

MemHRL: Aligning Hierarchical Memory and Policy Learning for Long-Horizon LLM Agents

Xin Yin

xinyin1990@gmail.com

New Jersey Institute of Technology

Chenyi Huang

Johns Hopkins University

Ruiqi Liu

Rice University

Yue Ning

University of Toronto

Research Article

Keywords: Large language model agents, hierarchical reinforcement learning, long-term memory, long-horizon planning

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11435233/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

MemHRL: Aligning Hierarchical Memory and Policy Learning for Long-Horizon LLM Agents

Xin Yin^{1*}, Chenyi Huang², Ruiqi Liu³, Yue Ning⁴

¹ New Jersey Institute of Technology, Newark, 07102, USA

² Johns Hopkins University, Baltimore, 21218, USA

³ Rice University, Houston, 77005, USA

⁴ University of Toronto, Toronto, M5S 1A1, Canada

*xinyin1990@gmail.com, chuan128@jh.edu, ruiqiliu@alumni.rice.edu, lydianing59847@gmail.com

Abstract—Long interactive tasks require agents to retain useful experience while selecting actions at different temporal scales. MemHRL explicitly aligns action records and option summaries with low- and high-level policies through level-aware retrieval and failure-aware consolidation. A frozen language model scores candidates, while two small actor-critic selectors learn from terminal task credit, local progress, and interaction costs. We evaluate Qwen2.5-1.5B-Instruct on fixed ScienceWorld and WebShop subsets, using 100 test tasks per environment and three learning seeds. Comparisons include constrained ReAct-style, Reflexion-style, hierarchical-only agents, and five ablations. MemHRL obtains success rates of 0.0% and 12.0%, with normalized scores of 14.5% and 33.0%, respectively. Its success-rate differences from hierarchical-only are +0.0 and -1.0 percentage points. Removing long-term memory raises WebShop success to 15.7%, although no paired success-rate comparison remains significant after multiplicity correction. The results characterize the limits of this compact policy-memory alignment mechanism under constrained proposals; they do not establish performance on the full benchmarks. Raw trajectories, trained selectors, data fingerprints, and analysis scripts support independent verification.

Keywords—Large language model agents, hierarchical reinforcement learning, long-term memory, long-horizon planning

I. INTRODUCTION

An interactive language agent must preserve useful information while deciding which earlier experiences matter for its next action. A fact needed for manipulating an object may be too detailed for selecting a task stage, whereas a summary of a failed stage may omit the local feedback needed to recover. Long interaction sequences therefore couple forgetting, error accumulation, and inconsistent decisions across stages. Enlarging the context window alone does not specify the abstraction level at which an experience should influence a policy.

ReAct interleaves reasoning and actions [1], and Reflexion turns environmental feedback into verbal experience [2]. Temporal abstraction supplies a complementary mechanism: a high-level policy selects an option, and a low-level policy executes actions over its duration [3]. These mechanisms can coexist without ensuring that the high-level policy receives stage-level experience or that the low-level policy retains the details needed for execution. The resulting question is operational: does explicitly matching memory granularity to decision timescale improve a constrained language agent when other components are held fixed?

This question also requires a narrow account of novelty. GLIDER already combines hierarchical language-agent policies and reinforcement learning [4]. AgeMem learns unified management of short- and long-term memory [5], while HORMA separates hierarchical memory construction from learned navigation [6]. MemHRL studies a specific alignment rule within this broader design space: action records and failure-aware option summaries share a bounded library, and a fixed level preference routes them to separate trainable selectors. We do not claim that combining memory and hierarchy is new, or that the routing rule itself is learned.

Our contributions are an executable alignment mechanism with distinct write and retrieval granularities; a lightweight two-timescale policy update that retains terminal task credit while providing local progress to the action selector; and a controlled empirical evaluation with memory, consolidation, reward, and learning ablations. The language model remains frozen, making the intervention feasible on a single laptop GPU. The evaluation is deliberately bounded to text interaction and fixed task subsets. Success, partial progress, and interaction counts are analyzed separately so that an agent that fails earlier cannot be mistaken for a more efficient solver. Figure 1 shows the alignment mechanism evaluated under this limited-capacity setting.

