

MemOPD: On-Policy Distillation through Memory State Alignment for Long-Horizon Agents

Zhiyuan Liu^{1*}, Tinghong Ye^{2*}, Chenghao Liu^{1*}, Yizhuo Li³, Songfang Huang^{1†}

¹School of Advanced Manufacturing and Robotics, Peking University

²Zhejiang University ³Harbin Institute of Technology

liuzhiyuan26@stu.pku.edu.cn, yetinghong@zju.edu.cn, chliu@stu.pku.edu.cn, yizhuoli815@gmail.com, hsf@pku.edu.cn

Abstract

Long-horizon agents accumulate growing contexts during interaction, impairing performance and stability. Compact memory mitigates this problem by compressing and rewriting the history retained between model invocations. Learning what to retain typically relies on proximal policy optimization (PPO) with final task rewards, but sparse rewards provide little guidance for individual memory updates. This limitation motivates on-policy distillation (OPD), which supplies dense teacher supervision on student rollouts. For such supervision to be valid, the teacher must evaluate each sampled action under the same state in which it was generated. However, the context rewriting performed during memory compression can break this alignment. When sampled responses are retained and re-encoded for later invocations, flattening the interaction into a persistent history may cause the teacher to score the action under a state that the student never visited during rollout. The action therefore remains on-policy by provenance, but not necessarily by state. We therefore propose Memory-Aligned On-Policy Distillation (MemOPD). MemOPD records the inputs and sampled outputs of each model invocation, restores its original token positions and causal visibility, and packs the reconstructed invocations for efficient teacher scoring. The teacher provides full-vocabulary supervision at the sampled action positions, while PPO preserves the final task objective. Experiments verify state alignment across several context updates and show that it improves F1 by 7.0% over persistent-history teacher scoring in a matched control. Overall, MemOPD-3B improves F1 over PPO by up to 416.2%, while packing yields up to a 1.63 $\times$ speedup in actor computation during training. The code for this work is publicly available at: <https://github.com/TPssp/MemOPD>.

1 Introduction

Large language models (LLMs) have driven rapid advances in language agents, enabling them to reason, plan, and interact with external tools (Wang et al. 2024; Yao et al. 2023; Nakano et al. 2022). These agents have demonstrated strong performance in a wide range of long-horizon tasks such as information-seeking workflows, where multiple rounds of reasoning, action execution, and environment feedback are required (Yao et al. 2022; Deng et al. 2023; Gur et al. 2024; Jin et al. 2025; Zheng et al. 2025). During such interactions,

each model invocation produces new responses and observations that may influence future decisions. As a result, maintaining the complete interaction history leads to continuously growing contexts, increasing Transformer computation costs and making it more difficult for the model to identify and utilize relevant information (Vaswani et al. 2017; Liu et al. 2024; Wu et al. 2025; An et al. 2025).

To address this challenge, existing approaches introduce retrieval mechanisms, external memory systems, or context compression strategies to control the amount of information provided to the model (Lewis et al. 2020; Karpukhin et al. 2020; Borgeaud et al. 2022; Park et al. 2023; Zhong et al. 2023; Packer et al. 2024; Chhikara et al. 2025; Xu et al. 2025; Yoon et al. 2024; Li et al. 2023; Cao et al. 2024; Lee et al. 2024; Jiang et al. 2023, 2024). Among them, compact memory provides a more direct solution by learning to rewrite the context retained between model invocations, selectively keeping useful information and removing outdated content. For example, MEM1 (Zhou et al. 2025) learns a compact internal state that jointly supports memory consolidation and reasoning, allowing the agent to operate with a bounded context while solving long-horizon tasks.

A common strategy for training such agents is to optimize the entire interaction trajectory with proximal policy optimization (PPO) using the final task reward (Schulman et al. 2017; Jin et al. 2025; Zhou et al. 2025). However, the reward is often sparse and delayed, providing limited supervision for individual memory updates.

Teacher distillation provides an alternative source of dense learning signals by transferring the teacher’s output distribution to the student model (Hinton, Vinyals, and Dean 2015; Kim and Rush 2016). Recent language model studies further extend distillation to student-generated sequences (Gu et al. 2024; Ko et al. 2024; Agarwal et al. 2024). On-policy distillation (OPD) can therefore guide sampled memory decisions while PPO preserves the final task objective.

However, directly applying OPD in this setting introduces an additional requirement: the state used for teacher evaluation must exactly match the state in which the student generated the action. Standard OPD assumes that the autoregressive prefix that produces an action remains unchanged when the action is scored, but context rewriting can break this assumption. During interaction, the agent may generate

*These authors contributed equally.

†Corresponding author.

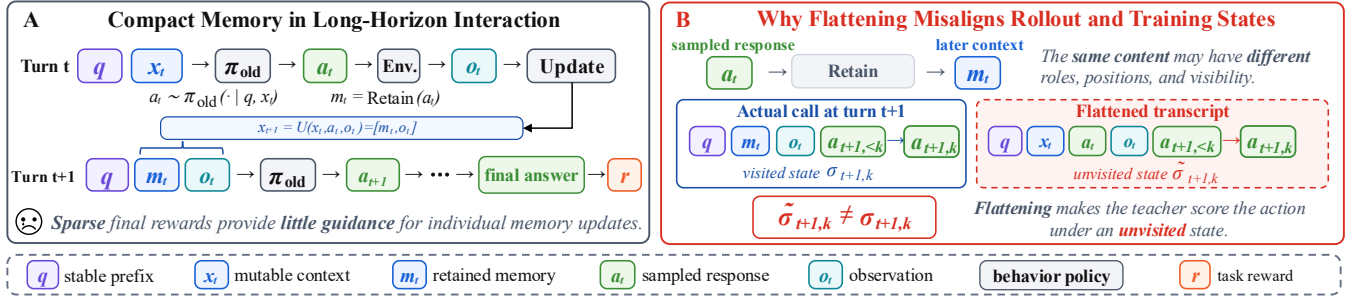


Figure 1: Motivation for MemOPD. (A) During compact memory interaction, the behavior policy retains selected content from each response and combines it with the next observation, thereby controlling context growth. PPO learns from the final task reward, which provides limited guidance for individual memory updates. (B) Directly flattening these calls changes the state used to score a sampled action. Content retained as later context can have different roles, positions, and visibility, causing the reconstructed training state $\tilde{\sigma}_{t+1, k}$ to differ from the rollout state $\sigma_{t+1, k}$. This mismatch motivates memory state alignment.

