HiDe: Hindsight Deletion of Environment-Substitutable Memory for Efficient LLM Agents

Anonymous authors
Paper under double-blind review

Abstract

Long-horizon LLM agents repeatedly carry historical reasoning and observations into subsequent decisions, even as environmental feedback refreshes task information and makes some records replaceable. Learning when to delete this history is difficult: task rewards provide little direct feedback on individual memory choices, while token savings alone cannot distinguish useful deletion from information loss. We introduce **HiDe (Hindsight Deletion)**, a framework that lets the agent's realized future teach its past memory decisions through **dual-level hindsight self-teaching**. Its **dual-view counterfactual scoring** evaluates the same next realized response under the reduced history and a matched context restoring only the newly deleted blocks. With the new observation shared across both views, score differences provide an approximate measure of the response's remaining dependence on removed history. HiDe combines this dependence signal with cumulative input savings into a shared hindsight utility, discounting savings when the response remains sensitive to the deleted content. At the decision level, the utility supplies dense credit to earlier DELETE/WAIT choices; at the trajectory level, it shapes task-gated rewards to favor strong task performance and useful memory reduction. Hindsight supervision requires only forward rescoring, without additional rollouts or a separate teacher model; the trained policy makes memory decisions from its current context alone. Across WebShop, BabyAI, SciWorld, TextCraft, and SearchQA with two model scales, HiDe achieves the best or tied-best task scores among compared methods in nine of ten settings. WebShop ablations support the contribution of hindsight credit to task performance and input efficiency, while independent Keep/Delete continuations and information-support analyses examine deletion consequences and replacement evidence.

1 Introduction

An agent may still need a fact after it no longer needs the history that first supplied it. Consider an LLM agent comparing products in WebShop (Yao et al., 2022). A search result initially provides attributes needed to choose a product. After the agent opens its detail page, the current observation may supply the same attributes, reducing the need to retain the earlier result. A comparison with another product, however, may still depend on information absent from that page. The relevant question is therefore whether a historical record continues to provide necessary support as interaction proceeds. Figure 1 illustrates how product information can recur across observations, motivating selective retirement of earlier reasoning and observation blocks.

This question matters because agent histories are repeatedly supplied as model inputs. ReAct-style agents accumulate reasoning, actions, and observations while obtaining new evidence from the environment (Yao et al., 2023). Retaining a block incurs a cost on every subsequent invocation, even when its information is already available elsewhere. Environment feedback can change that block's value: a previously essential record may become replaceable by the current observation and the remaining context. We call this task-dependent property *environment substitutability*. It motivates selective deletion, but does not make deletion straightforward. Observations can be incomplete, and a record unused at the next step may still support a later decision.

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.

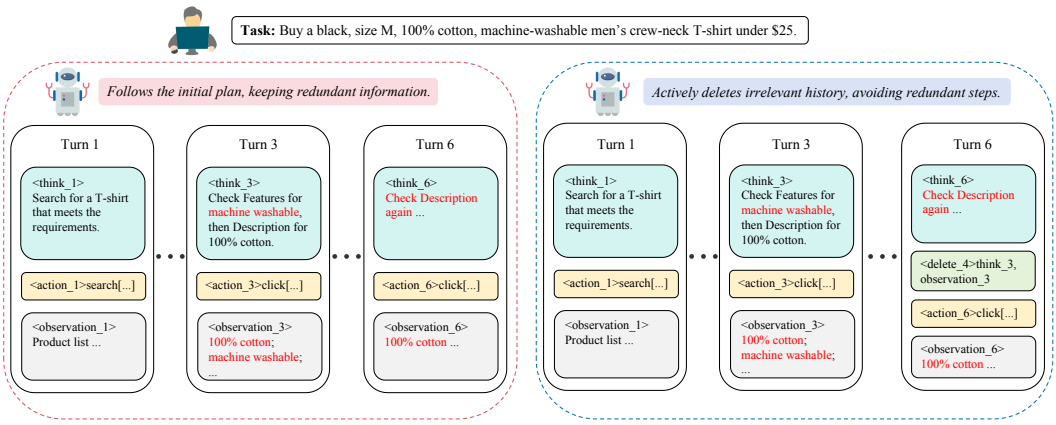

Figure 1: **Motivating historical deletion in a WebShop interaction.** Left: retaining the full history carries earlier search plans and product attributes into later turns. Right: selective deletion retires historical reasoning and observation blocks from subsequent inputs. Red text highlights repeated instructions or attributes; the green block denotes a deletion command. A record's value depends on the task and the information still available in current observations and retained history. This schematic illustrates the opportunity for context reduction, rather than a measured improvement in task success or action count.

Learning a deletion policy requires feedback that distinguishes useful context reduction from the removal of necessary information. Existing approaches compress prompts, maintain compact memory, or learn omission policies from task and efficiency rewards (Jiang et al., 2023; Zhou et al., 2026; Ning et al., 2026). Recent methods also assign counterfactual or hindsight credit to memory operations (Yan et al., 2026; Jia et al., 2026). For historical deletion, a specific opportunity remains: the agent's next response, generated during ordinary interaction, can reveal how strongly its realized behavior depends on the blocks it just removed. Such feedback could train the preceding deletion choice without constructing a separate target answer or executing an additional environment continuation for the local probe.

We introduce **HiDe (Hindsight Deletion)**, a framework for learning when historical reasoning and observations can be retired as environmental feedback refreshes task information. Its core mechanism, **dual-level hindsight self-teaching**, lets **the agent's realized future teach its past memory decisions**. After new feedback arrives, HiDe uses the agent's next reasoning and action as a fixed target for dual-view counterfactual scoring under matched contexts: the reduced history and a view restoring only the newly deleted blocks. With the new observation shared across both views, this comparison estimates how strongly the realized continuation still depends on the retired history after the environment has supplied fresh task information. This hindsight evidence guides individual deletion choices and rewards trajectories that combine strong task performance with useful memory reduction across subsequent model invocations.

Our contributions are threefold:

1. **The realized future as a self-teacher.** HiDe uses dual-view counterfactual scoring of the agent's next realized response under matched deletion and restoration contexts. By holding the new observation fixed and restoring only the newly deleted blocks, it estimates the same response's remaining dependence on that history. This provides approximate supervision for earlier memory decisions without extra rollouts.
2. **Dual-level hindsight RL for memory deletion.** We combine response dependence and cumulative input savings into a shared hindsight utility. At the decision level, it provides dense credit to earlier DELETE/WAIT choices; at the trajectory level, it shapes task-gated rewards to favor strong task performance and useful memory reduction.
3. **Empirical performance and analysis of deletion behavior.** Across five environments and two model scales, HiDe achieves the best or tied-best task scores among compared methods in nine of ten settings. We also analyze deletion timing and replacement evidence, distinguishing information available at the decision from evidence arriving afterward.