II. RELATED WORK

A. Memory and Feedback in Language Agents

A-Mem dynamically organizes and evolves linked memory notes [7]. MemoryBank retrieves and updates long-term conversational memories [8], and Generative Agents use stored observations,

reflection, and planning to support coherent behavior [9]. These systems motivate retaining experience beyond a short context, but their settings and optimization targets differ from an interactive task policy. Reflexion [2] uses linguistic feedback across trials without conventional weight updates. Our single-trial reflection baseline adapts this idea to failed subgoals; it is not a reproduction of the original multi-trial protocol.

AgeMem [5] learns memory operations through staged reinforcement learning. HORMA [6] constructs a file hierarchy and trains evidence-guided navigation. ExpeL [10] extracts insights from task experience and recalls them with past trajectories without updating model parameters. UI-Mem [11] already combines hierarchical workflows, subtask skills, and failure knowledge with online GUI reinforcement learning. Our implementation fixes a lexical routing rule and learns two residual candidate selectors. The contribution is therefore a compact, controlled intervention on policy-level memory routing, not the general combination of experience memory and reinforcement learning.

B. Hierarchy and Interactive Evaluation

The options framework formalizes actions that extend over multiple time steps [3]. GLIDER [4] applies offline hierarchical reinforcement learning to language-based decision making, including separate rewards and efficient adaptation of a frozen backbone. MemHRL uses on-policy simulator trajectories and small selectors without demonstration-based initialization. Consequently, GLIDER’s published scores are not interchangeable with results under our model and action interface.

Interactive benchmarks expose different forms of grounding: ALFWorld [12] aligns textual and embodied tasks, ScienceWorld [13] tests scientific interaction, and WebShop [14] evaluates goal-directed shopping. We run the latter two with their native transition and scoring engines. The selected tasks and shopping catalog permit local experimentation, but do not constitute a full benchmark evaluation or evidence about general web autonomy.

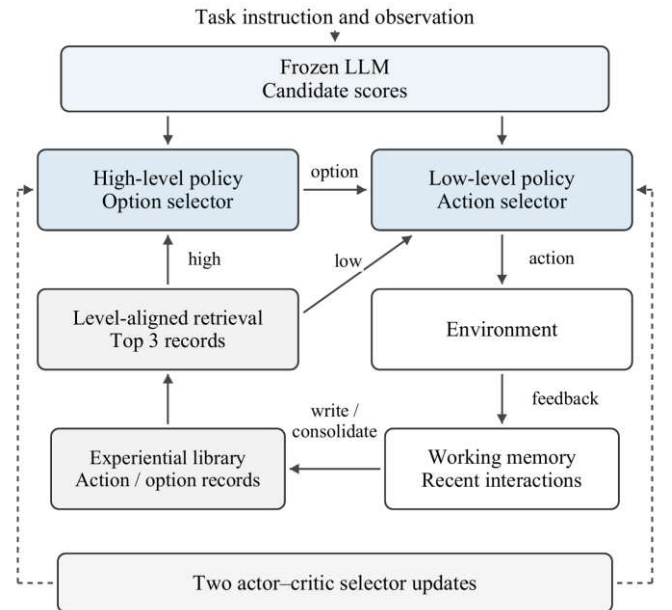


Fig. 1. Policy-level memory routing. Solid arrows show data flow; dashed arrows show selector updates. The language model and lexical retrieval remain fixed.

III. METHOD

A. Hierarchical Agent and Candidate Selection

We model interaction as a partially observed decision process with task instruction z , observation o , recent working memory W , and experiential library E . At an option boundary, the high-level policy conditions on the task, observation, working memory, and high-level retrieval and chooses one of five intents: locate, inspect, acquire,

transform, or complete. The low-level policy conditions additionally on the current intent and low-level retrieval. Figure 1 summarizes the information flow. These intents are fixed interpretable labels, not generated plans with independently verified completion predicates.