a response, retain only part of it as memory, and later insert the retained content into a new context. If training simply flattens these interactions into a persistent history, the same response tokens may have different positions, causal visibility, or prediction locations.

Consequently, an action can remain on-policy in terms of its origin, since it was sampled from the student policy, while the reconstructed training state no longer corresponds to any state visited during rollout. The teacher may therefore evaluate an action under an incorrect context, causing the distillation objective to optimize a different conditional distribution from the one that produced the behavior. In our 3B model audit, persistent-history reconstruction produced a p99 log probability error of 1.774, changed the top prediction at 651 sampled action positions, and falsely triggered PPO clipping for 13.29% of the actions.

This observation motivates Memory-Aligned On-Policy Distillation (MemOPD), which reconstructs the call state that produced each sampled action before applying teacher supervision. Instead of treating the interaction history as a single flattened sequence, MemOPD records the exact input and output of every model invocation during rollout, restores the original token positions, causal visibility, and prediction locations, and packs the reconstructed invocations into an efficient training representation. The teacher can then provide dense full vocabulary supervision under the state that produced each action, while PPO continues to optimize the final task objective.

To verify that the reconstruction preserves the original computation, we introduce rollout context equivalence (RCE), which requires the packed representation to produce the same action logits as independently executing each model invocation. RCE therefore tests whether packed training reproduces the conditional computation used during rollout rather than merely forming a valid tensor. Moreover, the reconstructed representation enables efficient batched execution by sharing common computation across invocations without sacrificing state fidelity.

We evaluate MemOPD on long-horizon retrieval tasks in which the agent uses compact memory. Across five independently trained seeds, MemOPD-3B raises F1 over PPO by up to 416.2%. In a matched Q2 control, it further improves F1

over persistent-history teacher scoring by 7.0%. Its packing strategy yields a maximum $1.63\times$ speedup in actor computation during training.

Our contributions are summarized as follows:

1. We identify and formulate the state mismatch problem introduced by context rewriting in compact memory agents. We show that student-generated actions alone are insufficient for valid on-policy distillation, and introduce memory state alignment as a necessary condition for applying OPD to long-horizon agents.
2. We propose MemOPD, a training framework that reconstructs each model invocation, separates sampled actions from their later context occurrences, and packs the reconstructed states for efficient teacher-guided optimization.
3. We introduce RCE as a verification criterion for memory-aware training and show that MemOPD exactly recovers computation across diverse context-update strategies. Experiments on long-horizon retrieval tasks demonstrate consistent gains in performance and efficiency.

2 Related Work

Memory management for long-horizon agents. Long-horizon agents combine reasoning with environment actions across web navigation, information seeking, and other multi-turn tasks (Yao et al. 2023, 2022; Nakano et al. 2022; Deng et al. 2023; Gur et al. 2024). To keep interaction histories bounded, agent systems maintain episodic or hierarchical memories outside the active context (Park et al. 2023; Zhong et al. 2023; Packer et al. 2024; Chhikara et al. 2025; Xu et al. 2025), while prompt compression and learned reading shorten the context presented to the model (Yoon et al. 2024; Lee et al. 2024; Jiang et al. 2023, 2024). Other approaches learn context management inside the agent through trajectory folding, explicit memory operations, bounded memory control, or memory specific training signals (Sun et al. 2025; Yu et al. 2026; Wang et al. 2026; Li et al. 2026b; Zhou et al. 2025; Li et al. 2026a). MEM1, for example, learns a consolidated internal state and observes that position changes prevent its packed representation from exactly recovering rollout computation (Zhou et al. 2025). These methods mainly study

what information an agent should retain, retrieve, or compress. They do not provide the state aligned OPD interface needed to score a sampled decision after its context has been rewritten. MemOPD addresses this training condition by recovering the exact state that produced each decision.

Learner states and on-policy distillation. Imitation learning has long recognized that learner decisions visit states that differ from a fixed expert trajectory. DAGger therefore queries an expert on states visited by the learner (Ross, Gordon, and Bagnell 2011). This strategy determines where supervision is collected, but it does not ensure that a stored trajectory later reproduces the same conditioning state. Knowledge distillation matches a student to a teacher distribution, while sequence level methods transfer supervision to generated sequences (Hinton, Vinyals, and Dean 2015; Gou et al. 2021; Kim and Rush 2016). Language model methods further use reverse Kullback–Leibler (KL) objectives or student generated data (Gu et al. 2024; Ko et al. 2024). Generalized knowledge distillation formulates OPD by querying a teacher on student generated sequences (Agarwal et al. 2024). Standard OPD assumes that the autoregressive prefix used to generate an action remains available when the action is scored. Context rewriting breaks this assumption because a response can later reappear as context with different positions or visibility. Consequently, applying standard OPD directly does not guarantee valid teacher supervision. MemOPD instead reconstructs the conditioning state and preserves the sampled action domain; its reverse KL term is unchanged.