2 Related Work

Context reduction and agent memory. Prompt compression and memory management reduce the information carried between model invocations through different mechanisms. LLMLingua performs budgeted token compression (Jiang et al., 2023); MemGPT manages movement between memory tiers (Packer et al., 2023); MEM1 learns a compact recurrent memory that incorporates new observations (Zhou et al., 2026). ACM extends context management to learned compression, offloading, and retrieval (Li et al., 2026). HiDe focuses on removing previously generated reasoning and historical observations from future prompts. Its deletion operation requires neither a replacement summary nor a retrieval call. The research question concerns how to supervise those removals as the agent continues interacting.

Learning omission and deciding when to retire history. Agent-Omit combines cold-start demonstrations with omission-aware reinforcement learning, including full-trajectory and pre-omission partial-trajectory sampling and a task-gated efficiency reward (Ning et al., 2026). It learns to skip current thought generation and to remove historical observations. HiDe retains current reasoning generation and selects already generated reasoning and observation blocks for deletion. It builds on task-gated efficiency rewards by adding local feedback: the next response's dependence on the removed set supplies credit to the earlier deletion decision.

History can also be retired using changes in its measured influence. CommitKV compares page-deletion effects before and after tool feedback, then jointly checks candidate pages for KV-cache eviction (Huang et al., 2026). This addresses the distinction between a page whose role has completed and one that is temporarily inactive. HiDe instead learns a policy over explicit text deletions. Its two scoring views hold the new observation fixed, so the probe estimates dependence after feedback rather than the change in dependence caused by that feedback. This distinction also limits what HiDe's probe can establish about environment substitutability.

Hindsight credit for memory operations. HiMPO evaluates a memory write by comparing updated and previous memory through the likelihood of a target outcome, and filters memory-specific credit using hindsight relevance (Yan et al., 2026). Hindsight Memory-PRM uses retrieval and citation evidence together with controlled deletion-and-reanswer probes to supervise memory management (Jia et al., 2026). Both approaches assign credit to memory operations through outcome-conditioned or intervention-based evidence. HiDe uses a different supervision target: the agent's next realized reasoning and action. It restores exactly the set selected for deletion, holds the sampled response fixed, and sends a savings-weighted event utility back to the preceding deletion span. This local signal requires neither a target answer sequence nor a separately generated probe question; the broader training procedure still uses supervised initialization and environment rewards.

Response rescoring and policy-context alignment. On-policy self-distillation (OPSD) uses the same model as teacher and student under different contexts, with privileged information guiding learning on the student's sampled responses (Zhao et al., 2026). HiDe adapts this pattern into dual-view counterfactual scoring to assign credit to an earlier deletion decision, rather than distilling the scored response. OPID also rescores sampled responses using hindsight skills extracted from completed trajectories (Yang et al., 2026). MemOPD reconstructs original invocation states for aligned teacher scoring (Liu et al., 2026), while MemoryWalker studies the conditioning mismatch introduced by training on histories altered by eviction (J et al., 2026). HiDe preserves the actual invocation context for policy updates and uses restored contexts only to construct detached deletion targets for the preceding memory decision.

3 HiDe: Learning to Forget from the Realized Future

HiDe learns to retire historical reasoning and observations as new feedback and retained context take over their informational role. Its **dual-level hindsight self-teaching** uses the next realized response to supply decision-level credit and shape trajectory-level rewards. During training, dual-view counterfactual scoring evaluates that response with the same policy under matched deletion and restoration views. Response dependence and cumulative input savings define a shared utility for both objectives (Figure 2).

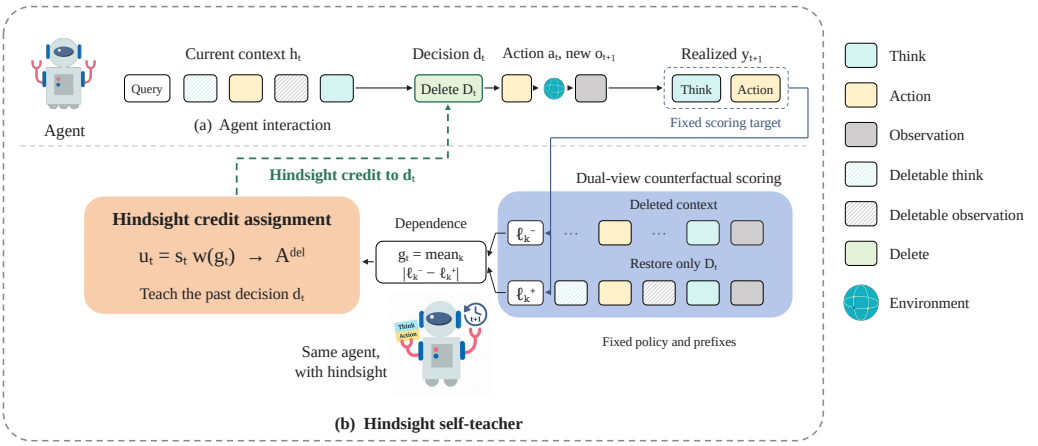

Figure 2: **HiDe interaction and hindsight self-teaching.** (a) At turn t, the agent selects historical blocks D_t for deletion, executes an action, and receives a new observation. Its next realized reasoning and action tokens form the fixed scoring target. (b) Dual-view counterfactual scoring evaluates those tokens with the fixed rollout policy under the deleted context and a matched context restoring only D_t, using identical response prefixes. The mean absolute log-probability gap g_t discounts the cumulative input-saving fraction s_t to form utility u_t. Group-relative utilities yield deletion advantages A^{del}, which supervise the preceding DELETE/WAIT decision d_t through the green dashed feedback path. This auxiliary update is combined with trajectory-level policy optimization; deployment requires neither restoration nor hindsight scoring.

3.1 Formulating Historical Memory Deletion

Let x denote a task instruction and o_t the observation available at turn t. Given its current prompt h_t, the policy generates $y_t = (r_t, d_t, a_t)$, consisting of reasoning, a memory decision, and an environment action. The memory decision selects a set D_t of historical reasoning and observation blocks for retirement, with $D_t = \varnothing$ denoting WAIT. Executing a_t yields o_{t+1}, and the next invocation uses a prompt from which the selected blocks have been removed. The agent can inspect those blocks while choosing d_t. The instruction, action history, protocol structure, and latest observation remain protected and cannot be selected for deletion.