An option ends after positive score progress, two recent ineffective interactions, four actions, or episode termination. Replanning therefore uses observed progress and stagnation rather than an oracle subgoal detector. A shared filter ranks admissible actions using only visible task and observation text. It retains at most 16 candidates with navigation and manipulation coverage. Frozen Qwen2.5-1.5B-Instruct [15] scores candidate letters. The low-level proposal retains at most four actions; ScienceWorld additionally diversifies primitive command families. All methods share this grounding interface.

For level 1, the selector adds a trainable residual to the frozen candidate score:

$$\pi_l(c | x_t) = \text{softmax}_c[b_l(c)/2 + f_{\theta_l}(\varphi_c)] \quad (1)$$

Here b is the candidate logit relative to the highest logit, clipped to $[-8, 0]$. The 14-dimensional feature vector includes this prior, lexical task and observation similarity, retrieved positive and negative utility, recent repetitions, option–action compatibility, action type, and length. Each level has separate actor and critic networks with one 32-unit tanh layer. The two actors and two critics together contain 2,052 trainable parameters. Training samples from the resulting categorical policy; evaluation uses its argmax.

B. Level-Aligned Memory and Consolidation

Working memory holds six recent interaction records. Each action also writes a low-level record containing context, current option, action, observed outcome, utility, source episode, and timestamp. Low-record utility is score increment minus 0.05 per ineffective interaction. At an option boundary, consolidation deduplicates the action trace, records total progress and failure flags, and writes a high-level summary. Its utility is clipped to $[-1, 1]$ after subtracting 0.05 per ineffective interaction from score gain. The working buffer is then cleared. A shared recent-history text budget remains available to every method.

The experiential library retains the latest 256 action and option records. Retrieval ranks records by

$$S_l(m, q) = e(q)^T e(m) + \frac{0.05}{1 + \text{age}(m)} \quad (2)$$

$$+ 0.1u(m) + 0.6 \mathbb{1}[L(m) = l]$$

where e is a normalized 256-dimensional BLAKE2b-hashed token-count vector, age is the nonnegative timestamp difference, u is recorded utility, and L is the record's level. The query combines task, current observation, and option; the indexed record text combines context and option. The top three records are routed to the requested policy and also inform selector features. The level term is a soft preference, so cross-level retrieval remains possible. This is deterministic lexical similarity, not a pretrained semantic embedding or a learned retrieval gate.

During training, records are merged after each eight-episode collection batch. During evaluation, every episode starts from an independent copy of the training library. New evaluation records are available only within that episode. No held-out trajectory, gold solution, or hidden shopping goal enters the policy or the next episode's memory.

C. Rewards and Policy Optimization

In (3), R is benchmark score before clipping (ScienceWorld divided by 100; WebShop unchanged), ΔR its increment, I flags ineffective interactions, and T marks episode termination, including the step limit. The low-level reward is

$$r_t^L = T_t R_t + 0.5 \max(\Delta R_t, 0) \quad (3)$$

$$- 0.02 I_t - 0.002$$

The high-level reward is the duration-discounted sum of terminal raw score and interaction penalties within its option; it excludes the local positive-progress term. The high-level return discounts the next option by γ raised to the option duration. Fatal ScienceWorld scores remain negative during training; clipping to zero is used only for reported normalized scores.

Both levels optimize the clipped policy-gradient surrogate with clipping 0.2, Monte Carlo returns minus behavior-policy values, value-loss weight 0.5, and entropy weight 0.01. There is no generalized advantage estimator. Adam uses learning rate 0.003, $\gamma = 0.97$, four passes per episode, and gradient-norm limit 1. Eight trajectories are collected under fixed parameters, then updated sequentially with their recorded behavior probabilities. The language model and lexical encoder receive no gradient. This joint training changes policy selectors; it does not differentiate through memory writes or retrieval.

