MEMCREDIT: REINFORCE MEMORY-BASED SEARCH AGENTS VIA RETAINED-STATE CREDIT ASSIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory-based search agents compress interaction histories into compact memories, limiting context growth during multi-turn search. However, useful evidence can be omitted or distorted during memory updates, so successful retrieval alone does not ensure that later turns can use it. Outcome-level supervision does not explicitly distinguish the contributions of individual search–memory conversations to the retained state. To address this gap, we introduce **MemCredit**, a reinforcement learning framework that couples retained-state reward design with conversation-level credit assignment. Specifically, MemCredit defines a memory-state reward based on changes in answerability across consecutive memories, using gold-answer likelihood under a frozen reader to evaluate what remains useful after compression rather than what is merely retrieved. For credit assignment, MemCredit incorporates the resulting feedback into the search and memory conversations, encouraging the reasoning, querying, and memory-writing actions to jointly produce more useful retained states. The framework preserves outcome-level supervision, requires only final-answer annotations, and introduces no additional inference-time scoring module. Experiments on seven question-answering benchmarks with Qwen2.5-3B and Qwen2.5-7B show that MemCredit improves Exact Match (EM) over strongest baseline by 1.97% and 1.03% respectively. Further analyses show that when newly retrieved evidence raises the gold-answer likelihood, MemCredit increases the rate at which this gain is retained in the updated memory by 10.9% over the outcome-only baseline. Our code is available at `https://anonymous.4open.science/r/MemCredit-C05B`.

1 INTRODUCTION

Multi-turn search agents answer complex questions by interleaving reasoning with retrieval: the agent issues a query, reads the returned evidence, and uses it to decide what to do next (Yao et al., 2022; Trivedi et al., 2023; Jin et al., 2025a; Li et al., 2025; Song et al., 2025a). Retaining the full interaction history increases context length with every turn and can make relevant evidence harder to use (Liu et al., 2024; Ai et al., 2026). Compact-memory workflows such as MemSearcher (Yuan et al., 2026) address this problem by learning to rewrite a textual memory after retrieval. The next turn then starts from the question and updated memory. Consequently, memory writing determines which facts and relations remain available for subsequent reasoning.

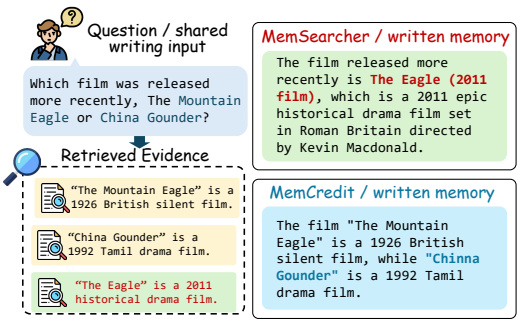

Figure 1: Memory writing for a film-comparison question. MemSearcher introduces an unrelated film, whereas MemCredit retains the two target films and their release years.

Retrieving useful evidence does not ensure that it survives this update. Figure 1 illustrates the problem with a film-comparison question. The retrieved evidence gives the release years of both target films, but MemSearcher substitutes the unrelated *The Eagle* (2011) for *The Mountain Eagle* (1926).

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.

MemCredit's memory preserves the two relevant films and their release years. The distinction lies in what the memory retains, even when the writing input already contains the necessary evidence.

Outcome-level reinforcement learning gives every search–memory conversation in a trajectory the same terminal advantage. Here, a conversation comprises the reasoning, query, and memory update generated at one search turn. This supervision does not explicitly distinguish an update that preserves a useful fact from one that discards it. A failed trajectory may contain a helpful early update, while a successful trajectory may recover information lost at an earlier turn. When all sampled answers receive the same reward, their outcome advantages vanish altogether. These observations motivate a local signal and two associated design choices: which information state to evaluate, and which actions should receive the resulting credit.

We propose **MemCredit**, which evaluates the state retained after memory writing and assigns local credit to the complete interaction that produces it. A frozen reader measures the change in gold-answer likelihood between consecutive memories. Comparing this change among sibling trajectories at the same search turn produces a relative signal, which we project onto a bounded, zero-sum set. The local term then augments the terminal advantage on reasoning, query, and memory-writing tokens. This credit span reflects the joint role of evidence acquisition and memory formation: the query determines what can be retained, and the writer determines what actually survives. The final-answer conversation continues to receive terminal supervision alone.

Across seven QA benchmarks, MemCredit achieves 45.41% and 48.89% average exact match at 3B and 7B under the evaluation protocol in Section 4.1. The corresponding differences from MemSearcher are 1.97 and 1.03 percentage points. Component comparisons favor retained-state scoring over evidence-state scoring and conversation-level credit over memory-only credit. An offline analysis also shows a higher fraction of positive memory gains. Training curves and case studies connect these results to intermediate feedback, while showing that improved accuracy can coexist with longer training responses.

Our contributions are as follows:

- We formulate a retained-state reward that measures changes in compact-memory answerability using only final-answer annotations.
- We develop bounded, group-relative credit for complete search–memory conversations and characterize its ordering, gain alignment, and relation to outcome supervision.
- We evaluate MemCredit on seven QA benchmarks and analyze its reward target, credit span, memory-update behavior, and success and failure cases.

2 Preliminaries

2.1 Policy Optimization and Credit Assignment

From PPO to GRPO. Proximal Policy Optimization (PPO (Schulman et al., 2017)) optimizes a clipped policy objective using advantages typically estimated with a learned value function. To eliminate critic-learning overhead, Group Relative Policy Optimization (GRPO (Shao et al., 2024)) retains the clipped surrogate but estimates advantages from relative rewards among G trajectories sampled for the same question q. Under outcome supervision, trajectory i receives

$$\widehat{A}_{i,k} = A_i^{\text{out}} = \frac{R_i^{\text{out}} - \bar{R}_q}{s_{R,q} + \epsilon_R}, \qquad k = 1, \ldots, L_i, \tag{1}$$

where R_i^{out} is its terminal reward, $\bar{R}_q$ and $s_{R,q}$ are the group reward mean and standard deviation, and L_i counts policy-generated tokens. With $\rho_{i,k}(\theta) = \pi_\theta(w_{i,k} \mid x_{i,k})/\pi_{\text{old}}(w_{i,k} \mid x_{i,k})$, the reference-regularized GRPO objective is

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\text{group}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{k=1}^{L_i}\Big(\min\Big\{\rho_{i,k}(\theta)\widehat{A}_{i,k},\\ \text{clip}(\rho_{i,k}(\theta), 1-\epsilon_c, 1+\epsilon_c)\,\widehat{A}_{i,k}\Big\} - \beta\mathcal{K}_{i,k}(\theta)\Big)\Bigg],\end{aligned} \tag{2}$$

where the expectation is over questions and sampled trajectory groups, ϵ_c controls clipping, and $\mathcal{K}_{i,k}$ penalizes divergence from a fixed reference policy with weight β. Tool observations provide context but are excluded from policy-generated targets.

The credit-assignment gap. Outcome-supervised GRPO distinguishes trajectories but assigns the same outcome coefficient to their constituent decisions. In MemSearcher (Yuan et al., 2026), this coefficient is propagated across search and memory conversations, so an update that preserves useful evidence and one that discards it receive the same outcome advantage. Their token-level gradients may differ, but the coefficient does not explicitly reflect their different contributions to the retained state. When all sibling rewards tie, $A_i^{\text{out}} = 0$ removes the outcome-derived policy-gradient signal, although regularization may remain active. This motivates local credit that evaluates each memory transition and guides the reasoning, querying, and memory-writing actions that jointly produce it.

2.2 Memory-Based Search Agents

A memory-based search agent answers a question q through T_i search conversations and a final-answer conversation. At turn t, the state available for its next decision is the question and a compact memory $M_{i,t-1}$. The agent generates reasoning $r_{i,t}$ and a query or tool action $a_{i,t}$, receives an observation $o_{i,t}$, and writes the next memory:

$$(q, M_{i,t-1}) \longrightarrow (r_{i,t}, a_{i,t}) \longrightarrow o_{i,t} \longrightarrow M_{i,t}. \tag{3}$$

In workflows such as MemSearcher (Yuan et al., 2026), the next conversation is rebuilt from this updated memory rather than the entire interaction history. Memory writing thus determines which retrieved facts, entity bindings, and intermediate conclusions remain available for subsequent reasoning. Algorithm 1 summarizes trajectory sampling and terminal evaluation. Observations influence the interaction but are supplied by the environment, not generated as policy actions.

