.6% for unfiltered retrieval, and forgetting does not help (71.3%), as Section 3.4 predicts for a leak. Pricing alone reaches 96.3%, and Ours reaches 98.3% in seed 42, 95.8% and 93.3% in two further seeds; its difference from the writer oracle at 97.1% is not significant (Table 3). The supplementary material reports these runs together with pricing by a constant baseline. Two conditions are bounded by use rather than selection. Under Ours, Gemma 3 4B succeeds on 29% of reads that serve only correct facts, against 95% for Qwen3 8B, so the oracle scores below unfiltered retrieval; on MQuAKE, Ours serves the edited fact on most reads, but the reader uses it in place of its own knowledge on about a third of them.

5 Conclusion and limitations

Reputation learned from task outcomes is misled by successes the reader would have had without memory and by evidence that outlives a change in writer behavior. PRL prices each success against the reader's closed-book baseline and lets evidence expire, and reaches the writer oracle to within reader variation. PRL needs a task checker and k closed-book probes per newly credited task, and its task-level baseline cannot tell which served fact produced a success. Reputation is kept per writer identity, so a version copied under several identities is detected once per identity.

AI USE STATEMENT

AI assistants were used to aid and polish the writing and to draft sections of the paper. All reported measurements come from recorded experiment outputs, and the authors are responsible for the methods, the interpretation, and the final manuscript.

ETHICS STATEMENT

The experiments study factual memory poisoning in controlled agent workflows. Public benchmark facts and constructed entities provide the evaluation material. The work examines how corrupted information affects downstream decisions and how task feedback can improve source selection. The attack descriptions are intended to support reproducible evaluation of memory defenses.

REPRODUCIBILITY STATEMENT

The supplementary material describes dataset construction and specifies prompts and operating settings. The experiment records contain task outcomes, writer provenance, retrieval events, probe counts, and run manifests. Tables and figures are generated from these records and the accompanying statistical summaries. Upon acceptance at a peer-reviewed venue, we will release the complete experimental codebase, including the shared memory implementation, the data construction scripts so that every reported result can be reproduced.

REFERENCES

Yihang Chen, Pin Qian, Su Wang, Sipeng Zhang, Huan Xu, Shuhuai Lin, and Xinpeng Wei. Does RAG know when retrieval is wrong? Diagnosing context compliance under knowledge conflict. *arXiv preprint arXiv:2605.14473*, 2026. URL https://arxiv.org/abs/2605.14473.

Yu Cheng, Yongkang Hu, Jiuan Zhou, Yushuo Zhang, Yihang Chen, Huichi Zhou, Mingang Chen, Zhizhong Zhang, Kun Shao, Yuan Xie, and Zhaoxia Yin. TAME: A trustworthy test-time evolution of agent memory with systematic benchmarking. *arXiv preprint arXiv:2602.03224*, 2026. URL https://arxiv.org/abs/2602.03224.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *European Conference on Artificial Intelligence (ECAI)*, 2025. URL https://doi.org/10.3233/FAIA251160.

Sana Ebrahimi, Mohsen Dehghankar, and Abolfazl Asudeh. An adversary-resistant multi-agent LLM system via credibility scoring. In *International Joint Conference on Natural Language Processing and Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL)*, 2025. URL https://aclanthology.org/2025.ijcnlp-long.90/.

Peilin Feng, Suorong Yang, and Soujanya Poria. Σ-Mem: An online reliability memory for LLM-based multi-agent systems. *arXiv preprint arXiv:2607.27958*, 2026. URL https://arxiv.org/abs/2607.27958.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Google DeepMind. Gemini 3.5 Flash model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-5-flash/.

Tiancheng Han, Yong Li, Wuzhou Yu, Qiaosheng Zhang, and Wenqi Shao. InfoMem: Training long-context memory agents with answer-conditioned information gain. *arXiv preprint arXiv:2606.03329*, 2026. URL https://arxiv.org/abs/2606.03329.

Audun Jøsang and Roslan Ismail. The Beta reputation system. In *Bled Electronic Commerce Conference*, 2002. URL https://aisel.aisnet.org/bled2002/41/.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=B41hNBoWLo.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL https://aclanthology.org/2023.acl-long.546/.

Reshef Meir, Ofra Amir, Omer Ben-Porat, Tsviel Ben Shabat, Gal Cohensius, and Lirong Xia. Frustratingly easy truth discovery. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25750.

Yuqiao Meng, Sakshi Sunil Narvekar, Luoxi Tang, Rupali Rajendra Vaje, Yingxue Zhang, Muchao Ye, and Zhaohan Xi. EquiMem: Calibrating shared memory in multi-agent debate via game-theoretic equilibrium. *arXiv preprint arXiv:2605.09278*, 2026. URL https://arxiv.org/abs/2605.09278.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL https://aclanthology.org/2023.acl-long.559/.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

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 https://arxiv.org/abs/2310.08560.

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 *Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023. URL https://doi.org/10.1145/3586183.3606763.

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/66a026c0d17040889b50f0dfa650e5e0-Abstract-Conference.html.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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 (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Zhewen Tan, Yilun Yao, Huiyan Jin, Wenhan Yu, Guoan Wang, Mengyuan Fan, Liang Lu, Feng Liu, Xiangzheng Zhang, Duohe Ma, Tong Yang, and Lin Sun. MemAudit: Post-hoc auditing of poisoned agent memory via causal attribution and structural anomaly detection. *arXiv preprint arXiv:2605.23723*, 2026. URL https://arxiv.org/abs/2605.23723.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018. URL https://aclanthology.org/N18-1074/.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. UCOB: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026. URL https://arxiv.org/abs/2606.29502.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/136a45cd9b841bf785625709a19c6508-Abstract-Conference.html.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL https://aclanthology.org/2023.emnlp-main.971/.