Our objective is to preserve task performance while reducing cumulative input cost,

$$C_{\text{in}}(\tau) = \sum_{t=1}^{T} |h_t|_{\text{tok}}, \tag{1}$$

where T is the number of model invocations in trajectory τ. Retained blocks incur input cost at every subsequent invocation. Their value depends on current observations and remaining context; HiDe has no privileged access to internal environment state. Final rewards provide coarse trajectory feedback, whereas token savings alone cannot identify useful deletion. The next realized response supplies local evidence for the preceding decision.

3.2 The Realized Future as a Self-Teacher

For a nonempty deletion D_t followed by another invocation, let $h_{t+1}^{-D_t}$ be the actual prompt after retirement. During the ordinary rollout, the agent generates y_{t+1} from this reduced context. We then construct $h_{t+1}^{+D_t}$ by restoring exactly the newly deleted set at its original positions. Both contexts contain the same new observation o_{t+1} and preserve all earlier deletions, differing only in whether D_t is present.

Using the fixed rollout policy π_{old}, **dual-view counterfactual scoring** evaluates the **same realized response** under both contexts to assess its remaining dependence on D_t:

$$\begin{aligned} \ell^{-}_{t+1,k} &= \log \pi_{\text{old}}\Big(y_{t+1,k} \mid h^{-D_t}_{t+1}, y_{t+1,<k}\Big) \quad \text{(deleted view)}, \\ \ell^{+}_{t+1,k} &= \log \pi_{\text{old}}\Big(y_{t+1,k} \mid h^{+D_t}_{t+1}, y_{t+1,<k}\Big) \quad \text{(restored view)}, \\ g_t &= \frac{1}{|\mathcal{K}_{t+1}|} \sum_{k \in \mathcal{K}_{t+1}} \Big|\ell^{-}_{t+1,k} - \ell^{+}_{t+1,k}\Big|. \end{aligned} \tag{2}$$

Here, $\mathcal{K}_{t+1}$ contains all reasoning and action token positions in the next realized response, excluding the memory-decision span. We average the absolute log-probability differences over this combined token set, giving each token equal weight. When several blocks are retired together, this comparison evaluates their joint effect without assuming additive contributions from individual blocks.

Dual-view counterfactual scoring without extra rollouts. We adopt the privileged-context scoring pattern of on-policy self-distillation (OPSD; Zhao et al., 2026): the same model evaluates its own sampled response under different conditioning contexts. Here, the counterfactual intervention restores only D_t, while holding the realized response and all token prefixes fixed. The paired log-probabilities are obtained through teacher-forced forward evaluation, without additional continuation sampling, environment interaction, or a separate teacher model.

The gap g_t measures dependence of this sampled continuation on the restored history after feedback. A low gap does not certify later safety or identify environment substitution as its cause.

3.3 Decision-Level Hindsight Self-Teaching

We combine response dependence with cumulative input savings, then assign the resulting utility to the preceding memory decision.

Measuring the savings of a deletion. Let b_t be the number of tokens removed by D_t, and let $n_t = T - t$ count the subsequent invocations whose prompts omit them. Removing a block saves its tokens repeatedly: the cumulative input reduction attributable to this decision is $b_t n_t$. We normalize this reduction by the input cost of retaining all historical blocks along the same realized trajectory:

$$C_{\text{full}}(\tau) = \sum_{j=1}^{T} \left(|h_j|_{\text{tok}} + \sum_{k<j} b_k \right) = C_{\text{in}}(\tau) + \sum_{k=1}^{T} b_k n_k, \qquad s_t = \frac{b_t n_t}{C_{\text{full}}(\tau)}. \tag{3}$$

The baseline C_{full} adds back all avoided token exposures along the observed trajectory, without a separate full-history rollout. Thus, s_t measures repeated input savings, and $\sum_t s_t = (C_{\text{full}} - C_{\text{in}})/C_{\text{full}}$.

Turning savings into hindsight utility. A larger saving should receive less credit when the next response remains sensitive to the removed history. We therefore multiply the saving fraction by a dependence-based discount that reflects the measured response sensitivity:

$$u_t = s_t\, w_t, \qquad w_t = w(g_t), \qquad w(g) = \begin{cases} 1, & g \le 0.2, \\ \dfrac{1-g}{0.8}, & 0.2 < g < 1, \\ 0, & g \ge 1. \end{cases} \tag{4}$$

The discount changes credited utility, not measured savings, and introduces no separate negative deletion penalty. WAIT and deletions without a valid next-response probe receive zero utility; only realized deletion sets are probed. Appendix A.2 gives a numerical example.

From utilities to relative advantages. We assign each utility to the decision d_t that caused the deletion, even though its gap is measured from response y_{t+1} and its savings accumulate afterward. Within rollout group q for a common task, we compare decision utilities across trajectories at the same invocation index t. Let $\mathcal{E}_{q,t}$ contain the eligible nonterminal DELETE/WAIT decisions actually

observed at that index; t counts model invocations, not deletion events. We standardize their utilities to form a relative teaching signal:

$$\widetilde{A}^{\mathrm{del}}_{i,t} = \frac{u_{i,t} - \bar{u}_{q,t}}{\sigma_{q,t} + 10^{-6}}. \tag{5}$$

Here, $\bar{u}_{q,t}$ and $\sigma_{q,t}$ are the mean and standard deviation over $\mathcal{E}_{q,t}$. Bounded rescaling produces $A^{\mathrm{del}}_{i,t}$, reinforcing above-average utility and discouraging below-average utility. Appendix A.1 specifies rescaling and edge cases.

3.4 Trajectory-Level Hindsight Self-Teaching

Trajectory-level teaching reuses the same utilities to reward task performance and useful memory reduction. Both levels jointly train a single policy.

Trajectory rewards and policy optimization. For a deletion-enabled trajectory τ, accumulated deletion utility augments the task score:

$$R_{\mathrm{traj}}(\tau) = R_{\mathrm{task}}(\tau)\left[1 + 0.2\sum_{t} u_t\right]. \tag{6}$$

Here, $R_{\mathrm{task}} \in [0, 1]$ is the environment score. Rewards yield group-relative trajectory advantages A^{traj}_i following GRPO (Shao et al., 2024); Appendix A.2 specifies the implementation cap and normalization used to construct these trajectory-level advantages.