2.3 From Evidence Acquisition to Retained Utility

Evidence acquisition and memory improvement are distinct events. A passage may support the answer while the resulting memory drops a bridge entity, confuses two relations, or overwrites an earlier useful fact. Conversely, organizing already retrieved information can improve the next decision without adding a new passage. A reward attached only to retrieval or the final answer cannot explicitly distinguish these transitions. The relevant intermediate target is the utility of the memory that survives the update. Figure 4(a) examines retained utility through the fraction of memory updates with positive reader-measured gain. Panels (b) and (c) describe training-group statistics and response length. The memory diagnostic uses the answerability potential defined in Section 3.1; Appendix E.2 distinguishes raw memory gains from the relative coefficients used for training.

3 MemCredit

To address the acquisition–retention gap, we propose MemCredit, a reinforcement learning framework that couples retained-state reward design with search–memory credit assignment. As illustrated in Figure 2, MemCredit first evaluates how each memory update changes the information available for answering the question. It then assigns the resulting comparative credit to the complete interaction that acquires evidence and writes the updated memory. Finally, this local signal augments the outcome advantage used for policy optimization.

3.1 Retained-State Reward

The reward should evaluate the information that remains accessible after an interaction. A score based only on retrieved observations cannot distinguish whether their useful content survives compression, whereas an absolute memory score conflates the contribution of the current update with information already retained. We measure the change in answerability between consecutive memories. Given a question q, a memory M, and accepted-answer aliases $\mathcal{Y}$, a frozen reader p_ψ defines

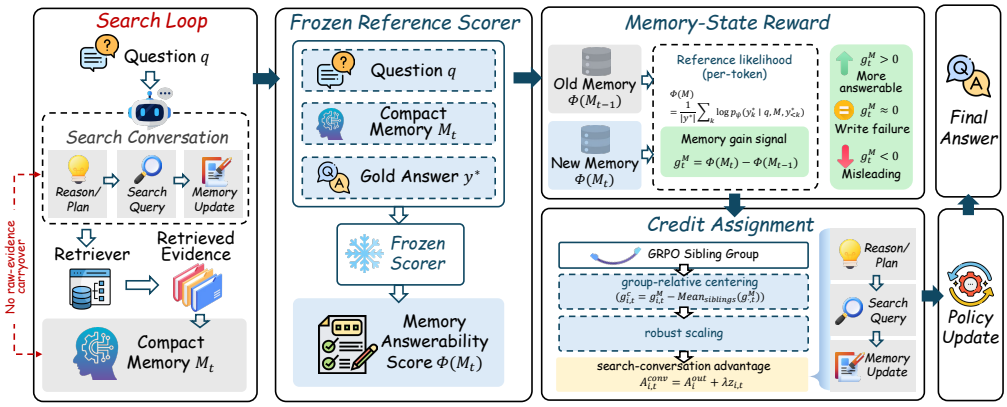

Figure 2: Overview of MemCredit.

the potential

$$\ell_\psi(q, M, y) = \frac{1}{|y|} \sum_{k=1}^{|y|} \log p_\psi(y_k \mid \mathcal{P}(q, M), y_{<k}),$$
$$\Phi_\psi(q, M, \mathcal{Y}) = \max_{y \in \mathcal{Y}} \ell_\psi(q, M, y). \tag{4}$$

Here $\mathcal{P}(q, M)$ presents the question and retained memory to the reader. Length normalization averages over answer tokens, while the maximum accommodates accepted answer forms. Figure 2 shows the single-answer case for readability. The reader is fixed throughout training, and accepted answers are used only as scoring targets, not as additional actor inputs. The potential measures reader-based answerability rather than certifying factual support.

For the t-th search conversation in trajectory i, the retained-state reward is

$$g_{i,t}^M = \Phi_\psi(q, M_{i,t}, \mathcal{Y}) - \Phi_\psi(q, M_{i,t-1}, \mathcal{Y}). \tag{5}$$

Positive and negative gains indicate increases and decreases in reader-measured answerability, respectively. Merely retrieving useful evidence does not yield a positive reward if the retained-state potential fails to improve; conversely, a better synthesis of existing evidence can improve the potential without acquiring new facts. For a fixed scoring function, these rewards satisfy

$$\sum_{t=1}^{T_i} g_{i,t}^M = \Phi_\psi(q, M_{i,T_i}, \mathcal{Y}) - \Phi_\psi(q, M_{i,0}, \mathcal{Y}). \tag{6}$$

Thus, the cumulative raw reward measures the net change in retained answerability: losing and subsequently recovering the same potential creates no additional net gain. This identity concerns the raw rewards, not the normalized credit introduced next.

3.2 Search–Memory Credit Assignment

A retained memory is produced jointly by evidence acquisition and memory writing. The reasoning and query determine the observation available to the writer, while the writer determines what remains available afterward. To capture this dependency, fix a pre-search state $s = (q, M)$, including any required runtime context. With a retriever independent of the policy parameters, the joint interaction distribution factorizes as

$$P_\theta(r, a, o, M' \mid s) = \pi_\theta(r, a \mid s)\, P_{\text{ret}}(o \mid a)\, \pi_\theta(M' \mid s, r, a, o). \tag{7}$$

Let $U(\theta; s) = \mathbb{E}_{P_\theta}[g^M(s, M')]$ be the expected raw gain of this interaction. The following result identifies the policy decisions that contribute to this local objective.

Lemma 1 (Joint interaction gradient). *For a fixed pre-search state, reader, and retriever, assume that differentiation and expectation can be interchanged and that the policy has parameter-independent support. Then*

$$\nabla_\theta U(\theta; s) = \mathbb{E}_{P_\theta}\Big[g^M(s, M')\big(\nabla_\theta \log \pi_\theta(r, a \mid s) + \nabla_\theta \log \pi_\theta(M' \mid s, r, a, o)\big)\Big]. \tag{8}$$

Lemma 1 motivates assigning the memory signal to the complete search–memory conversation. Restricting the local term to memory-writing tokens omits the explicit acquisition term in this raw-utility gradient, although outcome supervision and shared parameters can still change querying behavior. The lemma concerns a fixed-state, one-step surrogate; it does not establish an unbiased gradient for terminal answer accuracy or isolate the responsibility of individual components.

To obtain comparative credit, we consider eligible sibling updates at the same search turn. Let $\mathcal{I}_{q,t}$ be their index set and define

$$\bar{g}_{q,t} = \frac{1}{|\mathcal{I}_{q,t}|} \sum_{i \in \mathcal{I}_{q,t}} g^M_{i,t}, \qquad u_{i,t} = g^M_{i,t} - \bar{g}_{q,t}. \tag{9}$$

This separates absolute potential improvement from relative performance: an update can improve memory yet receive less credit than a stronger sibling. Because later memories may differ across trajectories, the comparison is stage-relative rather than an identical-state intervention. We scale the centered gains by a shared MAD-based batch scale $s_{\text{batch}} > 0$ and project them jointly onto a zero-sum box:

$$\begin{aligned} \mathbf{v}_{q,t} &= \mathbf{u}_{q,t}/s_{\text{batch}}, \\ \mathbf{z}_{q,t} &= \arg\min_{\mathbf{z} \in \mathcal{C}_c} \frac{1}{2}\|\mathbf{z} - \mathbf{v}_{q,t}\|_2^2, \\ \mathcal{C}_c &= \{\mathbf{z} \in \mathbb{R}^{|\mathcal{I}_{q,t}|} : \mathbf{1}^\top \mathbf{z} = 0,\ \|\mathbf{z}\|_\infty \le c\}, \quad c > 0. \end{aligned} \tag{10}$$

The zero-sum constraint preserves the group-relative interpretation, while the box constraint limits individual coefficients. Joint projection enforces both constraints even when large coefficients saturate. Equivalently, $z_{i,t} = \text{clip}(v_{i,t} - \alpha_{q,t}, -c, c)$, where the shared shift $\alpha_{q,t}$ is chosen so that $\sum_i z_{i,t} = 0$ (Appendix B.3).