3 Method

3.1 Overview

An agent with compact memory repeatedly invokes a trainable student policy, executes an external action, and rewrites the context for the next invocation. If training flattens these invocations into one persistent sequence, action likelihoods and teacher targets can be evaluated under inputs that the student never encountered. The resulting objective no longer represents the policy trajectory that produced the data, even though the batch remains syntactically valid.

MemOPD prevents this mismatch by recording each student invocation, reconstructing its memory state, and packing the independent computations. At each update, a frozen snapshot π_{old} of the trainable student π_{θ} generates the rollout, whereas the frozen teacher π_T only scores recorded actions after reconstruction. OPD then supplies local guidance on these actions, while PPO retains the task objective. The reference policy π_{ref} and critic V_{ϕ} use the same reconstruction, so every objective uses consistent state and action semantics. Figure 2 illustrates the pipeline.

3.2 Compact Memory Rollouts

To identify the state that training must reproduce, we first formalize how compact memory generates each sampled action. We write the fixed task and system prefix as q ; it specifies the task and the interaction protocol. Before invocation t , the mutable context x_t contains the memory and latest observation available to the student. Conditioned on $[q, x_t]$, the

behavior policy samples a response a_t that updates memory and selects a search or final-answer action. The environment executes this action and returns an observation o_t , a task reward r_t , and a terminal indicator d_t . The context update rule U then constructs the input for the next invocation:

$$a_t \sim \pi_{\text{old}}(\cdot \mid q, x_t), \quad (o_t, r_t, d_t) = \mathcal{E}(a_t), \quad (1)$$

$$x_{t+1} = U(x_t, a_t, o_t).$$

The student policy, the environment, and U jointly determine the states visited during rollout. The teacher does not generate a separate trajectory; it is queried only after these student states have been reconstructed.

Our end-to-end study instantiates U with the compact-memory protocol of MEM1 (Zhou et al. 2025). The stable prefix contains the task and interaction instructions. Each response contains reasoning, retained memory, and either a search command or a final answer. After a search, the returned evidence becomes the new observation. The runtime keeps the response content required for the next invocation, appends the new evidence, and removes the earlier mutable context. A three-invocation interaction therefore has the form

$$[q] \rightarrow a_0, \quad [q, m_0, o_0] \rightarrow a_1, \quad [q, m_1, o_1] \rightarrow a_2, \quad (2)$$

where m_t is the content retained from response a_t . The third invocation receives $[q, m_1, o_1]$, not an accumulated history such as $[q, a_0, o_0, a_1, o_1]$. This distinction is central to training because a sampled response and its later context copy can have different token positions and visibility.

Training first uses supervised fine tuning (SFT) to teach the required reasoning, memory, search, and answer format. The frozen behavior policy π_{old} then samples complete student interactions, and PPO optimizes the student using the final task reward. The reference policy limits drift, while the critic and generalized advantage estimation (GAE) provide token-level learning signals. Before any objective scores an action, MemOPD reconstructs the invocation that produced it. The frozen teacher then adds dense supervision without generating the trajectory or replacing the task reward.

3.3 Memory State Alignment

Rollout and training states. Valid training requires every sampled action to be scored under the invocation that produced it. For invocation t , let $z_t = [q, x_t]$ be the exact tokenized input. We define its realized memory state as

$$\sigma_t = (z_t, \rho_t, V_t, \gamma_t), \quad (3)$$

where ρ_t gives token positions, V_t specifies causal visibility, and $\gamma_t(k)$ identifies the position whose output distribution predicts action token $a_{t,k}$. Thus, a memory state is the complete model input presented at an invocation, including its token IDs and computation structure. It is neither the retained text alone nor a Transformer hidden state, and identical decoded text can represent different memory states.

The state used to predict action token $a_{t,k}$ also includes the earlier tokens $a_{t,<k}$ from the same response. Let $\sigma_{t,k} = (z_t, a_{t,<k}, \rho_t, V_t, \gamma_t(k))$ denote this rollout state and $\tilde{\sigma}_{t,k}$ the state reconstructed during training. Memory state alignment requires $\tilde{\sigma}_{t,k} = \sigma_{t,k}$ for every sampled action