Joint training with trajectory and deletion objectives. Trajectory advantages A^{traj}_i supervise all valid reasoning, memory-decision, and action tokens through $\mathcal{L}_{\mathrm{traj}}$. Deletion advantages $A^{\mathrm{del}}_{i,t}$ supply an additional update only to the DELETE/WAIT tokens at turn t through $\mathcal{L}_{\mathrm{del}}$. Within $\mathcal{L}_{\mathrm{del}}$, token losses are averaged within each decision span and then across decisions. Both objectives train the same policy, using the following relative signals:

$$\widetilde{A}^{\mathrm{traj}}_i = \frac{R_{\mathrm{traj}}(\tau_i) - \bar{R}_q}{\sigma^R_q + 10^{-6}}, \qquad \widetilde{A}^{\mathrm{del}}_{i,t} = \frac{u_{i,t} - \bar{u}_{q,t}}{\sigma_{q,t} + 10^{-6}},$$

$$\mathcal{L}_{\mathrm{total}}(\theta) = \underbrace{\mathcal{L}_{\mathrm{traj}}\left(\theta; A^{\mathrm{traj}}_i\right)}_{\substack{\text{Task completion} \\ \text{and deletion benefits}}} + 0.25 \quad \underbrace{\mathcal{L}_{\mathrm{del}}\left(\theta; A^{\mathrm{del}}_{i,t}\right)}_{\substack{\text{Credit for} \\ \text{DELETE/WAIT decisions}}}. \tag{7}$$

Here, $\bar{R}_q$ and σ^R_q denote the mean and standard deviation of trajectory rewards within group q. The standardized advantages are rescaled to obtain A^{traj} and A^{del} for their respective objectives. Advantage rescaling and PPO optimization details are provided in Appendix A.

The documented WebShop training uses eight groups of eight trajectories: six deletion-enabled groups and two no-delete groups. Only deletion-enabled groups receive the bonus and auxiliary loss; initialization and group supervision are detailed in Appendix A.2.

Online deployment. At deployment, the agent produces one response per turn, and valid retirement commands remove selected blocks from subsequent prompts. Neither restoration copies nor hindsight scoring are required. Training converts future evidence into memory decisions made from the current context.

4 Experiments

We evaluate task performance and token cost, ablate training components, and analyze deletion behavior. RQ5 isolates deployment input cost using a paired evaluation. Protocols and outstanding checks are detailed in Appendix B.

Table 1: **Task performance and token cost across five environments.** Score is mean native task score on a 0–100 scale; Tok. is mean cumulative input plus output tokens per task, summed across calls. Bold marks each group's highest scores and lowest token costs, including ties.

	WebShop		BabyAI		SciWorld		TextCraft		SearchQA	
Method	Score ↑	Tok. ↓	Score ↑	Tok. ↓	Score ↑	Tok. ↓	Score ↑	Tok. ↓	Score ↑	Tok. ↓
Qwen3-4B										
MemAgent	4.81	10,497	56.94	11,486	4.67	11,783	48.00	11,429	14.92	5,507
DEPO	7.93	10,681	60.36	12,948	8.43	13,621	53.00	13,587	16.42	6,989
Agent-Omit	15.94	**7,241**	71.58	7,543	9.86	**8,619**	**73.00**	**5,218**	21.23	3,229
HiDe SFT	40.73	18,376	64.33	4,543	23.13	12,845	65.00	16,625	22.00	5,125
HiDe	**52.83**	10,087	**77.99**	**3,857**	**34.48**	10,867	**73.00**	9,804	**22.25**	**3,125**
Qwen3-8B										
MemAgent	5.73	10,237	62.17	11,083	5.83	11,864	51.00	11,142	16.21	5,496
DEPO	9.98	10,094	67.13	12,568	10.84	13,421	59.00	13,017	18.13	6,774
Agent-Omit	20.63	8,991	78.26	6,508	16.27	9,826	**80.00**	**7,184**	24.08	**4,469**
HiDe SFT	49.08	9,940	66.97	4,482	27.25	11,916	63.00	12,163	24.00	7,856
HiDe	**66.93**	**6,948**	**84.33**	**3,482**	**54.15**	**8,942**	75.00	9,008	**30.00**	5,953

4.1 Experimental Setup

We compare MemAgent (Yu et al., 2026), DEPO (Chen et al., 2026), Agent-Omit (Ning et al., 2026), HiDe SFT, and HiDe using Qwen3-4B/8B. Benchmark and model sources appear in Appendix B. Pass@1 is mean native task score on a 0–100 scale and can include partial rewards. Avg Tok. averages cumulative input plus output tokens per task, summed across model calls. Ablations report cumulative *input* tokens only. Neither metric is final retained context length. The same-trajectory baseline C_{full} (Eq. (3)) is distinct from an independently executed no-delete rollout.

4.2 RQ1: How Does HiDe Compare with Existing Methods?

Performance and training-stage comparison. HiDe obtained the highest score in eight of ten backbone–environment settings and tied in one more (Table 1). The largest gains over Agent-Omit occurred on WebShop and SciWorld: 36.89 and 24.62 points at 4B, and 46.30 and 37.88 points at 8B, respectively. Relative to HiDe SFT, it improved scores by 0.25–26.90 points and reduced total tokens by 15.1–45.1% across all ten settings. On Qwen3-8B WebShop, the score rose from 49.08 to 66.93 while tokens fell from 9,940 to 6,948. On SciWorld, the corresponding changes were 27.25 to 54.15 and 11,916 to 8,942.

Performance–cost trade-offs. Against Agent-Omit, HiDe achieved higher scores with fewer tokens on 4B BabyAI and SearchQA, and 8B WebShop, BabyAI, and SciWorld. Higher scores on 4B WebShop and SciWorld and 8B SearchQA required more tokens. TextCraft remained unfavorable: HiDe used 87.9% more tokens at the same 4B score, and 25.4% more tokens with a 5.00-point lower 8B score. The gains therefore depended on the environment. These method comparisons do not isolate deployment-time deletion effects; RQ5 reports a paired delete/no-delete comparison from the same WebShop checkpoint.

4.3 RQ2: Which Training Components Matter?

Without the auxiliary loss (no hindsight), score fell 2.75 points and input tokens rose 16.3%; the gated bonus remained. Removing no-delete groups (no control) reduced score by 3.37 points and increased inputs by 95.5%. The no g_t variant used 4,827 versus 5,553 input tokens but scored 8.84 points lower; its precise intervention remains unconfirmed (Figure 3a).