Lemma 2 (Order-preserving, gain-aligned credit)**.** *For a fixed eligible group with finite gains and $s_{\text{batch}} > 0$, Eq. 10 has a unique solution. For any $i, j \in \mathcal{I}_{q,t}$,*

$$g^M_{i,t} \ge g^M_{j,t} \implies z_{i,t} \ge z_{j,t}. \tag{11}$$

Moreover, its alignment with the sampled gains satisfies

$$\sum_{i \in \mathcal{I}_{q,t}} g^M_{i,t} z_{i,t} \ge s_{\text{batch}} \|\mathbf{z}_{q,t}\|_2^2 \ge 0. \tag{12}$$

Lemma 2 shows that the constraints can compress gain differences but cannot reverse their ordering. The alignment inequality further connects the assigned coefficients to the sampled retained-state gains, rather than to an arbitrary redistribution of credit. These are properties of the local coefficients; their effect on policy learning also depends on the sampled actions and the optimization objective.

3.3 Policy Optimization

We combine local retained-state credit with the outcome advantage from Eq. 1:

$$A^{\text{search}}_{i,t} = A^{\text{out}}_i + \lambda z_{i,t}, \qquad A^{\text{answer}}_i = A^{\text{out}}_i, \quad \lambda \ge 0. \tag{13}$$

The search coefficient is shared by the generated reasoning, query, and memory-writing tokens of the corresponding conversation. The final-answer conversation retains outcome supervision, and observations are excluded from the policy-target loss. This combines evaluation of task completion with a finer comparison of the interactions that form the retained state.

Lemma 3 (Bounded outcome refinement)**.** *The coefficients in Eq. 13 satisfy*

$$\begin{aligned} |A^{\text{search}}_{i,t} - A^{\text{out}}_i| &\le \lambda c, \\ \sum_{i \in \mathcal{I}_{q,t}} (A^{\text{search}}_{i,t} - A^{\text{out}}_i) &= 0. \end{aligned} \tag{14}$$

The outcome sign is preserved whenever $|A^{\text{out}}_i| > \lambda c$. If all sibling outcome rewards tie, an eligible group with nonconstant gains still has nonzero search coefficients provided that $s_{\text{batch}} > 0$, $c > 0$, and $\lambda > 0$.

Lemma 3 bounds the local change to each conversation coefficient while permitting distinctions within an outcome-tied group. A nonzero coefficient does not by itself ensure a nonzero aggregate policy gradient, and coefficient balance is not a policy-invariance guarantee. MemCredit optimizes the resulting coefficients using the same proximal surrogate and token weighting as its baseline. For a policy token $w_{i,t,k}$ under context $x_{i,t,k}$, write $\rho_{i,t,k}(\theta) = \pi_\theta(w_{i,t,k} \mid x_{i,t,k})/\pi_{\text{old}}(w_{i,t,k} \mid x_{i,t,k})$. With $\widehat{A}_{i,t,k}$ given by the corresponding search or answer coefficient, the policy-gradient loss is

$$\mathcal{L}_{\text{PG}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\text{pol}}}\left[\widetilde{h}\Big(\rho_{i,t,k}(\theta), \widehat{A}_{i,t,k}\Big)\right], \tag{15}$$

where $\widetilde{h}$ is the baseline's clipped surrogate and $\mathcal{D}_{\text{pol}}$ denotes its weighted distribution over policy-generated tokens. Existing KL and entropy terms are retained. The reader and constructed coefficients are held fixed during optimization, so gradients act through the generated interaction rather than through the memory scorer. Algorithm 1 and Appendix C specify the execution details.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate MemCredit on seven QA benchmarks: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). These cover factual question answering and multi-hop reasoning. We compare against direct prompting, retrieval-based methods (RAG, IRCoT, and Search-o1), and training-based methods (SFT, R1-Instruct, Search-R1, Dr. Zero, and MemSearcher). Results for the first eight baselines are taken from Yue et al. (2026). MemSearcher (Yuan et al., 2026) provides the closest memory-based reference. Dataset and baseline descriptions appear in Appendix C.

Evaluation metric. We report exact match (EM, %). The main-table average weights the seven datasets equally. For validation, pooled EM weights questions equally, whereas macro EM weights datasets equally. We analyze the seven-benchmark comparison and checkpoint-validation curves separately, with the corresponding populations and aggregation rules described in Appendix C.3.

Implementation details. We use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024) as backbones, with a frozen reader initialized from the corresponding model. Training samples five sibling trajectories per question with a prompt batch size of 256 and learning rate 10^{-6}. Hyperparameters, scoring conventions, and evaluation populations are described in Appendix C.

4.2 Question-Answering Performance

Table 1 presents the results across seven benchmarks. MemCredit obtains the highest listed average EM at both scales: 45.41% at 3B and 48.89% at 7B. Relative to MemSearcher, the gains are 1.97 and 1.03 percentage points, respectively. MemCredit exceeds the reference on all seven datasets at 3B and six at 7B. The exception is 7B HotpotQA, where its EM is 49.49%, compared with 49.56% for MemSearcher.

The largest dataset-level improvements occur on 3B 2Wiki and on Bamboogle. On 2Wiki, 3B EM increases from 43.50% to 48.14%, a gain of 4.64 points. On Bamboogle, the gains are 3.20 points at 3B and 4.00 points at 7B. Averaged over HotpotQA, 2Wiki, MuSiQue, and Bamboogle, MemCredit reaches 39.60% and 43.58%, compared with 36.73% and 42.42% for MemSearcher. This pattern is consistent with the motivation for preserving information across successive searches.

Figure 3 complements the summary scores with checkpoint-level comparisons. All three methods improve rapidly at the start of training. MemCredit separates from A1 and A2 later and achieves higher late-stage pooled and macro EM. A2 shows a more pronounced decline around the middle of training, especially in the macro panels. The advantage of MemCredit therefore appears across multiple later checkpoints.

Table 1: Exact match (%) on seven QA benchmarks. Bold and underline mark the best and second-best values in each model group.

Method	NQ	Trivia	PopQA	Hotpot	2Wiki	MuSiQue	Bam.	Avg.
Qwen2.5-3B-Instruct								
Prompting	10.60	28.80	10.80	14.90	24.40	2.00	2.40	13.40
IRCoT	11.10	31.20	20.00	16.40	17.10	6.70	24.00	18.10
Search-o1	23.80	47.20	26.20	22.10	21.80	5.40	32.00	25.50
RAG	34.80	54.40	38.70	25.50	22.60	4.70	8.00	27.00
SFT	24.90	29.20	10.40	18.60	24.80	4.40	11.20	17.60
R1-Instruct	21.00	44.90	17.10	20.80	27.50	6.00	19.20	22.40
Search-R1	32.30	53.70	36.40	30.80	33.60	10.50	31.50	32.70
Dr. Zero	39.70	57.20	43.10	29.80	29.10	9.10	20.00	32.60
MemSearcher	47.00	63.66	46.55	43.90	43.50	17.90	41.60	43.44
MemCredit	**47.65**	**63.95**	**47.90**	**45.08**	**48.14**	**20.36**	**44.80**	**45.41**
Qwen2.5-7B-Instruct								
Prompting	13.40	40.80	14.00	18.30	25.00	3.10	12.00	18.10
IRCoT	22.40	47.80	30.10	13.30	14.90	7.20	22.40	23.90
Search-o1	15.10	44.30	13.10	18.70	17.60	5.80	29.60	20.60
RAG	34.90	58.50	39.20	29.90	23.50	5.80	20.80	30.40
SFT	31.80	35.40	12.10	21.70	25.90	6.60	11.20	20.70
R1-Instruct	27.00	53.70	19.90	23.70	29.20	7.20	29.30	27.10
Search-R1	39.70	60.60	40.40	38.00	32.60	16.80	40.80	38.40
Dr. Zero	40.60	60.80	41.60	36.20	34.70	10.40	36.00	37.20
MemSearcher	50.75	66.79	47.80	**49.56**	48.52	24.41	47.20	47.86
MemCredit	**51.11**	**67.28**	**49.50**	49.49	**49.13**	**24.49**	**51.20**	**48.89**