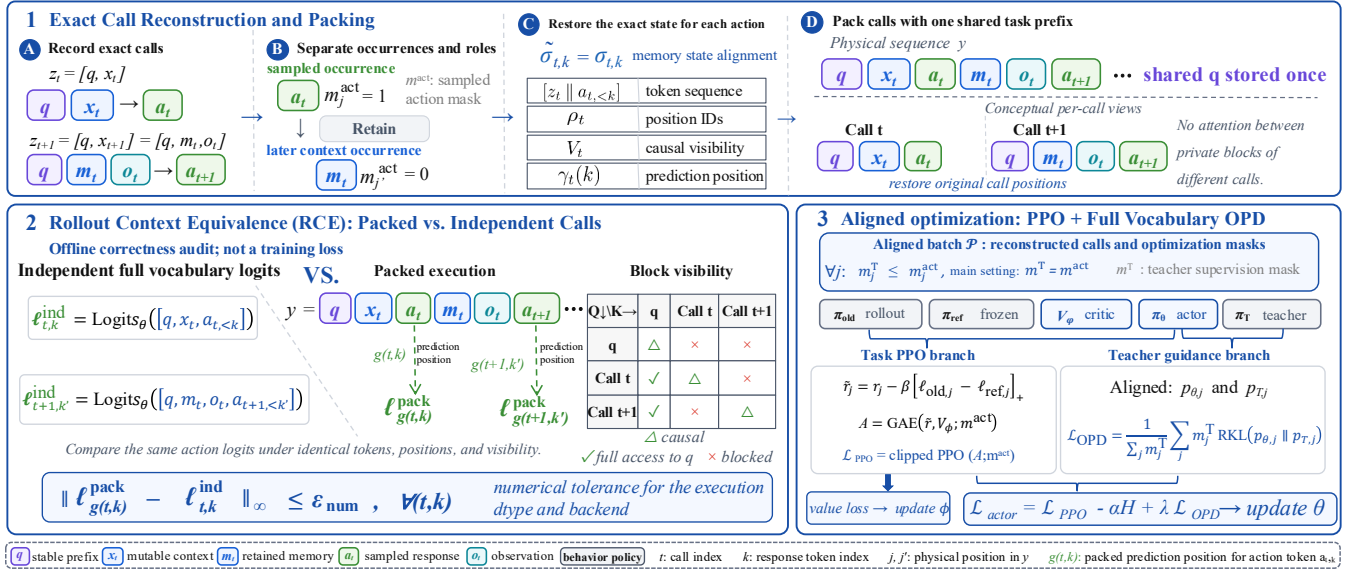


Figure 2: Overview of MemOPD. Here, a *call* is a model invocation. (1) MemOPD reconstructs each call, separates sampled actions from later context occurrences, and packs the aligned computations under a shared task prefix. (2) RCE compares packed and independent full vocabulary logits within numerical tolerance. (3) The aligned batch supports PPO over the sampled action domain and full vocabulary teacher guidance over the teacher supervision mask.

token. When this equality fails, behavior likelihoods, PPO ratios, and teacher targets no longer describe the sampled decision, so action provenance alone is insufficient.

Reconstruction and packing. To enforce this equality, MemOPD records z_t and the exact sampled token IDs of a_t during rollout rather than decoding and tokenizing the response again. Keeping these IDs preserves the sampled action and its token boundaries throughout reconstruction.

The reconstruction compiler places the recorded invocations in one physical sequence,

$$y = [q, x_0, a_0, x_1, a_1, \dots, x_{T-1}, a_{T-1}], \quad (4)$$

while treating physical adjacency and causal visibility separately. The stable prefix q has one physical occurrence, keeps positions $0, \dots, |q| - 1$, and is visible to every invocation block. This sharing is valid because its token IDs and positions are identical in all independent executions. A token in x_t sees only q and preceding tokens in the context of invocation t . A token in a_t additionally sees preceding tokens of a_t , while tokens assigned to other invocations are blocked. Finally, positions restart from the indices used by the corresponding independent invocation instead of increasing across the packed sequence.

When a response is retained for a later invocation, the compiler represents it twice because its two occurrences serve different purposes. The first is the exact sampled action in the policy trajectory. The second is context for a later invocation and therefore receives the positions and visibility of that invocation. If a sequence limit would remove a conditioning token or sampled action, the compiler rejects the example rather than silently changing the training state.

A correct reconstruction should reproduce, for every sampled action token, the same full vocabulary logits as indepen-

dent execution. We call this property rollout context equivalence (RCE). Let $g(t, k)$ denote the position in the packed sequence whose output distribution predicts action token $a_{t,k}$. Let $\ell_{g(t,k)}^{\text{pack}}$ and $\ell_{t,k}^{\text{ind}}$ be the corresponding full vocabulary logits from packed and independent execution. RCE requires

$$\|\ell_{g(t,k)}^{\text{pack}} - \ell_{t,k}^{\text{ind}}\|_{\infty} \leq \epsilon_{\text{num}} \quad \text{for every } (t, k), \quad (5)$$

where ϵ_{num} is the numerical tolerance for the execution dtype and backend. We test this condition directly against independent invocation logits for the same recorded actions.

Sampled action domain. Correct conditioning does not determine which physical tokens are policy decisions. Retained response copies and environment observations condition future actions, but they were not sampled at those physical positions. We mark these decisions as follows:

$$m_j^{\text{act}} = 1[y_j \text{ lies in the sampled occurrence of a recorded } a_t]. \quad (6)$$

The selected positions form the sampled action domain. Although a later context occurrence may repeat the same token IDs, it occupies different physical positions and receives $m^{\text{act}} = 0$. Policy loss, entropy, reference regularization, value learning, and GAE (Schulman et al. 2016) use this domain. GAE advances across ordered sampled actions and skips positions used only for context. Teacher supervision has an indicator m^T satisfying $m_j^T \leq m_j^{\text{act}}$. The teacher mask may select a subset of the action domain; in the main experiment, it covers every sampled response token, so $m^T = m^{\text{act}}$.

This sampled-action-domain contract is separate from state reconstruction. Correct visibility can still mark a retained response copy as another action, causing the actor,