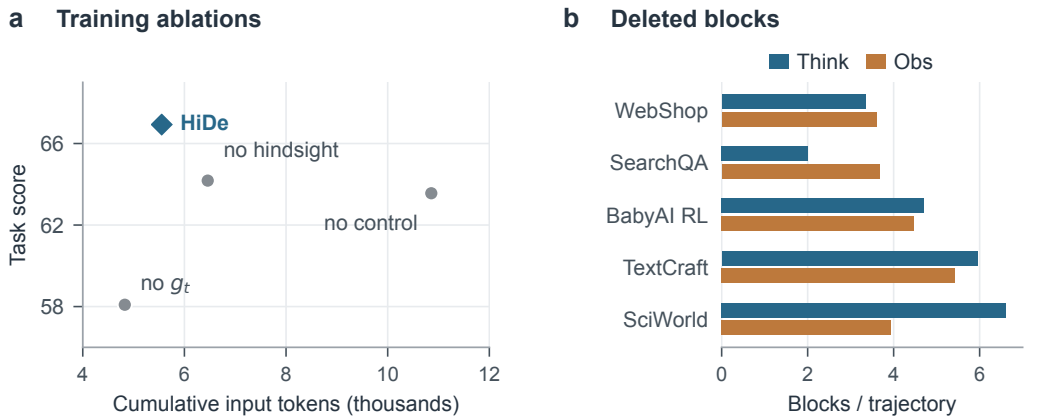

Figure 3: **Training ablations and deletion counts.** (a) WebShop 8B score versus cumulative input cost (point estimates). (b) Mean deleted Think/Obs blocks per trajectory; only BabyAI has an explicit RL stage label. Exact values and switches are in Tables 3 and 4.

4.4 RQ3: Does the Gap Predict Deletion Consequences?

We evaluated 367 deletion events using independent paired Keep/Delete continuations, with three outcome-seed pairs per event (Appendix B.2). Figure 4 groups events using the thresholds in Eq. (4).

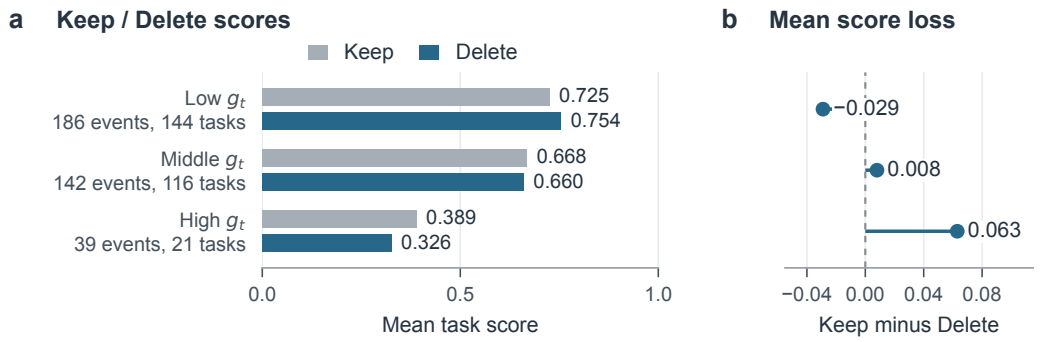

Figure 4: **Deletion consequences across hindsight-gap groups.** WebShop 8B, checkpoint-250: low $g_t \leq 0.2$, middle $0.2 < g_t < 1$, high $g_t \geq 1$. Three paired continuations per event; task counts may overlap across bins. Scores and losses use the 0–1 scale; no intervals are available. Full values are in Table 2.

Mean deletion loss increased from -0.029 to 0.008 and 0.063 across the low, middle and high groups, supporting g_t as a group-level risk indicator. These point estimates do not establish significance or individual deletion safety; task-clustered uncertainty and within-bin damage rates remain needed to assess uncertainty and variation in deletion outcomes.

4.5 RQ4: What Is Deleted, and What Supports Its Removal?

Both block types were deleted in every configuration (Figure 3b). Observation counts were higher in WebShop and SearchQA; the reverse held in BabyAI RL, TextCraft, and SciWorld. Counts do not measure timing or token savings.

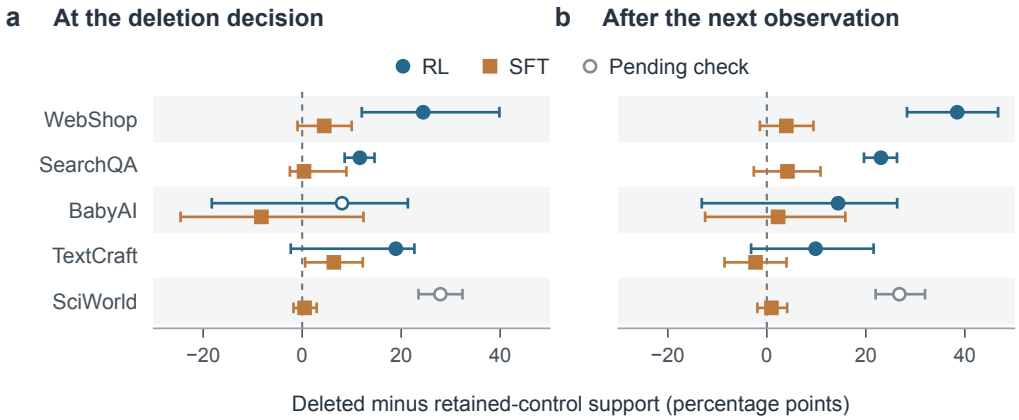

Figure 5: **Information support for deleted versus retained blocks.** Differences and reported 95% intervals at decision time (a) and after feedback (b). Hollow points flag pending BabyAI/SciWorld RL checks; grey SciWorld RL points are excluded from interpretation. Full rates and provenance are reported in Table 5 for all listed configurations.

WebShop RL and SearchQA RL had positive contrasts at both times, with decision-time estimates of 24.44 and 11.65 percentage points. RL estimates exceeded SFT, but no direct RL–SFT interval was supplied. TextCraft SFT had a positive interval only at decision time; BabyAI and TextCraft RL intervals included zero.

Support includes environmental evidence, retained history, and obsolete content. Wider contrasts need not reflect new information: WebShop RL control support fell from 68.64% to 54.31%, while deleted-block support stayed near 93%. These diagnostics do not establish deletion motives or downstream safety for individual deletion decisions or later task steps.

4.6 RQ5: Does Deletion Reduce Deployment Input Cost?

On 200 paired WebShop tasks, deletion reduced cumulative input tokens by 28.2%, with a 0.25-point lower mean score. Full results and implementation details are in Appendix B.3–B.4; runtime and training stability remain unmeasured.

5 Conclusion

HiDe turns the realized future of an interaction into supervision for historical memory deletion. By combining response dependence with cumulative input savings, its decision-level and trajectory-level objectives train an agent to select deletions that earn hindsight credit while completing tasks. Across five environments and two model scales, HiDe achieved the highest score in eight of ten settings and tied for the highest in one more. On Qwen3-8B, it exceeded Agent-Omit by 46.30 points on WebShop and 37.88 points on SciWorld, achieving higher scores with fewer total tokens on WebShop, BabyAI, and SciWorld. Paired WebShop evaluation further showed 28.2% lower cumulative input use when deletion was enabled. Independent Keep/Delete continuations connected larger hindsight gaps with greater average deletion loss, providing evidence for the gap's role in guiding selective deletion. Once trained, the agent manages history through its normal responses without additional hindsight scoring at deployment. Together, these results establish hindsight-guided deletion as an effective approach to learning compact interaction histories for LLM agents.