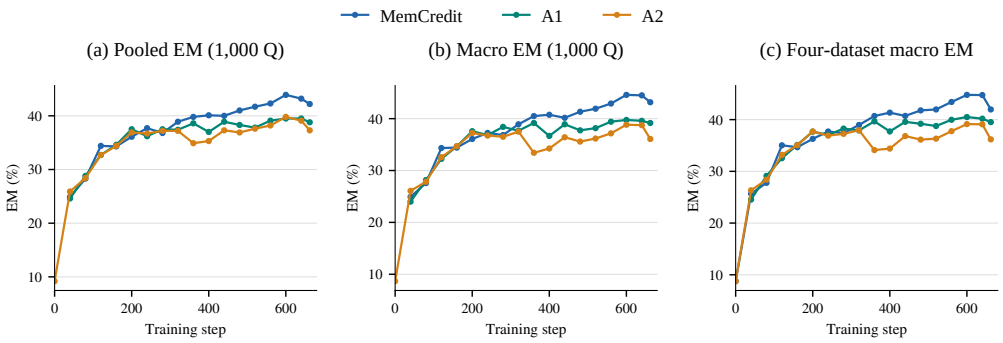

Figure 3: Validation EM of MemCredit and its ablations. (a) Pooled EM and (b) macro EM on the fixed 1,000-question subset; (c) four-dataset macro EM on the 1,025-question validation pool. Markers denote evaluated checkpoints. A1 uses evidence-state scoring, and A2 restricts local credit to memory-writing tokens.

4.3 Memory Quality and Training Behavior

Memory-update quality. Figure 4(a) compares frozen-reader gains on trajectories from 1,152 questions across HotpotQA, 2Wiki, and MuSiQue. The overall positive-gain fraction rises from 46.4% for MemSearcher to 51.4% for MemCredit, corresponding to a 5.03-point absolute and 10.9% relative gain computed before rounding. Improvements occur on all three datasets and are largest on 2Wiki (7.81 points). These results indicate that more MemCredit updates improve reader-based answerability.

Training signal. Figure 4(b) shows a higher logged valid-group percentage for MemCredit throughout training. The aggregate valid-group trace and this within-group example capture dif-

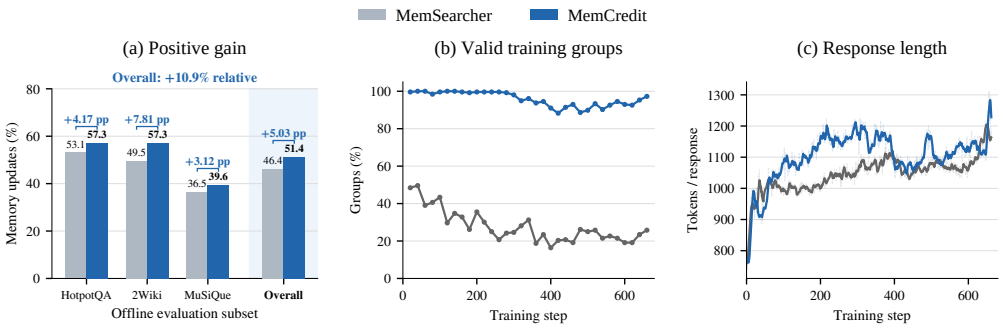

Figure 4: Memory gains and training behavior. (a) Percentage of memory updates with positive frozen-reader gain, evaluated on trajectories from 1,152 questions. (b) Logged valid-group percentage and (c) tokens per response during 3B training. Gray: MemSearcher; blue: MemCredit. Faint and solid lines in (c) denote original and smoothed traces.

ferent aspects of training feedback. Appendix E.1 further distinguishes group eligibility, coefficient activity, and activity conditional on tied outcomes.

Response length and outcome reward. Figure 4(c) shows that MemCredit often produces longer training responses, particularly mid-training. The curves become closer later, but these measurements do not establish reduced inference tokens or latency. MemCredit's terminal reward rises rapidly early in training at both model sizes and then fluctuates around a higher level (Appendix E.3). The method adds no inference-time reader calls; total deployment cost still depends on the learned trajectory lengths.

4.4 Ablation Study

We examine three variants of the memory-based workflow using the 3B model. Table 2 reports macro EM on the same seven datasets under the same evaluation protocol as the main comparison. **A1: Evidence-state** evaluates the prior memory together with the retrieved observation, while retaining conversation-level credit. **A2: Memory-only** evaluates the updated memory but restricts local credit to memory-writing tokens. **Outcome-only** removes the local term.

Table 2: Ablation results on credit-state and credit-span designs.

Method	EM
MemCredit	**45.41%**
A1: Evidence-state	44.68% (↓ 1.61%)
A2: Memory-only	44.03% (↓ 3.04%)
Outcome-only	43.44% (↓ 4.34%)

As shown in Table 2, the full method obtains 45.41% EM. Replacing retained-state scoring with evidence-state scoring reduces EM to 44.68%. Restricting local credit to the writer reduces EM to 44.03%. Removing local credit gives 43.44%. The parenthesized percentages in the table are relative reductions from the full method. The ordering supports both scoring what survives compression and assigning feedback to the actions that acquire the evidence.

5 Conclusion

We presented MemCredit, which evaluates changes in retained-memory answerability and assigns local credit to the search–memory conversations that produce them. A frozen reader provides the training signal, while bounded group-relative coefficients refine terminal supervision. Across seven benchmarks, MemCredit reaches the best average EM at 3B and 7B. Ablations support both retained-state evaluation and conversation-level credit, while offline analysis shows more positive memory gains.

The answerability potential remains a proxy and may not fully capture evidential support or delayed utility from intermediate facts. In addition, experiments use one training seed, and MemCredit can produce longer trajectories despite requiring no inference-time reader calls.

AI USE STATEMENT

Generative AI tools assisted with manuscript organization, language editing, LaTeX preparation, and consistency checks between the text and experimental figures and tables. The authors are responsible for reviewing the final text, numerical claims, references, and research artifacts.

ETHICS STATEMENT

This work studies memory-based agents for benchmark question answering. A likelihood-based memory score does not establish that an answer is supported by reliable evidence. Search agents may retain inaccurate or misleading content, and memory compression can remove qualifications needed to interpret a source. Applications should therefore assess evidential support and source quality in addition to answer accuracy, particularly when errors have consequences beyond benchmark evaluation.

REPRODUCIBILITY STATEMENT

Appendix A gives the training algorithm and token support, and Appendix B proves the stated mathematical properties. Appendix C describes datasets, baselines, reported configurations, evaluation aggregation, and component definitions. Additional validation curves, diagnostic definitions, and case studies appear in Appendices F. The LaTeX source includes the tables and figure files used in the paper. Code: `https://anonymous.4open.science/r/MemCredit-C05B`.

REFERENCES

Qiuyuan Ai, Zenghuang Fu, Zhaoyang Li, Ping Jiang, Haoyu Wu, Jie Song, and Guannan He. Cognitive scaffold: From fluid context to crystallized memory for long-horizon deepresearch agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25526–25542, 2026.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *arXiv preprint arXiv:1806.07857*, 2018.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Fan Yang, Zenan Zhou, Weipeng Chen, Haofen Wang, Jeff Z. Pan, Wen Zhang, and Huajun Chen. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. doi: 10.18653/v1/2020.coling-main.580.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025a. URL `https://openreview.net/forum?id=Rwhi91ideu`.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025b.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

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

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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023. doi: 10.18653/v1/2023.acl-long.546.

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.

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. *arXiv preprint arXiv:2304.03442*, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023. doi: 10.18653/v1/2023.findings-emnlp.378.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025a.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025b.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *arXiv preprint arXiv:2306.07174*, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. Memsearcher: Iterative memory integration for search agent via end-to-end reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14965–14977, 2026.

Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. Dr. zero: Self-evolving search agents without training data. *arXiv preprint arXiv:2601.07055*, 2026.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. *arXiv preprint arXiv:2505.11821*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

A Training Algorithm

A.1 Training Procedure and Token Support

Algorithm 1 summarizes MemCredit. The behavior policy samples sibling trajectories, and a frozen reader evaluates each pre- and post-update memory. Gains are centered among eligible siblings at the same turn and normalized using a shared batch scale. Joint projection enforces the zero-sum and magnitude constraints before local credit is added to the outcome advantage.

A search conversation comprises the generated reasoning and query followed by the memory update conditioned on the returned observation. The local coefficient applies to all of these generated actions. The observation supplies context to the memory writer but has no policy-loss term. The final-answer conversation, including its reasoning, receives only the outcome advantage. In A2, reasoning and query tokens retain their outcome advantage while only memory-writing tokens receive the added local term.

A.2 Scorer Inputs and Memory Context

The reader input $\mathcal{P}(q, M)$ contains the question and the memory being evaluated. For each accepted alias, the scorer averages log likelihood over its answer tokens and takes the maximum across aliases, as in Eq. 4. Initial memory is scored by the same function as later memories. Accepted answers enter the reader as targets during training and are absent from the actor's generation context.

Each next search conversation is constructed from the question and the updated memory. The current observation is available while writing that update. This distinction is central to the reward: g^M measures the change in the state carried into later turns. Scorer context limits for the recorded runs are listed in Appendix C.

B Theoretical Analysis

B.1 Joint Interaction Gradient

Proof of Lemma 1. Fix $s = (q, M)$ and write $x = (r, a, o, M')$. The reader and initial state are fixed, so $g(x) = \Phi_\psi(q, M', \mathcal{Y}) - \Phi_\psi(q, M, \mathcal{Y})$ has no direct dependence on θ. On the parameter-independent support $\mathcal{X}$, the objective is

$$U(\theta; s) = \sum_{x \in \mathcal{X}} P_\theta(x \mid s) g(x). \tag{16}$$

Using the differentiation condition in the lemma and the identity $\nabla_\theta P_\theta = P_\theta \nabla_\theta \log P_\theta$ on the support gives

$$\begin{aligned} \nabla_\theta U(\theta; s) &= \sum_x g(x) \nabla_\theta P_\theta(x \mid s) \\ &= \sum_x P_\theta(x \mid s) g(x) \nabla_\theta \log P_\theta(x \mid s) \\ &= \mathbb{E}_{P_\theta}\left[g(x) \nabla_\theta \log P_\theta(x \mid s) \right]. \end{aligned} \tag{17}$$

The factorization in Eq. 7 yields

$$\begin{aligned} \nabla_\theta \log P_\theta(x \mid s) = \nabla_\theta \log \pi_\theta(r, a \mid s) + \nabla_\theta \log P_{\mathrm{ret}}(o \mid a) \\ + \nabla_\theta \log \pi_\theta(M' \mid s, r, a, o). \end{aligned} \tag{18}$$

The middle term is zero because the retriever is independent of θ. Substitution into Eq. 17 proves Eq. 8. With finite generation horizons and a finite vocabulary, the policy sums are finite; for a more general support, the stated interchange condition supplies the same calculation. □

To see how the two terms relate to token support, let $\mathcal{K}_{\mathrm{acq}}$ index the generated reasoning and query tokens and $\mathcal{K}_{\mathrm{mem}}$ the memory tokens. Autoregressive factorization gives

$$\nabla_\theta \log P_\theta(x \mid s) = \sum_{k \in \mathcal{K}_{\mathrm{acq}}} \nabla_\theta \log \pi_\theta(w_k \mid x_k) + \sum_{k \in \mathcal{K}_{\mathrm{mem}}} \nabla_\theta \log \pi_\theta(w_k \mid x_k). \tag{19}$$

Algorithm 1 MemCredit training

Require: Policy π_θ, frozen reader p_ψ, retriever P_{ret}, G, λ, c

for each batch of questions q and accepted answers $\mathcal{Y}$ **do**
  Sample G sibling trajectories with π_{old}
  Compute terminal rewards and A_i^{out} using Eq. 1
  Evaluate initial and updated memory potentials with Eq. 4
  Compute $g_{i,t}^M = \Phi_\psi(q, M_{i,t}, \mathcal{Y}) - \Phi_\psi(q, M_{i,t-1}, \mathcal{Y})$
  Initialize all local coefficients $z_{i,t} \leftarrow 0$
  Form eligible question–turn groups and centered gains $\mathbf{u}_{q,t}$
  Compute the shared batch scale s_{batch}
  if $s_{\mathrm{batch}} > 0$ **then**
    for each eligible group with at least two updates **do**
      $\mathbf{v}_{q,t} \leftarrow \mathbf{u}_{q,t} / s_{\mathrm{batch}}$
      $\mathbf{z}_{q,t} \leftarrow \Pi_{\mathcal{C}_c}(\mathbf{v}_{q,t})$ ▷ Eq. 10
    end for
  end if
  Assign $A_i^{\mathrm{out}} + \lambda z_{i,t}$ to search reasoning, query, and memory tokens
  Assign A_i^{out} to final-answer conversation tokens
  Exclude question, prior context, and tool observations from policy targets
  Optimize the proximal objective with the reader and coefficients held fixed
end for

Both sums are multiplied by the same realized gain in the expected raw-utility gradient. Moreover, the initial potential is a state baseline:

$$\Phi_\psi(q, M, \mathcal{Y})\, \mathbb{E}_{P_\theta}[\nabla_\theta \log P_\theta(x \mid s)] = \Phi_\psi(q, M, \mathcal{Y})\, \nabla_\theta \sum_x P_\theta(x \mid s) = 0. \tag{20}$$

Subtracting it therefore preserves this local expected gradient. Lemma 1 identifies the contributing action spans for the raw, fixed-state objective; the group normalization and proximal objective subsequently define the training surrogate.

B.2 Raw Potential Telescoping

Fix the question, accepted answers, and reader, and let $\Phi_t = \Phi_\psi(q, M_{i,t}, \mathcal{Y})$. Then

$$\begin{aligned} \sum_{t=1}^{T_i} g_{i,t}^M &= (\Phi_1 - \Phi_0) + (\Phi_2 - \Phi_1) + \cdots + (\Phi_{T_i} - \Phi_{T_i - 1}) \\ &= \Phi_{T_i} - \Phi_0. \end{aligned} \tag{21}$$

Each intermediate potential occurs once with each sign. The identity also holds for $T_i = 0$ by the empty-sum convention. Thus, a sequence that returns to the same potential has zero net raw gain. Group centering, scaling, and projection compare different siblings and stages, so this trajectory identity applies to g^M, rather than to the resulting coefficients z.

B.3 Zero-Sum Box Projection and Gain Alignment

Proof of Lemma 2. Write $n = |\mathcal{I}_{q,t}|$, $s = s_{\mathrm{batch}} > 0$, $\mathbf{g} = (g_{i,t}^M)_{i=1}^n$, and $\mathbf{v} = (\mathbf{g} - \bar{g}\mathbf{1})/s$. The set $\mathcal{C}_c$ contains zero and is closed, bounded, and convex. The continuous objective $\frac{1}{2}\|\mathbf{z} - \mathbf{v}\|_2^2$ attains its minimum on this set. Its strict convexity gives a unique minimizer $\mathbf{z}^*$.

Projection formula. For a scalar α, define

$$z_i(\alpha) = \mathrm{clip}(v_i - \alpha, -c, c), \qquad F(\alpha) = \sum_{i=1}^n z_i(\alpha). \tag{22}$$

The function F is continuous and nonincreasing. At $\alpha_- = \min_i v_i - c$, every coordinate equals c, whereas at $\alpha_+ = \max_i v_i + c$, every coordinate equals $-c$. Hence $F(\alpha_-) = nc$ and $F(\alpha_+) = -nc$,

so a root α^* exists. For this root, $\mathbf{z}(\alpha^*)$ is feasible and minimizes

$$\frac{1}{2}\|\mathbf{z}-\mathbf{v}\|_2^2+\alpha^*\mathbf{1}^\top\mathbf{z} \quad \text{over } [-c,c]^n, \tag{23}$$

because each coordinate is the minimizer of its corresponding scalar quadratic. On $\mathcal{C}_c$, the added linear term vanishes. Therefore $\mathbf{z}(\alpha^*)$ is exactly $\mathbf{z}^*$. The scalar root may be nonunique on a flat interval, but every root produces the same unique projected vector.