IV. EXPERIMENTS

A. Setup and Metrics

ScienceWorld 1.2.3 [13] supplies eight task families: finding plants, animals, living and non-living things, boiling, melting, freezing, and thermometer use. We select balanced subsets of 96 training, 24 development, and 100 test variations from the environment's corresponding official splits. Only the openDoors simplification is enabled, with a 48-interaction limit. ALFWorld was not included because its required planning dependency was unavailable in the tested Windows configuration.

WebShop [14] uses upstream commit 64fa2a5, native text observations, Lucene search, and unchanged purchase rewards. The first 50,000 entries in the original shuffled catalog yield 49,999 indexed products and 537 human goals. Because the official download required login, two public mirrors were independently checked: catalog prefixes matched after canonical JSON serialization, and attributes and human instructions matched byte-for-byte. We use custom product-disjoint subsets of 96/24/100 train/dev/test goals and a 24-interaction limit. These are not the official full-catalog leaderboard split. Mirror revisions, hashes, prices' initialization seed, and task identifiers are provided in the package.

All conditions use the same frozen Qwen model in float16, a 768-token prompt limit, candidate-logit temperature 2, and identical action interfaces and step limits. ReAct-style generates a concise next-step rationale; Reflexion-style additionally critiques unproductive feedback and keeps up to three reflections. Both use greedy generation capped at 24 tokens and choose a fresh intent at each step. Hierarchical-only retains persistent intents and recent context, without long-term memory or trained selectors. These constrained baseline adaptations permit a common action interface, but do not reproduce the original papers' prompts or multi-trial budgets.

The full method and four training ablations each run 96 training episodes for seeds 11, 23, and 47. Flat memory removes the level-match term; no consolidation omits option summaries; no progress reward removes the low-level shaping term; no long-term memory removes experiential retrieval and writes. The no-RL ablation greedily uses frozen-model scores with the full method's training library. Thus, it isolates selection-time learning while retaining memory produced during training. Deterministic baselines run once per task; learned conditions and no-RL run all three seeds. Model choice and implementation debugging use development tasks; the frozen test tasks are not used for tuning.

Success requires score 1. Reported score is 100 times nonnegative benchmark score. Mean steps cover all episodes and are not interpreted as efficiency gains in isolation. IIR divides ineffective interactions by all interactions: ScienceWorld compares visible room/inventory descriptions and score, while WebShop compares observation text and reward. This observable proxy can flag useful information-gathering actions and is not a complete simulator-state test. Segment progress ratio (SPR) counts option segments with positive net score gain; it is a progress proxy, not planning accuracy or verified subgoal completion.

Seeds are averaged within task. We compute 95% percentile intervals with 4,000 cluster-bootstrap resamples: task variations in ScienceWorld and product groups in WebShop, whose 100 goals cover 90 products. Task-weighted means are retained when product groups differ in size. Paired success differences use 10,000 cluster-level sign permutations and Holm correction across both benchmarks and all full-method comparisons. This preserves shared-product dependence. A zero-width interval under zero observed successes does not imply zero population uncertainty. Runs use one RTX 5060 Laptop GPU with 8 GB memory, an i7-14650HX CPU, PyTorch 2.8.0, and Transformers 4.56.2. Raw trajectories, checkpoints, update norms, configs, and analysis scripts support recomputation. One incomplete test condition was restarted after a process interruption; its partial records were excluded. Exact trajectory replay is not guaranteed.

B. Main Results

Table I summarizes the fixed test subsets; Fig. 2 includes partial-progress scores for all conditions. The paired success-rate differences from hierarchical-only are ScienceWorld: +0.0 percentage points, 95% CI [0.0, 0.0], adjusted $p = 1.000$; WebShop: -1.0 percentage points, 95% CI [-7.1, 4.8], adjusted $p = 1.000$. These comparisons concern the selected tasks and three trained seeds, not a broader model population. Section V discusses the performance bottlenecks, memory diagnostics, and limits of generalization.