AI Use Statement

AI tools were used to aid and polish writing, and to support retrieval and discovery of related work. All scientific claims, experiments, analyses, and final manuscript decisions were reviewed and controlled by the authors.

REFERENCES

Sirui Chen, Mengshi Zhao, Lei Xu, Yuying Zhao, Beier Zhu, Hanwang Zhang, Shengjie Zhao, and Chaochao Lu. DEPO: Dual-efficiency preference optimization for LLM agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30279–30287, 2026. doi: 10.1609/aaai.v40i36.40279.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJeXCo0cYX.

Weizhong Huang, Jinchao Zhang, and Xiawu Zheng. CommitKV: Lifecycle-aware KV cache compression via commit transitions for multi-turn agents. *arXiv preprint arXiv:2608.07855*, 2026. URL https://arxiv.org/abs/2608.07855.

Zinco J, Xunjie Zhu, Shen Huang, Zhenyi Wang, Pengjun Xie, and Jieping Ye. MemoryWalker: Stop training agents on contexts they never saw. *arXiv preprint arXiv:2609.00865*, 2026. URL https://arxiv.org/abs/2609.00865.

Haoxuan Jia, Yang Liu, Yingguang Yang, Yancheng Chen, Chongyang Zhang, Hao Zheng, Qian Li, Yulin Huang, Jianshen Zhang, Yongzhi Qi, Shang Luo, Kefu Xu, Hao Peng, Junyu Lu, Du Cheng, Philip S. Yu, and Bin Chong. Hindsight memory-PRM: Supervising memory management with auditable hindsight credit. *arXiv preprint arXiv:2608.29605*, 2026. URL https://arxiv.org/abs/2608.29605.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825.

Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. ACM: Agentic context management for long horizon tasks. *arXiv preprint arXiv:2607.23809*, 2026. URL https://arxiv.org/abs/2607.23809.

Zhiyuan Liu, Tinghong Ye, Chenghao Liu, Yizhuo Li, and Songfang Huang. MemOPD: On-policy distillation through memory state alignment for long-horizon agents. *arXiv preprint arXiv:2608.07068*, 2026. URL https://arxiv.org/abs/2608.07068.

Yansong Ning, Jun Fang, Naiqiang Tan, and Hao Liu. Agent-Omit: Adaptive context omission for efficient LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YX4rgoI8pB.

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.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. ADaPT: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4226–4252. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.264.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.775.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1355.

Jiangze Yan, Yi Shen, Wenjing Zhang, Jieyun Huang, Zhaoxiang Liu, Ning Wang, Kai Wang, and Shiguo Lian. HiMPO: Hindsight-informed memory policy optimization for less-entangled credit in long-horizon agents. *arXiv preprint arXiv:2606.16285*, 2026. URL https://arxiv.org/abs/2606.16285.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026. URL https://arxiv.org/abs/2606.26790.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022. doi: 10.52202/068431-1508.

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*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4264ee4376776907c0b87ed70b959585-Abstract-Conference.html.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5fc8b3bdfbb9167b5144df5d3fae4616-Abstract-Conference.html.

A Optimization Details

A.1 Deletion Decision Objective

Let $\mathcal{E}$ collect eligible nonterminal decision events (i, t) in the batch, and let $M_{i,t}$ be the token span expressing the decision: deletion commands for DELETE, or the boundary marker representing WAIT. The deletion advantages are constructed within each task group and invocation index as described in Section 3.3. Thus, an observed WAIT can enter the comparison with zero utility, whereas a missing decision in a shorter trajectory contributes no event. Matching invocation indices does not imply matched environment states.

Within each $\mathcal{E}_{q,t}$, we apply a common rescaling only when the largest absolute standardized advantage exceeds 0.2, bringing it to 0.2. Decisions at the group mean, singleton groups, and groups with identical utilities receive zero advantage.

Let $\rho_{i,t,k}(\theta)$ be the current-to-old policy probability ratio for token k, evaluated under the actual prompt $h_{i,t}$ and generated prefix. With the PPO clipped surrogate (Schulman et al., 2017) $S(\rho, A) = \min\{\rho A, \mathrm{clip}(\rho, 1-\eta, 1+\eta)A\}$ and $\eta = 0.2$, the deletion objective is

$$\mathcal{L}_{\mathrm{del}}(\theta) = -\frac{1}{|\mathcal{E}|} \sum_{(i,t)\in\mathcal{E}} \frac{1}{|M_{i,t}|} \sum_{k\in M_{i,t}} S\big(\rho_{i,t,k}(\theta), A^{\mathrm{del}}_{i,t}\big)\,. \tag{A.1}$$

Averaging within each span gives each decision equal auxiliary weight regardless of command length. The loss is zero if no eligible events exist. Paired scores, savings, and advantages are detached targets; restoration views construct the targets but do not replace the original policy-conditioning context. The reference objective includes neither response-side distillation nor a reference-policy KL penalty.

A.2 Trajectory Reward and Advantage Normalization

Equation (6) presents the reward before applying the implementation cap. During training, the deletion bonus is limited to 10% of the task score, so the reward used for trajectory optimization is

$$R_{\mathrm{traj}}(\tau) = R_{\mathrm{task}}(\tau)\left[1 + \min\left\{0.1,\ 0.2\sum_t u_t\right\}\right].$$

The bonus vanishes when $R_{\mathrm{task}}(\tau) = 0$.

For each task group q, let $\bar{R}_q$ and σ_q^R be the mean and standard deviation of these capped complete-trajectory rewards. Each trajectory contributes one scalar reward, regardless of its number of invocations. We compute

$$\widetilde{A}_i^{\mathrm{traj}} = \frac{R_{\mathrm{traj}}(\tau_i) - \bar{R}_q}{\sigma_q^R + 10^{-6}}. \tag{A.2}$$

If the largest absolute standardized advantage in a group exceeds 0.2, all advantages in that group are scaled by the same factor so that the largest absolute value becomes 0.2; otherwise they are unchanged. The resulting A_i^{traj} is shared across all valid response tokens of τ_i in the PPO objective $\mathcal{L}_{\mathrm{traj}}$ with clipping parameter $\eta = 0.2$. Singleton groups and groups with identical rewards receive zero advantages. This trajectory-level comparison requires no alignment of intermediate decisions or environment states.