Order preservation. If $g_i \geq g_j$, then $v_i \geq v_j$ since centering and division by $s>0$ are shared. Subtracting the same α^* preserves this inequality, and clipping is a nondecreasing function. Thus $z_i^* \geq z_j^*$, proving Eq. 11, including equality and saturated coordinates.

Gain alignment. For any $\mathbf{w} \in \mathcal{C}_c$, the segment $\mathbf{z}^*+\eta(\mathbf{w}-\mathbf{z}^*)$ is feasible for $0 \leq \eta \leq 1$. The one-sided derivative of the objective at its minimum is nonnegative:

$$\langle \mathbf{z}^*-\mathbf{v}, \mathbf{w}-\mathbf{z}^*\rangle \geq 0. \tag{24}$$

Taking $\mathbf{w}=\mathbf{0}$ gives $\langle \mathbf{v},\mathbf{z}^*\rangle \geq \|\mathbf{z}^*\|_2^2$. Since $\mathbf{1}^\top\mathbf{z}^*=0$ and $\mathbf{g}=s\mathbf{v}+\bar{g}\mathbf{1}$,

$$\mathbf{g}^\top\mathbf{z}^* = s\mathbf{v}^\top\mathbf{z}^*+\bar{g}\mathbf{1}^\top\mathbf{z}^* \geq s\|\mathbf{z}^*\|_2^2 \geq 0. \tag{25}$$

This proves Eq. 12. □

Equation 22 also gives a direct computation by bisection on $[\alpha_-,\alpha_+]$. The shared shift matters: ordinary coordinatewise clipping of $(3,-1,-2)$ to $[-1,1]$ gives $(1,-1,-1)$, whose sum is -1. Joint projection instead returns $(1,0,-1)$ and satisfies both constraints.

B.4 Bounded Coefficients and Outcome Ties

Proof of Lemma 3. By construction, $A_{i,t}^{\text{search}}-A_i^{\text{out}}=\lambda z_{i,t}$. Since $\lambda \geq 0$ and $\mathbf{z}\in\mathcal{C}_c$,

$$|A_{i,t}^{\text{search}}-A_i^{\text{out}}| = \lambda|z_{i,t}| \leq \lambda c, \qquad \sum_i (A_{i,t}^{\text{search}}-A_i^{\text{out}}) = \lambda\mathbf{1}^\top\mathbf{z}=0. \tag{26}$$

If $A_i^{\text{out}} > \lambda c$, then $A_{i,t}^{\text{search}} \geq A_i^{\text{out}}-\lambda c>0$. If $A_i^{\text{out}}<-\lambda c$, then $A_{i,t}^{\text{search}} \leq A_i^{\text{out}}+\lambda c<0$. This proves sign preservation under the strict bound.

It remains to establish the nonzero claim. When all sibling outcome rewards equal a common value, their centered numerators in Eq. 1 vanish, so $A_i^{\text{out}}=0$. Nonconstant finite memory gains and $s>0$ imply $\mathbf{v}\neq\mathbf{0}$, with $\mathbf{1}^\top\mathbf{v}=0$. Choose

$$\eta = \min\{1, c/\|\mathbf{v}\|_\infty\} > 0. \tag{27}$$

Then $\eta\mathbf{v}\in\mathcal{C}_c$, and

$$\frac{1}{2}\|\eta\mathbf{v}-\mathbf{v}\|_2^2-\frac{1}{2}\|\mathbf{0}-\mathbf{v}\|_2^2 = -\frac{1}{2}\eta(2-\eta)\|\mathbf{v}\|_2^2<0. \tag{28}$$

Consequently zero cannot be the projection minimizer, and $\mathbf{z}^*\neq\mathbf{0}$. For $\lambda>0$, the search coefficient vector $\lambda\mathbf{z}^*$ is nonzero. Because it sums to zero, it contains both a positive and a negative entry. □

Equal gains give $\mathbf{v}=\mathbf{z}=\mathbf{0}$; singleton groups have the same property. The disabled-scale rule also sets local credit to zero. For the recorded $\lambda=0.1$ and $c=2$, the per-conversation perturbation is at most 0.2. These are coefficient properties: different sampled actions have different score-function vectors, so coefficient balance does not imply cancellation of their parameter gradients.

C Experimental Details

C.1 Dataset Descriptions

We evaluate on seven benchmarks spanning factual retrieval and multi-hop question answering.

- **NQ** (Kwiatkowski et al., 2019) contains information-seeking questions derived from real search queries, with answers grounded in Wikipedia.
- **TriviaQA** (Joshi et al., 2017) pairs trivia questions with evidence documents and accepted answer aliases.
- **PopQA** (Mallen et al., 2023) tests factual knowledge across entities with different levels of popularity.
- **HotpotQA** (Yang et al., 2018) requires reasoning over multiple Wikipedia passages, including bridge and comparison questions.
- **2WikiMultiHopQA** (Ho et al., 2020) combines structured and textual information to construct questions with explicit multi-hop reasoning paths.
- **MuSiQue** (Trivedi et al., 2022) composes single-hop questions into questions requiring multiple connected reasoning steps.
- **Bamboogle** (Press et al., 2023) contains manually constructed questions designed to require combining information across searches.

NQ, TriviaQA, and PopQA form the factual-QA group in our analysis. HotpotQA, 2Wiki, MuSiQue, and Bamboogle form the multi-hop group. The four-dataset average in Section 4.2 assigns equal weight to these four multi-hop benchmarks.

C.2 Baseline Descriptions

Prompting and retrieval-based methods. Prompting answers directly from the instruction-tuned model. RAG (Lewis et al., 2020) conditions generation on retrieved evidence. IRCoT (Trivedi et al., 2023) interleaves retrieval with chain-of-thought reasoning, and Search-o1 (Li et al., 2025) equips a reasoning workflow with search.

Training-based methods. SFT is the supervised fine-tuning baseline reported by Yue et al. (2026); R1-Instruct uses reinforcement learning without search. Search-R1 (Jin et al., 2025a) trains reasoning and search with reinforcement learning. Dr. Zero (Yue et al., 2026) jointly evolves a question proposer and a solver without a curated question-answer training set. We use the benchmark scores reported by Yue et al. (2026) for these methods. Their training data and optimization settings need not coincide with those of MemCredit.

Memory-based reference. MemSearcher (Yuan et al., 2026) alternates search and memory writing and applies terminal supervision across the resulting conversations. It provides the workflow on which MemCredit adds retained-state credit. The MemSearcher row in Table 1 is a benchmark reference; the outcome-only variant in Table 2 is defined by disabling the local term.

C.3 Evaluation Populations and Metrics

Table 3 summarizes the populations used for benchmark evaluation, checkpoint validation, and offline memory analysis. The 1,000-question and 1,025-question validation populations are treated separately throughout the paper. The latter comprises NQ, HotpotQA, MuSiQue, and Bamboogle. The offline analysis counts memory transitions, so the number of questions is not the denominator of an update rate.

Let e_j denote the binary exact-match result for question j, and let $\mathcal{D}_d$ contain the evaluated questions from dataset d. For D datasets, the two aggregations are

$$\begin{aligned} \mathrm{EM}_{\mathrm{pooled}} &= 100\,\frac{\sum_{d=1}^{D}\sum_{j\in\mathcal{D}_d} e_j}{\sum_{d=1}^{D}|\mathcal{D}_d|}, \\ \mathrm{EM}_{\mathrm{macro}} &= \frac{100}{D}\sum_{d=1}^{D}\frac{\sum_{j\in\mathcal{D}_d} e_j}{|\mathcal{D}_d|}. \end{aligned} \tag{29}$$

Pooled EM gives more weight to larger datasets, while macro EM gives each dataset equal weight. Curves on different populations need not share the same absolute EM, even when their learning trends are similar.