Model	Q2				Q8				Q16			
	EM $\uparrow$	F1 $\uparrow$	Peak $\downarrow$	Time $\downarrow$	EM $\uparrow$	F1 $\uparrow$	Peak $\downarrow$	Time $\downarrow$	EM $\uparrow$	F1 $\uparrow$	Peak $\downarrow$	Time $\downarrow$
Qwen2.5-14B-Inst	0.734	0.904	15.45 $\pm$ 0.17	5.55 $\pm$ 0.17	1.552	1.872	44.52 $\pm$ 0.33	16.11 $\pm$ 0.25	0.567	0.704	38.18 $\pm$ 0.72	29.63 $\pm$ 0.62
Qwen2.5-7B-Inst	0.267	0.365	19.64 $\pm$ 0.33	4.65 $\pm$ 0.09	0.872	1.102	49.57 $\pm$ 0.40	14.04 $\pm$ 0.18	0.166	0.214	43.10 $\pm$ 0.53	15.58 $\pm$ 0.22
Qwen2.5-7B-Inst (A-MEM)	0.286	0.372	14.17 $\pm$ 0.11	24.66 $\pm$ 0.49	1.131	1.431	18.53 $\pm$ 0.11	54.16 $\pm$ 1.10	0.733	0.966	18.87 $\pm$ 0.15	92.21 $\pm$ 2.45
Qwen2.5-7B-Inst (truncate)	0.261	0.335	8.32 $\pm$ 0.05	6.02 $\pm$ 0.17	0.968	1.228	11.78 $\pm$ 0.10	11.63 $\pm$ 0.19	0.396	0.498	13.28 $\pm$ 0.14	22.12 $\pm$ 0.59
Search-R1	0.454	0.533	13.05 $\pm$ 0.08	4.29 $\pm$ 0.22	0.064	0.080	24.74 $\pm$ 0.17	4.33 $\pm$ 0.16	0.009	0.011	20.91 $\pm$ 0.03	4.66 $\pm$ 0.17
DeepResearcher	0.532	0.646	21.79 $\pm$ 0.41	4.01 $\pm$ 0.08	0.731	0.902	51.87 $\pm$ 0.38	11.16 $\pm$ 0.16	0.071	0.106	48.31 $\pm$ 0.60	15.79 $\pm$ 0.18
MEM1-QA (7B)	0.708	0.836	6.39 $\pm$ 0.02	6.49 $\pm$ 0.06	1.860	2.301	8.05 $\pm$ 0.06	8.61 $\pm$ 0.11	1.883	2.285	10.31 $\pm$ 0.08	8.71 $\pm$ 0.10
Mixed-horizon SFT (3B)	0.469	0.606	4.85 $\pm$ 0.08	4.44 $\pm$ 0.18	0.913	1.146	6.56 $\pm$ 0.33	14.43 $\pm$ 0.23	1.048	1.352	12.26 $\pm$ 0.05	18.23 $\pm$ 0.13
PPO (3B)	0.658 $\pm$ 0.015	0.806 $\pm$ 0.020	6.49 $\pm$ 0.03	5.53 $\pm$ 0.22	0.538 $\pm$ 0.015	0.655 $\pm$ 0.015	7.93 $\pm$ 0.65	22.32 $\pm$ 0.18	0.442 $\pm$ 0.011	0.549 $\pm$ 0.024	14.42 $\pm$ 0.16	24.60 $\pm$ 0.14
MemOPD (3B)	0.756$\pm$0.020	0.921$\pm$0.024	7.29 $\pm$ 0.03	5.91 $\pm$ 0.16	1.913$\pm$0.024	2.510$\pm$0.019	8.20 $\pm$ 0.11	20.77 $\pm$ 0.14	2.430$\pm$0.029	2.834$\pm$0.019	9.66 $\pm$ 0.09	21.14 $\pm$ 0.11

Table 1: Multi-objective multi-hop retrieval on Q2, Q8, and Q16. EM and F1 are summed across questions. Peak context is in 10^2 tokens and time is in seconds per trajectory. PPO and MemOPD report mean $\pm$ standard deviation over five seeds.

critic, and GAE to count one decision twice. Conversely, a correct action domain cannot repair logits computed with deleted history or wrong positions. State reconstruction preserves conditioning, while the action domain preserves which tokens count as decisions.

3.4 Teacher Guidance and Policy Optimization

Once the states and action domain are aligned, MemOPD combines local teacher guidance with task-level policy optimization. The student and frozen teacher receive the same tokens, positions, and visibility and share a tokenizer and vocabulary $\mathcal{V}$. For a supervised action position j containing token $a_{t,k}$, let $p_{\theta,j}$ and $p_{T,j}$ be the distributions read at $g(t,k)$ from the student and teacher. We use full vocabulary reverse KL divergence,

$$\mathcal{L}_{\text{OPD}} = \frac{1}{\sum_j m_j^T} \sum_j m_j^T \sum_{v \in \mathcal{V}} p_{\theta,j}(v) \log \frac{p_{\theta,j}(v)}{p_{T,j}(v)}. \quad (7)$$

The divergence itself is standard in language model distillation (Gu et al. 2024; Ko et al. 2024; Agarwal et al. 2024). Its role here is to expose the complete teacher distribution only after the student memory state has been recovered.

The reference policy is separate from both the teacher and the behavior snapshot. It remains frozen at the SFT initialization, whereas π_{old} is the policy that generated the current rollout batch. Following the KL regularization commonly used in reinforcement learning from human feedback (Ouyang et al. 2022), the reference policy constrains drift by adding a token-level KL penalty to the task reward. Let $\ell_{\text{old},j}$ and $\ell_{\text{ref},j}$ be the log probabilities assigned to the sampled action token by the behavior and reference policies, respectively. We define the reference KL penalty and the resulting token-level reward as

$$\text{KL}_j^{\text{ref}} = [\ell_{\text{old},j} - \ell_{\text{ref},j}]_+, \quad \tilde{r}_j = r_j - \beta \text{KL}_j^{\text{ref}}. \quad (8)$$

Here, β controls reference regularization. The scalar task reward is placed on the final sampled action token, while the reference penalty is defined at every sampled action position. The critic predicts values at these positions, and GAE propagates the resulting signal backward through the ordered student actions. Let $\tau_1, \dots, \tau_N$ be the positions for which $m^{\text{act}} = 1$. The recursion follows these decisions rather than adjacent storage positions:

$$\begin{aligned} \delta_n &= \tilde{r}_{\tau_n} + \gamma V_{\tau_{n+1}} - V_{\tau_n}, \\ A_{\tau_n} &= \delta_n + \gamma \lambda_{\text{GAE}} A_{\tau_{n+1}}. \end{aligned} \quad (9)$$

Here, γ is the discount factor and λ_{GAE} is the GAE decay parameter. The value after a terminal action is zero. Context copies and observations do not introduce additional decisions or value targets.