MemHRL uses 41.0 and 11.1 interactions per episode, compared with 8.0 and 13.1 for hierarchical-only. Early terminal failures and budget exhaustion make a shorter trajectory ambiguous. We therefore interpret interaction counts jointly with success and score, and provide successful-task counts and logical token demand in the package. SPR records observed score-bearing segments; different replanning frequencies prevent interpreting it as a common ground-truth planning-accuracy measure.

TABLE I. MAIN RESULTS ON FIXED TEST SUBSETS

Method	SR (%)	SPR (%)	Steps	IIR (%)
ScienceWorld				
ReAct-style	0.0 [0.0,0.0]	5.4	7.2	56.5
Reflexion-style	0.0 [0.0,0.0]	5.3	7.6	54.4
Hierarchical	0.0 [0.0,0.0]	9.6	8.0	63.1
MemHRL	0.0 [0.0,0.0]	9.8	41.0	37.3
WebShop				
ReAct-style	6.0 [1.9,11.1]	3.6	14.3	41.5
Reflexion-style	7.0 [2.7,12.4]	3.9	13.6	40.1
Hierarchical	13.0 [6.9,20.2]	11.3	13.1	48.9
MemHRL	12.0 [6.4,18.6]	15.5	11.1	43.7

SR shows mean and 95% cluster-bootstrap interval. SPR and IIR are percentages; steps include failures. Each condition uses 100 tasks per benchmark. Normalized scores appear in Fig. 2.

C. Ablation and Horizon Analysis

Table II isolates the specified memory, consolidation, reward, and selection changes. The paired effect of the level preference relative to flat retrieval is ScienceWorld +0.0 points (adjusted $p = 1.000$); WebShop +1.3 points (adjusted $p = 1.000$). The no-RL condition retains a training-derived memory library, so its comparison concerns learned selection rather than the entire experience-collection process. Removing long-term memory yields 15.7% shopping success versus 12.0% for MemHRL. The paired MemHRL-minus-no-memory difference is -3.7 percentage points (95% CI [-8.9, 1.2]; unadjusted $p = 0.202$; Holm-adjusted $p = 1.000$). The ablations therefore do not establish a consistent success benefit from the proposed memory mechanism. Because each ablation is trained separately, this contrast includes changes in the learned policy and collected experience; it does not isolate the immediate effect of a retrieved record.

ScienceWorld’s long-duration bucket contains 26 tasks (ReAct duration >2.0 steps), with MemHRL/ReAct success 0.0%/0.0%; WebShop has no nonempty long-duration tertile after retaining ties. Early failure confounds this duration proxy, so these strata do not establish robustness with respect to optimal solution length.

The most frequent recorded failure categories are ScienceWorld: budget with ineffective interactions, 228 of 300 failed episodes; WebShop: unsuccessful purchase, 184 of 264 failed episodes. These descriptive counts pool the three seeds and do not identify causal mechanisms.

One recorded failure is boil:21, seed 23: the final action “focus on fountain” produces raw score -1 despite 3 retrieved records. This illustrates that available experience does not ensure correct action grounding. The complete failure taxonomy and trajectories are included for inspection.

TABLE II. ABLATION SUCCESS RATES WITH 95% INTERVALS

Condition	ScienceWorld	WebShop
MemHRL	0.0 [0.0,0.0]	12.0 [6.4,18.6]
Flat memory	0.0 [0.0,0.0]	10.7 [5.7,16.7]
w/o consolidation	0.0 [0.0,0.0]	9.7 [5.3,15.1]
w/o progress reward	0.0 [0.0,0.0]	12.0 [6.2,18.5]
w/o long-term memory	0.0 [0.0,0.0]	15.7 [9.5,22.6]
w/o RL selection	0.3 [0.0,1.0]	11.0 [5.8,17.0]