Table 3: Evaluation populations. The coverage column identifies the corresponding comparison.

ID	Scope	Questions	Coverage
Test-7	Seven QA benchmarks	51,713	Table 1
Val-1000	Fixed validation subset	1,000	MemCredit, A1, A2
Val-1025	Four-dataset validation pool	1,025	MemCredit, A1, A2
Val-NQ	NQ validation	300	MemCredit, A1, A2
Val-Hotpot	HotpotQA validation	300	MemCredit, A1, A2
Diag-1152	HotpotQA, 2Wiki, MuSiQue	1,152	MemSearcher, MemCredit

C.4 Training and Evaluation Configuration

The actor is initialized from Qwen2.5-Instruct, and a frozen copy of the corresponding initial model supplies the reader. Accepted answers are available to the terminal evaluator and memory scorer during training; they are not appended to the actor's generation inputs. Table 4 summarizes the reported settings.

Both model sizes are evaluated on the seven benchmarks in Table 1, and the average is the arithmetic mean of the seven dataset EM scores. Local evaluation uses greedy decoding with one output per question.

The learning curves retain their training-step axes and are analyzed within each plotted comparison. They are not used to equate validation populations with the benchmark results. Results are from seed 123; variation across independent training seeds is not measured.

C.5 Ablation Definitions

The summary ablation in Table 2 uses Qwen2.5-3B-Instruct and reports the equal-weight macro EM over the same seven benchmarks and evaluation protocol as Table 1. The validation curves provide a separate view of component learning on the populations in Table 3.

Outcome-only. Setting $\lambda = 0$ removes retained-state credit while preserving the search–memory workflow and terminal supervision.

A1: Evidence-state scoring. A1 evaluates the prior memory together with the current observation:

$$g^E_{i,t} = \Phi_\psi(q, \mathrm{render}(M_{i,t-1}, o_{i,t}), \mathcal{Y}) - \Phi_\psi(q, M_{i,t-1}, \mathcal{Y}). \tag{30}$$

The rendering function constructs the reader's evidence-state input. The agent still writes a memory, and subsequent turns use that updated memory. A1 applies the same centering, scaling, and projection procedure to g^E and assigns the resulting local term to the complete search–memory conversation. This variant tests the choice between evaluating available evidence and evaluating the retained state.

A2: Memory-only credit. A2 retains the memory gain g^M but restricts the support of the local term:

$$\widehat{A}^{\mathrm{A2}}_{i,t,k} = \begin{cases} A^{\mathrm{out}}_i + \lambda z_{i,t}, & k \text{ is a memory-writing token,} \\ A^{\mathrm{out}}_i, & k \text{ is another generated token.} \end{cases} \tag{31}$$

Reasoning and query tokens continue to receive outcome supervision, and shared policy parameters remain trainable. Thus, A2 tests the span of local credit, not whether search actions receive any supervision. MemCredit combines retained-state scoring with local credit over the complete search–memory conversation.

C.6 Scoring and Numerical Conventions

The frozen reader scores accepted-answer tokens by Eq. 4; question and memory tokens supply context but are not included in the answer-token average. The initial memory is evaluated with the

Table 4: Reported training and benchmark-evaluation settings.

Setting	Value
Actor / frozen reader	Qwen2.5-3B-Instruct or Qwen2.5-7B-Instruct
Sibling trajectories	$G = 5$
Prompt batch / learning rate	256 / 10^{-6}
Local weight / box bound	$\lambda = 0.1$ / $c = 2$
Local normalization	Within-turn centering, shared MAD-based scale, joint zero-sum box projection
Benchmark reader context cap	2,048 tokens
Benchmark search budget	20 turns
Local evaluation decoding	Greedy, one output per question
Training seed	123

same reader and input construction as subsequent memories. Scores and constructed advantages are held fixed during policy optimization.

A local comparison group contains finite pre- and post-update scores for at least two siblings of the same sampled question occurrence and search turn. A trajectory that has already stopped contributes no update to subsequent turns. Centering uses only eligible siblings, and a shared positive MAD-based batch scale preserves their ordering. Singleton groups and a degenerate scale receive zero local credit. The joint projection can be computed by bisection using Eq. 22; independent coordinatewise clipping does not in general preserve the zero-sum constraint.

D Related Work

Reinforcement learning for search agents. Integrating external search into language-model reasoning has evolved from inference-time retrieval and tool use toward end-to-end policy optimization. Recent reinforcement-learning approaches train language models to decide when and how to search using final-answer feedback. R1-Searcher (Song et al., 2025b) adopts a two-stage outcome-based RL framework to elicit autonomous search behavior, while Search-R1 (Jin et al., 2025b) directly optimizes multi-turn interactions with a search engine using outcome rewards and retrieved-token masking. ReSearch (Chen et al., 2025) similarly treats search actions as part of the reasoning trajectory and learns reasoning-with-search without supervised intermediate traces. ZeroSearch (Sun et al., 2025) reduces the cost and instability of online search during RL by replacing the external search engine with a learned retrieval simulator and curriculum. More recently, Dr. Zero (Yue et al., 2026) studies data-free self-evolution of search agents through co-evolving question generation and solving. MemSearcher (Yuan et al., 2026) further introduces multi-context GRPO to jointly optimize reasoning, querying, and memory management under a compact-memory workflow. Despite their differences in training data and retrieval environments, these methods predominantly derive policy supervision from trajectory-level task outcomes. MemCredit instead focuses on how such outcome supervision can be complemented with local feedback about the information state produced by each search–memory interaction.

Memory-based search and language agents. Memory mechanisms have long been used to extend language agents beyond a fixed context window and preserve useful information across interactions. Generative Agents (Park et al., 2023) stores agents' experiences and retrieves relevant memories while periodically synthesizing higher-level reflections, whereas Reflexion (Shinn et al., 2023) maintains linguistic self-reflections as episodic memory for improving subsequent trials. MemoryBank (Zhong et al., 2023) develops a continually updated long-term memory with selective retention and forgetting, and MemGPT (Packer et al., 2023) formulates context management as a hierarchical memory system that moves information between memory tiers. LongMem (Wang et al., 2023) instead augments a frozen language model with a decoupled long-term memory retriever and reader. These approaches primarily address how information can be stored, retrieved, or compressed over

long interactions. For search agents, MemSearcher (Yuan et al., 2026) makes this bottleneck explicit by rewriting retrieved information into a compact textual memory after each search and conditioning subsequent turns on that memory. MemCredit builds on this setting but studies a different question: rather than redesigning the memory architecture, we optimize how the policy is rewarded for producing useful retained states. In particular, we distinguish evidence acquisition from evidence retention and directly evaluate how each memory update changes the task-relevant information available to later search decisions.

State utility and credit assignment. Assigning delayed outcome feedback to intermediate decisions is a longstanding problem in reinforcement learning. RUDDER (Arjona-Medina et al., 2018) addresses delayed rewards through return decomposition and reward redistribution, establishing the broader principle that trajectory outcomes can be transformed into more informative intermediate learning signals. In language-model reasoning, process supervision provides finer-grained feedback than outcome-only supervision: Lightman et al. (2023) demonstrate the effectiveness of step-level process rewards, while subsequent work explores automatically derived notions of intermediate progress. In particular, Setlur et al. (2024) defines process reward in terms of changes in the likelihood of eventual correctness, linking intermediate-state evaluation to step-level advantages. Extending this perspective to interactive agents, Zeng et al. (2025) formulates multi-turn tool use as a sequential decision process and studies turn-level advantage estimation rather than assigning a single trajectory-level signal to all interactions. MemCredit shares the goal of improving temporal credit resolution but specializes it to memory-based search. We define utility on the *retained memory state* rather than on a generic reasoning step or tool boundary, and assign the resulting group-relative signal to the complete search–memory conversation that jointly determines the evidence acquired and the information preserved. This connects state evaluation and action attribution within the particular information bottleneck induced by compact-memory search agents.