For each sampled action position j containing token $a_{t,k}$, let $\sigma_j = \sigma_{t,k}$ and $\tilde{\sigma}_j = \tilde{\sigma}_{t,k}$. Because another state would mix a policy change with a change in conditioning, both policies evaluate $a_j = a_{t,k}$ at $g(t,k)$ under the aligned state $\tilde{\sigma}_j = \sigma_j$. We therefore define $\varrho_j(\theta) = \pi_{\theta}(a_j | \tilde{\sigma}_j) / \pi_{\text{old}}(a_j | \tilde{\sigma}_j)$ and $\bar{\varrho}_j = \text{clip}(\varrho_j, 1 - \epsilon, 1 + \epsilon)$, where ϵ is the PPO clipping threshold. The PPO loss is

$$\mathcal{L}_{\text{PPO}} = - \frac{1}{\sum_j m_j^{\text{act}}} \sum_j m_j^{\text{act}} \min(\varrho_j A_j, \bar{\varrho}_j A_j). \quad (10)$$

The combined actor objective is

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{PPO}} - \alpha \mathcal{H} + \lambda \mathcal{L}_{\text{OPD}}, \quad (11)$$

where $\mathcal{H}$ is policy entropy, and α and λ control entropy regularization and teacher guidance, respectively. The teacher supplies dense local preferences, but it does not determine whether a sequence of memory updates and retrieval actions succeeds. PPO retains the task reward as the supervision signal for complete interaction success and can counteract local teacher preferences that harm the final answer.

4 Experiments

4.1 Experimental Setup

Benchmarks. Following MEM1, we use a multi-objective question answering benchmark constructed from HotpotQA and Natural Questions (Zhou et al. 2025; Yang et al. 2018; Kwiatkowski et al. 2019). Each example combines several questions into one query, and the agent must retrieve evidence over multiple turns from a local Wikipedia corpus before answering them all. Q2, Q8, and Q16 contain 2, 8, and 16 questions, creating progressively longer retrieval and memory update sequences. PPO and MemOPD are optimized on Q2, while Q8 and Q16 test transfer to longer horizons.

We also evaluate single-objective Wiki-RAG (Jin et al. 2025), where the agent repeatedly retrieves passages from a Wikipedia datastore before answering one open-domain question. This benchmark tests transfer from multi-objective retrieval to a standard single-objective setting.

System	EM $\uparrow$	F1 $\uparrow$	Peak $\downarrow$	Dependency $\downarrow$	Time $\downarrow$
Qwen2.5-7B-Inst (truncate)	0.296	0.386	6.31 ± 0.05	1.647 ± 0.04	2.241 ± 0.04
Qwen2.5-7B-Inst (A-MEM)	0.246	0.372	8.51 ± 0.11	0.932 ± 0.03	11.319 ± 0.36
Qwen2.5-7B-Inst	0.271	0.390	9.26 ± 0.21	1.151 ± 0.05	2.302 ± 0.04
Qwen2.5-14B-Inst	0.431	0.533	8.99 ± 0.20	2.260 ± 0.09	6.870 ± 0.20
Search-R1	0.452	0.518	11.12 ± 0.27	1.473 ± 0.05	2.218 ± 0.14
DeepResearcher	0.428	0.512	13.43 ± 0.36	6.844 ± 0.32	3.780 ± 0.08
MEM1-QA (SFT)	0.302	0.356	6.57 ± 0.05	3.296 ± 0.13	4.943 ± 0.20
MEM1-QA	0.409	0.470	5.62 ± 0.03	0.742 ± 0.02	3.751 ± 0.07
PPO (3B)	0.409 ± 0.020	0.486 ± 0.021	6.00 ± 0.05	1.049 ± 0.190	2.781 ± 0.053
MemOPD (3B)	0.434 ± 0.022	0.522 ± 0.030	6.13 ± 0.05	0.717 ± 0.028	1.988 ± 0.048

Table 2: Answer quality and efficiency on Wiki-RAG. Peak context and dependency are in 10^2 and 10^5 , respectively, and time is in seconds per trajectory. PPO and MemOPD report mean $\pm$ standard deviation over five seeds.

Metrics. Exact match (EM) and token F1 measure answer quality. For Q2, Q8, and Q16, both metrics are summed over the questions in each example. We also report peak context, dependency, and inference time per trajectory. PPO and MemOPD results are the mean and standard deviation over five independently trained seeds; efficiency uncertainty for fixed models is the standard error across test trajectories. Dependency is $\sum_s |s|(|p_s| + \lfloor |s|/2 \rfloor)$ over generated segments s , where p_s is the visible prefix preceding that segment.

Models and training. Our student is Qwen2.5 3B (Yang et al. 2025). We use gpt-oss-120b (OpenAI 2025) to generate agent trajectories, retaining only those with correct final answers and valid interaction formats. We convert each response turn into the corresponding model call format, yielding 20,036 turn-level examples. The student receives one epoch of SFT on this data before PPO or MemOPD training. Across five seeds, PPO and MemOPD use the same initialization, data order, reconstructed packed calls, sampled-action mask, and evaluation protocol. MemOPD additionally uses the frozen 7B teacher with $\lambda = 0.02$, selected on the Q2 development evaluation. For the matched Q2 control, PPO omits teacher guidance, while the two OPD variants use the same frozen teacher and coefficient and differ only in whether teacher scoring uses persistent history or reconstructed call states. All training runs use four NVIDIA H200 GPUs.

4.2 Multi-Objective Retrieval

Table 1 compares our 3B training variants with established systems on Q2, Q8, and Q16.

MemOPD-3B achieves the highest EM and F1 at all horizons. Averaged over five seeds, MemOPD improves F1 over PPO by 14.3%, 283.2%, and 416.2% on Q2, Q8, and Q16, respectively. On Q16, it reduces peak context by 33.0% and inference time by 14.1%. The growing advantage shows that aligned teacher guidance is more useful at longer horizons.