Training setup. Qwen3-8B (Yang et al., 2025) is initialized with Qwen3-235B demonstrations; RL hindsight scoring uses the agent's own policy. Each batch contains eight groups of eight trajectories: six deletion-enabled and two full-history groups with deletion effects disabled. Both receive trajectory supervision. Only deletion-enabled groups receive the bonus and auxiliary loss; full-history groups use $R_{\mathrm{traj}} = R_{\mathrm{task}}$.

Input-savings example. Deleting 1,000 tokens from five subsequent prompts saves 5,000 input tokens. With $C_{\mathrm{full}} = 50{,}000$ and $g_t = 0.6$, $s_t = 0.1$, $w_t = 0.5$, and $u_t = 0.05$. The actual saving is 10%, with half credited as utility; earlier deletion increases repeated savings when dependence remains low, because the removed blocks are omitted from more subsequent prompts.

B Experimental Details and Pending Evaluations

B.1 Evaluation Configuration and Evidence Status

Method and benchmark sources. The baseline methods are MemAgent (Yu et al., 2026), DEPO (Chen et al., 2026), and Agent-Omit (Ning et al., 2026); the backbone models are from the Qwen3 family (Yang et al., 2025). Our environments are WebShop (Yao et al., 2022), BabyAI (Chevalier-Boisvert et al., 2019), ScienceWorld (abbreviated SciWorld; Wang et al., 2022), TextCraft introduced in ADaPT (Prasad et al., 2024), and the AgentGym search environment (Xi et al., 2025), denoted SearchQA here. These citations identify the original methods and environments; the exact SearchQA dataset mixture and split remain to be documented for this evaluation.

The supplied WebShop 8B configuration initializes the policy with demonstrations from Qwen3-235B-A22B and identifies v29 checkpoint-250 for 200 paired tasks. The WebShop implementation is detailed in Appendix B.4; final checkpoints, splits, sample counts, baseline configurations, and evaluation seeds remain to be specified for the other environments. WebShop training partitions and reference-policy KL settings are not assumed for ported environments. Main and ablation results are point estimates; task-level or multi-seed uncertainty remains unavailable. The paired WebShop evaluation in Appendix B.3 compares deletion settings under the same trained policy.

The ablation switches follow the author-supplied table. The precise replacement for the gated deletion bonus in no g_t remains unspecified. An earlier no control configuration inherited settings that also disabled the local auxiliary loss, unlike the current table. Effective configuration provenance and checkpoint-selection rules therefore remain pending; the historical configuration does not override the supplied results.

Only BabyAI explicitly specifies RL in the deletion-count data. Stage and checkpoint labels for the other four rows remain pending, as do block availability, trajectory-length, and token denominators needed to normalize counts. Support denominators, annotation calibration, and updated resampling metadata also remain to be documented. Reported contrasts are preserved separately from rounded support percentages. SciWorld RL replay provenance and the BabyAI RL decision-time interval require the checks described in Table 5.

B.2 RQ3: Independent Deletion Consequences

We evaluated WebShop using Qwen3-8B checkpoint-250 under the independent Keep/Delete continuation protocol. Each selected event was replayed to the same post-action environment state. Delete retained the reduced history; Keep restored only D_t. The probe response was scored but not executed. Outcome continuations used $K = 3$ paired seeds, disjoint from the probe seeds, and disabled the effects of later deletion requests in both branches. The event-level consequence was $\Delta_i = K^{-1} \sum_{k=1}^{K} (R_{i,k}^{\text{keep}} - R_{i,k}^{\text{delete}})$; positive values indicate lower task scores after deletion.

The supplied summary contains 186, 142, and 39 events in the low, middle, and high gap bins, involving 144, 116, and 21 tasks within those bins. These task counts cannot be summed to obtain distinct tasks without task IDs. Mean scores and losses use the 0–1 native-score scale, rather than the main table's 0–100 scale. The low-gap loss is -0.029. The other losses are 0.008 and 0.063; all three agree with the reported Keep-minus-Delete mean scores.

The mean loss is not the fraction of harmful deletions, and different gap bins also differ in mean Keep score. These summaries therefore support a within-bin Keep/Delete comparison but not event-level risk calibration or a claim that gap alone determines task difficulty. Event records are needed for task-clustered uncertainty, damage rates ($\Delta_i > 0.05$), and risk at fixed acceptance rates. The original 40-task development pilot remains a separate protocol stage from the 200-task main evaluation.

Data pending: event/task IDs, paired outcome records, replay integrity and completion logs, and task-clustered uncertainty estimates for the paired Keep/Delete comparisons.

B.3 RQ5: Deployment Input Cost under Paired Evaluation

We compared deletion-enabled and no-delete execution of the same WebShop Qwen3-8B v29 checkpoint-250 policy on the frozen Agent-Omit 200-task test set. Each task was evaluated under both settings with paired sampling conditions, giving 400 trajectories. The no-delete arm disabled deletion effects while retaining the trained policy. It therefore differs from both the HiDe SFT baseline and the training ablation that removes no-delete groups.

Table B.1: **Paired WebShop deployment evaluation.** Both arms use Qwen3-8B v29 checkpoint-250 on the same 200 tasks. Scores are mean native task scores multiplied by 100; token counts are mean cumulative input tokens per task and exclude generated tokens. The last row subtracts no-delete from deletion-enabled.

Execution setting	Mean score	Cumulative input tokens
No-delete	67.18	7,734.4
Deletion-enabled	66.93	5,552.7
Difference	-0.25	$-2,181.7$ (-28.2%)

Observed performance–cost trade-off. Deletion reduced input tokens by 28.2%, calculated as $(7{,}734.4 - 5{,}552.7)/7{,}734.4$, while the mean score was 0.25 points lower (0.6693 versus 0.6718 on the native 0–1 scale). This comparison isolates the execution setting under a fixed trained policy; it does not estimate the effect of removing deletion during training. The supplied report contains aggregate means without paired uncertainty. The small observed score difference therefore does not establish statistical equivalence or non-inferiority, nor does it characterize outcomes on individual tasks. The full-success rate is a separate metric and was not supplied here.

Evaluation protocol and metric scope. Both arms used eight H20 GPUs, a maximum of 12 environment actions, up to 4,096 generated tokens per turn, temperature 0.6, top-p 0.95, and top-k 20. They shared the WebShop policy prompt, Think/Obs memory mode, and lenient action recovery. Input-token accounting sums prompt tokens across every model call. It differs from the input-plus-generation metric in Table 1 and from final retained history length. The no-delete arm is an independently executed rollout, whereas C_{full} in Eq. (3) reconstructs a cost baseline along the realized trajectory by adding back the input exposures avoided by deletion.