E Training Dynamics

E.1 Group Eligibility and Coefficient Activity

Outcome supervision is active in a sampled question group when at least one sibling has a nonzero outcome advantage. Local eligibility requires at least two finite memory gains at the same search turn. Eligibility alone does not ensure active local credit: equal gains, a degenerate scale, or $\lambda = 0$ can still give zero local coefficients.

For a sampled question group g, define

$$O_g = \mathbf{1}\{\exists i : A_i^{\text{out}} \neq 0\}, \qquad L_g = \mathbf{1}\{\exists (i,t) : \lambda z_{i,t} \neq 0\}. \tag{32}$$

Here L_g aggregates local activity over the eligible turns of that question group. When at least one group has tied outcomes, the conditional activity rate is

$$\frac{\sum_g (1 - O_g) L_g}{\sum_g (1 - O_g)}. \tag{33}$$

Its denominator is the number of outcome-tied groups, not the total number of sampled groups or eligible turns.

Figure 4(b) reports the training logger's aggregate valid-group percentage. This statistic is not identified with Eq. 33, which requires joint outcome and local coefficients for the same groups. The case in Appendix F.2 demonstrates local activity under an outcome tie, while the coefficient properties in Lemma 3 explain when such activity must occur. Neither the example nor the aggregate trace alone determines a population-wide conditional activity rate.

E.2 Memory-Gain Statistics

Raw memory gains measure the change in reader-based answerability before group normalization. For a nonempty set $\mathcal{U}$ of finite-scored transitions and threshold $\tau_M \geq 0$, define positive, negative,

and neutral rates by

$$P_+ = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \mathbf{1}\{g_u^M > \tau_M\},$$
$$P_- = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \mathbf{1}\{g_u^M < -\tau_M\}, \tag{34}$$
$$P_0 = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \mathbf{1}\{|g_u^M| \le \tau_M\}.$$

These disjoint categories satisfy $P_+ + P_- + P_0 = 1$. In particular, $P_+ + P_- = 1$ only when the neutral rate is zero. Invalid scores are excluded from $\mathcal{U}$, while finite zero gains are neutral under this definition. For pooled windows, category counts and transition counts are summed before division.

The offline comparison in Figure 4(a) shows a larger reported positive-gain fraction for MemCredit on HotpotQA, 2Wiki, and MuSiQue. The figure's overall bars are rounded to one decimal place, while its difference annotations retain greater precision. This measurement is based on memory transitions. A conditional evidence-retention rate would additionally require identifying transitions with positive evidence-state gain g^E, which is a different statistic.

Raw gain and relative credit also answer different questions. The sign of g^M indicates whether an update raises or lowers the reader potential. The sign of z depends on the sibling comparison and joint projection. An update with negative raw gain can therefore receive positive relative credit if other siblings perform worse. Likewise, a negative local term need not reverse a positive outcome coefficient; sign preservation holds under the bound in Lemma 3.

E.3 Outcome Reward and Response Length

Figure 5 shows MemCredit's terminal outcome reward for the 3B and 7B runs. Both improve rapidly early in training and fluctuate around a higher level later. The plotted reward evaluates the final outcome; it is separate from the reader potential, memory gain, and locally augmented advantage. Faint and solid lines distinguish the original and smoothed traces.

The response-length trace in Figure 4(c) measures tokens per training response. It shows that MemCredit is often more verbose than MemSearcher, particularly in the middle of training. This can affect training computation and does not directly measure test-time latency, total tokens per question, or the number of search turns. Although the frozen reader is absent at inference, a full efficiency comparison would also need these trajectory-level measurements.

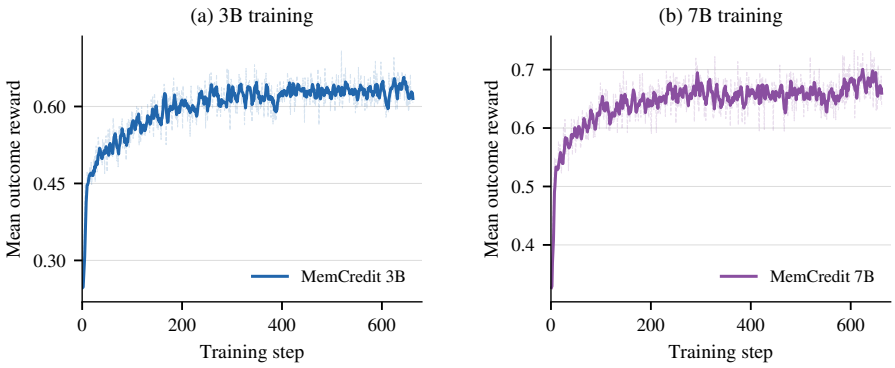

Figure 5: MemCredit terminal outcome reward at 3B and 7B. Faint and solid lines show the original and smoothed traces. The panels describe separate model-size runs.

F CASE STUDIES

The following examples connect the retained-state reward to memory writing and subsequent search. They illustrate specific behaviors rather than estimate their frequency across the evaluation population.

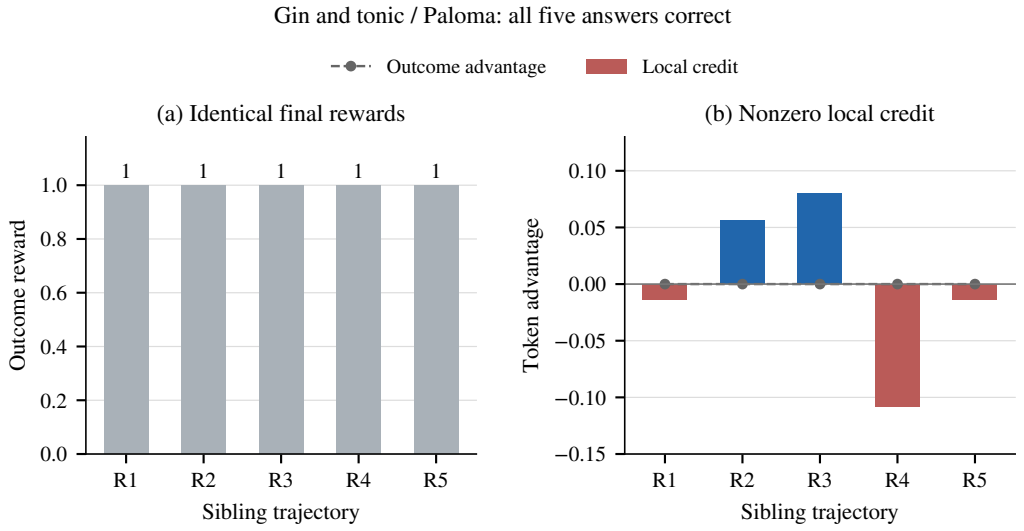

Figure 6: Local credit under tied terminal rewards. All five siblings receive reward 1 and outcome advantage 0, while their displayed local token coefficients differ in sign.

F.1 PRESERVING THE CORRECT ENTITY

Figure 1 compares memory writing for a question asking which of two films was released more recently. The shared evidence identifies *The Mountain Eagle* as a 1926 film and the other target film as a 1992 film, alongside an unrelated 2011 film with a similar title. MemSearcher writes about *The Eagle* (2011), losing the entity binding required by the question. MemCredit instead records the two relevant film-year pairs. The example illustrates why the state after writing is a useful evaluation target: the observation contains sufficient facts, but their presence alone does not ensure that the memory preserves the correct comparison.

F.2 LOCAL CREDIT UNDER TIED OUTCOMES

Figure 6 shows a training group for the Gin and tonic / Paloma question. All five sibling trajectories receive terminal reward 1. Their mean reward is therefore 1, and every outcome advantage in Eq. 1 is zero. Nevertheless, the displayed local coefficients contain both positive and negative values. Local credit thus distinguishes intermediate interactions within a group that terminal supervision treats identically.

The example concerns the assigned coefficients. It does not imply that all policy gradients vanish under outcome-only training, since regularization can remain active, or that a nonzero coefficient guarantees a nonzero aggregate gradient. Its role is to demonstrate the additional resolution of the retained-state signal.