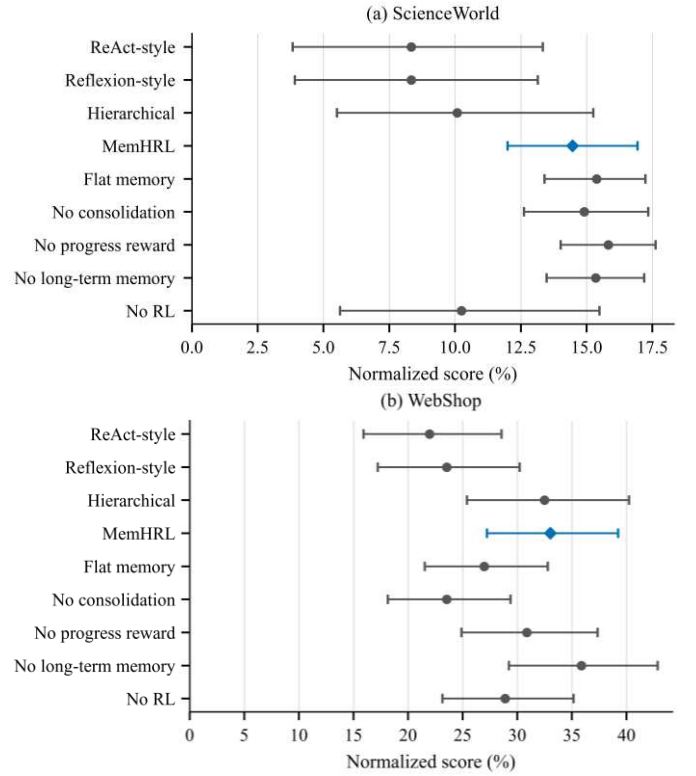


Fig. 2. Mean normalized scores and 95% cluster-bootstrap intervals on (a) ScienceWorld and (b) WebShop. Three seed outcomes are averaged within each task for learned conditions and no-RL; deterministic baselines run once per task.

V. DISCUSSION

A. Interpreting the Limited Performance

ScienceWorld exposes a gap between partial progress and completion: MemHRL scores 14.5% but completes none of 300 episodes. Its failures comprise 228 budget-limited episodes with ineffective actions and 72 fatal-focus failures, compared with 85 fatal-focus failures in 100 hierarchical-only episodes. This distribution is consistent with avoiding some early failures without completing the remaining procedure. Longer trajectories and lower IIR therefore do not demonstrate successful planning. A selector cannot recover a required action excluded from the four-action shortlist. Sparse completion feedback and 96 training episodes per seed may further limit procedure learning; these contributions are not individually isolated.

On WebShop, MemHRL slightly underperforms hierarchical-only in success (12.0% versus 13.0%), despite similar scores (33.0% versus 32.5%). Its 184 unsuccessful purchases among 264 failures indicate that purchase actions often fail to satisfy all constraints. Generic intents may organize navigation without resolving attributes, options, and price matching. Experience from another product could also suggest an inappropriate action. These are plausible explanations rather than demonstrated causes. The paired success interval includes zero, and the comparison changes both memory and selector training, preventing isolation of a memory effect.

B. Memory Effectiveness and Possible Interference

Removing long-term memory yields 47 successes in 300 WebShop episodes, compared with 36 for MemHRL. Its higher score (35.9% versus 33.0%) accompanies longer trajectories (12.3 versus 11.1 steps), so not every interaction metric improves. The uncertain success difference in Section IV-C nevertheless motivates investigating interference. In Eq. (2), the 0.6 level bonus exceeds the maximum 0.05 recency or 0.1 positive-utility contribution. Level matching cannot guarantee product or state relevance. Mandatory top-three retrieval lacks a relevance threshold or abstention, and weak recency weighting does not invalidate experience when task constraints or policies change.

We additionally audit the existing WebShop MemHRL logs without new environment runs. Across 300 episodes and 4,555 high- and low-level decisions, the 13,665 logged retrieval slots contain 3,220 positive-utility records (23.6%), 4,994 zero-utility records (36.5%), and 5,451 negative-utility records (39.9%). We resolve logged record identifiers against each seed’s training checkpoint and reconstruct episode-local writes and consolidation using the recorded trajectories. Each occurrence is counted, including repeated retrievals. All retrieved records match the requested policy level, yet 76.4% have non-positive utility; 33.5% of slots refer to training-library records and the remainder

to episode-local records. These are pooled descriptive counts, not independent observations for significance testing. Figure 3 summarizes the pooled retrieval-utility distribution.