4.3 Single-Objective Wiki-RAG Transfer

Table 2 compares the resulting PPO and MemOPD models with established systems on Wiki-RAG.

On Wiki-RAG, MemOPD improves EM and F1 over PPO by 6.1% and 7.4%, respectively, when averaged over five seeds. It also has the lowest dependency and inference time

Representation	p99 $ \Delta \log p \downarrow$	Top-1 $\uparrow$	False clip $\downarrow$
Persistent history	1.7738	94.61	13.29
Batching numerical floor	5.31×10^{-5}	100.00	0.00
Reconstructed packing	3.43×10^{-5}	100.00	0.00

Table 3: State reconstruction on 64 student trajectories. Errors compare the same sampled actions with independent invocations in float32. False clipping is the fraction of unchanged-policy ratios outside the PPO clipping interval.

Context update	Tokens	p99 $ \Delta \log p \downarrow$	Top-1 $\uparrow$
Full response	3,329	3.43×10^{-5}	100.00
Suffix retention	3,329	3.83×10^{-5}	100.00
Summary replacement	3,329	3.59×10^{-5}	100.00
Sliding window	3,329	3.72×10^{-5}	100.00
Retrieval refresh	3,329	3.81×10^{-5}	100.00
MEM1	12,083	3.43×10^{-5}	100.00

Table 4: RCE across five controlled context updates and native MEM1 trajectories.

in the table. Relative to PPO, MemOPD reduces dependency by 31.6% and inference time by 28.5%, while peak context increases by 2.2%.

4.4 State Alignment Verification

We next test the mechanism behind these gains by scoring the same recorded actions under persistent history, independent invocations, and reconstructed packing. Independent invocations reproduce the original call states and serve as the reference, allowing us to measure how each training representation changes the action probabilities. The audit contains 64 real trajectories, 199 model invocations, and 12,083 sampled action tokens.

Persistent history changes the top prediction at 651 action positions and falsely clips 13.29% of the actions even before any policy update. Reconstructed packing instead matches independent invocations at the numerical floor. The mismatch also changes the supervision from the actual 7B teacher. Before the first memory update, exact and persistent states give identical teacher distributions. Across later invocations,

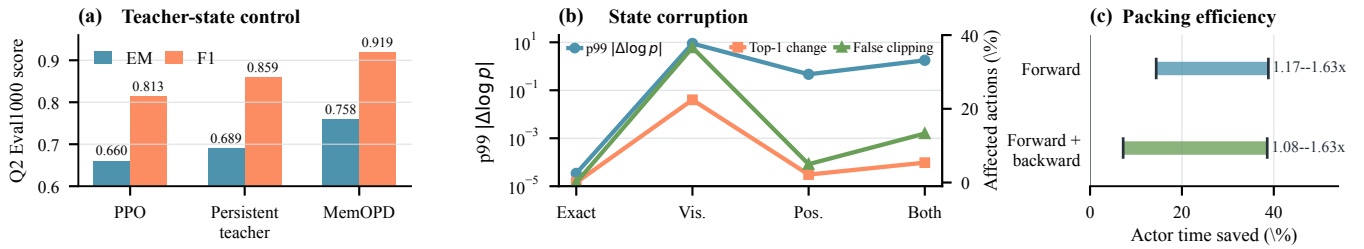


Figure 3: Experiment diagnostics. (a) Q2 results for PPO, persistent-history OPD, and MemOPD. (b) Log probability error, top-1 changes, and false clipping under state corruption. (c) Actor time saved relative to independent invocations.

however, top-1 agreement falls to 81.20%, the p99 sampled log probability difference reaches 19.854, and the mean KL divergence reaches 0.793, changing the teacher prediction at 444 action positions. Having established this effect on MEM1 trajectories, we next test whether the same reconstruction interface supports other forms of context update.

Table 4 tests whether the interface depends on one memory topology. In every case, the compiler receives the realized token context of each invocation rather than symbolic memory roles. It therefore preserves RCE across several context updates, while MEM1 remains the complete downstream training instance.

4.5 Ablation Studies

Matched teacher-state control. We compare PPO, OPD with persistent-history teacher scoring, and MemOPD under the same Q2 protocol. PPO uses reconstructed student-side calls and the sampled-action mask without teacher guidance. The persistent-history variant adds the frozen teacher but scores sampled actions under a flattened interaction history, whereas MemOPD scores them under reconstructed call states. All other training and evaluation settings are fixed. As shown in Figure 3a, persistent-history teacher scoring improves F1 and EM over PPO by 5.6% and 4.4%, respectively, showing that dense teacher guidance remains useful despite the state mismatch. Reconstructing the teacher state further improves F1 and EM by 7.0% and 10.0%, respectively. Overall, MemOPD improves F1 and EM over PPO by 13.0% and 14.8% in this matched control.

Reconstruction requirements. Figure 3b independently corrupts visibility and positions while keeping the recorded actions fixed. Visibility is the larger source of error, but incorrect positions alone still change 260 top predictions and falsely clip 4.97% of the actions. State reconstruction is also distinct from identifying which tokens are decisions. The exact action domain selects 12,083 sampled action tokens, whereas trajectory and response masks add 66,830 and 74,357 spurious positions.

Packing Efficiency. Figure 3c compares reconstructed packing with correct independent invocations across nine configurations. Packing achieves up to a $1.63\times$ actor speedup while preserving RCE. This speedup comes from shared computation without changing the optimized states.