Compute costs and remaining evaluations. Hindsight scoring constructs detached training targets; deployment does not require that scoring procedure. This does not establish a measured latency or GPU-hour reduction. The implementation record gives hardware and schedules but no wall-clock durations, memory measurements, checkpoint performance curves, or repeated training runs. Training stability, total training cost, and delayed-dependence failure behavior consequently remain open evaluations. Task-level paired scores and token counts are also needed to quantify uncertainty in the deployment comparison and characterize variation across the evaluated tasks.

B.4 WebShop Implementation and Reproducibility Details

The supplied implementation record identifies Qwen3-235B-A22B as the demonstration teacher. SFT used 1,380 training and 169 validation records; the converted SFT policy initialized HiDe RL. SFT ran for three epochs, and the RL schedule specified 300 outer steps.

Table B.2: **WebShop Qwen3-8B implementation settings.** The reported configuration applies to WebShop; training-group partitions and reference-policy KL settings may differ in the other environments and must be checked against each environment's configuration.

Setting	SFT	HiDe RL
Hardware	8× H20	2 × 8 L20X
Training schedule	3 epochs	300 outer steps

Interaction and hindsight settings. RL used the same sampling and action limits as the paired evaluation. Generation returned one sequence with one beam and stopped after `</answer>`, retaining the stop string. Multiple deletions were credited as one atomic set. Maximum-length response masking was enabled. The hindsight thresholds were $\epsilon = 0.2$ and $\delta = 1.0$; the saving coefficient was $\mu = 0.2$, the trajectory bonus cap was 10% of task score, and the local auxiliary weight was 0.25. Group-normalized advantages were bounded at 0.2. Response-side OPD had coefficient zero, and this WebShop configuration used no reference-policy KL penalty. Some transferred environments used deletion-only training and a KL-to-SFT anchor; the WebShop settings should not be applied to them without their configurations.

Source records and independent validation. The supplied record links the paired result to the official WebShop Agent-Omit test configuration and its v29 checkpoint-250 evaluation report. Its RQ3 pilot used 32 single-GPU workers on L20X devices and a 40,960-token inference context limit; this worker count describes evaluation parallelism. The pilot's 40 development tasks were disjoint from the official 200-task test set. The larger supplied RQ3 result summary is described separately in Appendix B.2.

C Complete Validation, Ablation, and Mechanism Results

Tables 2–5 retain every supplied configuration and numerical value underlying Figures 3–5. The ablation input-token metric excludes generation; the main table's Avg Tok. includes both input and generation accumulated over all model calls in a trajectory.

Table 2: **Deletion outcomes stratified by g_t.** WebShop, Qwen3-8B, checkpoint-250; three paired continuations per event. Scores use the 0–1 scale; loss is Keep minus Delete.

	Low g_t $g_t \leq 0.2$	Middle g_t $0.2 < g_t < 1.0$	High g_t $g_t \geq 1.0$
Deletion events	186	142	39
Tasks within bin	144	116	21
Mean Keep score ↑	0.725	0.668	0.389
Mean Delete score ↑	0.754	0.660	0.326
Mean score loss ↓	−0.029	0.008	0.063

Table 3: **Training-component ablations on WebShop with Qwen3-8B.** On and Off denote the reported presence and absence of each component; the gated bonus uses g_t. Score is the mean native task score on a 0–100 scale. Token counts are mean cumulative *input* tokens per task, excluding generated tokens. Bold indicates the highest score and lowest input-token count.

Configuration	Local decision auxiliary loss	No-delete training groups	Gap-gated deletion bonus	Score ↑	Cumulative input tokens ↓
HiDe	On	On	On	**66.93**	5,553
no hindsight	Off	On	On	64.18	6,459
no control	On	Off	On	63.56	10,858
no g_t	On	On	Off	58.09	**4,827**

Table 4: **Mean deleted blocks per trajectory.** Think and observation blocks are reported separately. Configuration labels follow the supplied table; training stages for rows without an explicit stage label remain to be specified. Counts do not measure deletion timing, block length, or cumulative token savings, since blocks differ in length and the number of later invocations.

Environment / configuration	Think blocks / trajectory	Obs blocks / trajectory
WebShop	3.345	3.600
SearchQA	2.010	3.670
BabyAI RL	4.700	4.470
TextCraft	5.950	5.420
SciWorld	6.610	3.940

Table 5: **Information support for deleted blocks and retained controls.** Support rates are percentages; differences and their 95% confidence intervals are in percentage points. Reported differences are retained separately from the rounded support rates. Panels describe different information views and do not estimate the causal effect of new observations.

(a) At the deletion decision

Configuration	Deleted	Retained control	Difference [95% CI]
WebShop RL	93.09	68.64	+24.44 [12.04, 39.84]
WebShop SFT	88.12	83.67	+4.45 [−0.94, 10.00]
SearchQA RL	86.97	75.32	+11.65 [8.58, 14.61]
SearchQA SFT	82.43	82.06	+0.37 [−2.48, 8.93]
BabyAI RL [b]	84.29	76.24	+8.05 [−18.26, 21.35]
BabyAI SFT	80.67	88.91	−8.24 [−24.57, 12.38]
TextCraft RL	88.76	69.84	+18.92 [−2.31, 22.67]
TextCraft SFT	82.53	76.14	+6.39 [0.60, 12.24]
SciWorld RL [a]	97.22	69.31	+27.92 [23.49, 32.38]
SciWorld SFT	92.36	91.87	+0.48 [−1.75, 2.93]

(b) After the next environment observation

Configuration	Deleted	Retained control	Difference [95% CI]
WebShop RL	92.77	54.31	+38.46 [28.31, 46.72]
WebShop SFT	84.85	80.91	+3.94 [−1.40, 9.42]
SearchQA RL	88.00	64.96	+23.04 [19.62, 26.29]
SearchQA SFT	83.58	79.41	+4.17 [−2.63, 10.84]
BabyAI RL	86.50	72.09	+14.40 [−13.14, 26.33]
BabyAI SFT	81.36	79.08	+2.28 [−12.47, 15.86]
TextCraft RL	85.47	75.63	+9.84 [−3.18, 21.56]
TextCraft SFT	76.82	79.10	−2.28 [−8.55, 3.97]
SciWorld RL [a]	93.06	66.28	+26.77 [21.98, 31.96]
SciWorld SFT	89.08	88.15	+0.93 [−1.91, 4.11]

[a] The same SciWorld RL values were previously attributed to a mismatched replay. They are reproduced pending provenance confirmation and are excluded from the main interpretation here. [b] The BabyAI RL decision-time contrast has changed from an earlier table, while its interval is unchanged; confirmation of the interval is pending.