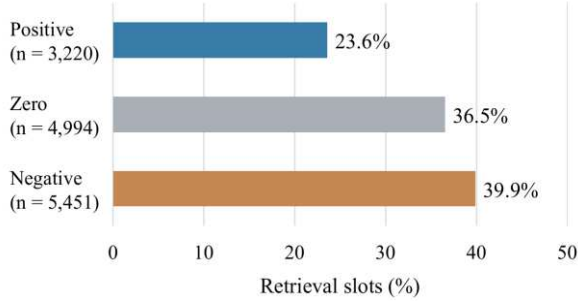


Fig. 3. Utility distribution of 13,665 retrieval slots across 300 WebShop episodes. Repeated retrievals are counted. Utility sign alone does not establish relevance or harm.

Non-positive utility does not label memory irrelevant or harmful: search and inspection may receive zero immediate reward, and negative records can warn against failure. The audit shows that level alignment does not predominantly select reward-bearing experience. Lexical overlap can retrieve another product or an incompatible earlier state. Retrieved text enters the model prompt and utility-based selector features, creating possible interference. Prompt text omits numeric utility, source, and age, which may hinder applicability judgments; failure labels and signed features may conversely help. The logs establish neither the dominant route nor whether older records cause reduced success.

A causal follow-up should hold a trained selector and starting library fixed and compare complete rollouts with retrieval enabled, disabled, or filtered by current constraints and age. Separate retraining would test adaptation to each memory policy. Product and state compatibility audits, top-k variation, relevance thresholds with abstention, and stronger recency decay or learned relevance gates could distinguish useful warnings from misleading experience. Age requires analysis conditional on relevance, since old transferable procedures may remain useful. These are proposed interventions, not new results.

C. Conditions Under Which Alignment May Help

Alignment may help when tasks revisit transferable procedures, summaries preserve prerequisites and recovery rules, and action records remain compatible with the current state. Adequate proposal coverage and selectors able to use evidence are also required. Routing procedural summaries to high-level policies and local feedback to action policies could then reduce interference between abstractions. Product-specific choices with changing attributes may instead favor current observations. A relevance gate with abstention is therefore a natural extension.

These conditions remain hypotheses. MemHRL numerically exceeds flat memory in WebShop success (12.0% versus 10.7%) and score (33.0% versus 27.0%), but the success difference is unresolved. Removing memory performs better numerically, and ScienceWorld completion remains absent. Neither broad alignment benefits nor greater benefits on longer tasks are established. Comparing recurring versus novel procedures and reliable versus mismatched retrieval under matched backbones and training budgets would test these conditions.

D. Experimental Limitations and Generalizability

The single frozen 1.5B backbone, 768-token prompt budget, and constrained shortlist restrict inference about larger agents. Grounding or proposal limits may mask memory benefits, while memory text competes with current context. Stronger models or broader proposals could improve completion, but stronger models might also reduce the incremental value of external memory by using recent context better. Neither outcome is established. Model-scale comparisons should match task splits, retrieval settings, and interaction budgets while reporting completion and inference cost.

Only 96 training episodes per condition and seed supply the selectors and library. Three seeds characterize some variability but establish neither convergence nor stability under sparse terminal rewards and short collection batches. Longer training could improve

selection and experience coverage, or reinforce partial-progress strategies and misleading memories. No training-budget sweep was performed. Development-set learning curves across larger budgets, followed by fixed test evaluation with multiple seeds, are needed before attributing low success to undertraining or predicting improvement.

External validity is limited to eight ScienceWorld families with openDoors and a 48-step cap, and 100 WebShop goals from a restricted catalog and custom product-disjoint split with a 24-step cap. These settings affect coverage, search difficulty, and opportunities for completion. The 300 learned-condition episodes revisit 100 tasks, with only 90 WebShop product clusters. Cluster-aware inference cannot expand coverage, and zero ScienceWorld successes does not imply zero population success probability. Broader official splits, additional families, and separately controlled horizon and training budgets are needed to assess transfer.

VI. CONCLUSION

MemHRL makes policy-memory alignment explicit through level-tagged experience, boundary-triggered consolidation, and two small learned selectors. No full-method success-rate comparison survives the specified multiplicity correction. ScienceWorld partial progress does not translate into completion, and removing long-term memory improves WebShop success numerically. The retrieval audit shows that matching memory level does not ensure positive utility; it does not establish that non-positive or older records are harmful. Alignment may help when transferable experience is relevant to the current state and the proposer supplies adequate actions, but these conditions remain to be tested. The small frozen backbone, limited training budget, and selected benchmark subsets restrict the conclusions. Stronger models, longer training, and relevance-aware retrieval are concrete evaluation directions, rather than demonstrated remedies. The released trajectories and ablations provide a basis for testing them while keeping success, partial progress, and interaction cost separate.

REFERENCES

- [1] S. Yao et al., "ReAct: Synergizing reasoning and acting in language models," in Proc. ICLR, 2023.
- [2] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in Advances in Neural Information Processing Systems, vol. 36, 2023.
- [3] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," Artificial Intelligence, vol. 112, nos. 1–2, pp. 181–211, 1999, doi: 10.1016/S0004-3702(99)00052-1.
- [4] Z. Hu et al., "Divide and conquer: Grounding LLMs as efficient decision-making agents via offline hierarchical reinforcement learning," in Proc. ICML, PMLR, vol. 267, pp. 24570–24590, 2025.
- [5] Y. Yu et al., "Agentic memory: Learning unified long-term and short-term memory management for large language model agents," in Proc. ACL, pp. 21457–21483, 2026, doi: 10.18653/v1/2026.acl-long.981.
- [6] H.-L. Hsu, N. L. Kuang, B. Liu, Z. Yao, and Y. He, "Organize then retrieve: Hierarchical memory navigation for efficient agents," arXiv:2606.11680, 2026.
- [7] W. Xu et al., "A-Mem: Agentic memory for LLM agents," in Advances in Neural Information Processing Systems, vol. 38, 2025, doi: 10.52202/085713-0593.
- [8] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "MemoryBank: Enhancing large language models with long-term memory," in Proc. AAAI, vol. 38, no. 17, pp. 19724–19731, 2024, doi: 10.1609/aaai.v38i17.29946.
- [9] J. S. Park et al., "Generative agents: Interactive simulacra of human behavior," in Proc. ACM UIST, art. 2, pp. 1–22, 2023, doi: 10.1145/3586183.3606763.
- [10] A. Zhao, D. Huang, Q. Xu, M. Lin, Y.-J. Liu, and G. Huang, "ExpeL: LLM agents are experiential learners," in Proc. AAAI, vol. 38, no. 17, pp. 19632–19642, 2024, doi: 10.1609/aaai.v38i17.29936.
- [11] H. Xiao et al., "UI-Mem: Self-evolving experience memory for online reinforcement learning in mobile GUI agents," arXiv:2602.05832, 2026.
- [12] M. Shridhar et al., "ALFWorld: Aligning text and embodied environments for interactive learning," in Proc. ICLR, 2021.
- [13] R. Wang, P. Jansen, M.-A. Côté, and P. Ammanabrolu, "ScienceWorld: Is your agent smarter than a 5th grader?" in Proc. EMNLP, pp. 11279–11298, 2022, doi: 10.18653/v1/2022.emnlp-main.775.
- [14] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "WebShop: Towards scalable real-world web interaction with grounded language agents," in Advances in Neural Information Processing Systems, vol. 35, 2022.
- [15] Qwen Team, "Qwen2.5 technical report," arXiv:2412.15115, 2024.