5 Discussion

MemOPD separates action provenance from state validity. A student rollout guarantees that the learner generated each action, but compact memory may delete, replace, or reencode its conditioning context before training. The resulting batch can therefore contain genuine student actions evaluated under states that never occurred during interaction. Memory state alignment enforces this requirement, and RCE audits the reconstructed logits against independent invocations.

Teacher guidance and PPO serve different roles in the objective. The teacher provides dense guidance at sampled action tokens, whereas PPO evaluates whether the complete interaction solves the task. The matched control makes this distinction measurable: persistent-history teacher scoring improves F1 over PPO by 5.6%, showing that teacher guidance remains useful before state alignment, while reconstructing the teacher state adds a further 7.0% improvement.

Beyond the optimization objective, state correctness also determines how packing can be used. MemOPD restores each invocation before sharing stable prefixes across otherwise independent computations. RCE verifies that this sharing preserves the action logits, so the measured speedup does not depend on an approximate rollout state.

The same reconstruction interface is not tied to semantic roles defined by MEM1. Because the compiler consumes realized token contexts rather than a fixed memory topology, it preserves RCE under full response retention, suffix retention, summary replacement, sliding windows, and retrieval refresh. These results show that memory state alignment depends on the actual computation presented to the model rather than the particular form of memory rewriting, allowing MemOPD to support diverse context update mechanisms through a unified training interface.

6 Conclusion

Compact memory can separate sampled actions from the states later used to train them. MemOPD resolves this mismatch by reconstructing every invocation, distinguishing sampled actions from later context copies, and combining state-aligned teacher guidance with task-level PPO. RCE verifies the packed computation, while the matched control shows that teacher guidance already improves F1 over PPO by 5.6% and state alignment adds a further 7.0%. Across five seeds, MemOPD improves F1 over PPO by up to 416.2%, while packing yields up to a $1.63\times$ speedup in actor computation during training.

References

- Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Ramos, S.; Geist, M.; and Bachem, O. 2024. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *Proceedings of ICLR*.
- An, C.; Zhang, J.; Zhong, M.; Li, L.; Gong, S.; Luo, Y.; Xu, J.; and Kong, L. 2025. Why Does the Effective Context Length of LLMs Fall Short? In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Driessche, G. B. V. D.; Lespiau, J.-B.; Damoc, B.; Clark, A.; Casas, D. D. L.; Guy, A.; Menick, J.; Ring, R.; Hennigan, T.; Huang, S.; Maggiore, L.; Jones, C.; Cassirer, A.; Brock, A.; Paganini, M.; Irving, G.; Vinyals, O.; Osindero, S.; Simonyan, K.; Rae, J.; Elsen, E.; and Sifre, L. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *Proceedings of the 39th International Conference on Machine Learning*.
- Cao, Z.; Cao, Q.; Lu, Y.; Peng, N.; Huang, L.; Cheng, S.; and Su, J. 2024. Retaining Key Information under High Compression Ratios: Query-Guided Compressor for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Chhikara, P.; Khant, D.; Aryan, S.; Singh, T.; and Yadav, D. 2025. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413.
- Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129: 1789–1819.
- Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *The Twelfth International Conference on Learning Representations*.
- Gur, I.; Furuta, H.; Huang, A.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2024. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.
- Jiang, H.; Wu, Q.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2023. LLMingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 13358–13376.
- Jiang, H.; Wu, Q.; Luo, X.; Li, D.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2024. LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1658–1677.
- Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arik, S.; Wang, D.; Zamani, H.; and Han, J. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. arXiv:2503.09516.
- Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Kim, Y.; and Rush, A. M. 2016. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1317–1327.
- Ko, J.; Kim, S.; Chen, T.; and Yun, S.-Y. 2024. DistiLLM: Towards Streamlined Distillation for Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 24872–24895. PMLR.
- Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Kelcey, M.; Devlin, J.; Lee, K.; Toutanova, K. N.; Jones, L.; Chang, M.-W.; Dai, A.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.
- Lee, K.-H.; Chen, X.; Furuta, H.; Canny, J.; and Fischer, I. 2024. A Human-Inspired Reading Agent with Gist Memory of Very Long Contexts. In *The Twelfth International Conference on Learning Representations*.
- Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*.
- Li, R.; Zhang, X.; Yu, H.; Duan, S.; Li, X.; Xiang, W.; Liao, C.; Guo, X.; Li, Y.; and Suo, J. 2026a. MemPO: Self-Memory Policy Optimization for Long-Horizon Agents. arXiv:2603.00680.
- Li, X.; Lyu, T.; Yang, Y.; Shan, L.; Yang, S.; Zhang, L.; Huang, Z.; Liu, Q.; and Li, Y. 2026b. Escaping the Context Bottleneck: Active Context Curation for LLM Agents via Reinforcement Learning. arXiv:2604.11462.
- Li, Y.; Dong, B.; Guerin, F.; and Lin, C. 2023. Compressing Context to Enhance Inference Efficiency of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.
- Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2022. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332.
- OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925.

- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*.
- Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S. G.; Stoica, I.; and Gonzalez, J. E. 2024. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560.
- Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.
- Ross, S.; Gordon, G. J.; and Bagnell, J. A. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, 627–635. PMLR.
- Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.
- Sun, W.; Lu, M.; Ling, Z.; Liu, K.; Yao, X.; Yang, Y.; and Chen, J. 2025. Scaling Long-Horizon LLM Agent via Context-Folding. arXiv:2510.11967.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; Zhao, W. X.; Wei, Z.; and Wen, J.-R. 2024. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science*, 18(6): 186345.
- Wang, X.; Li, M.; Lu, P.; Chang, X.-W.; Shang, L.; Li, J.; Mi, F.; Parthasarathi, P.; and Cui, Y. 2026. InfMem: Learning System-2 Memory Control for Long-Context Agent. arXiv:2602.02704.
- Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.-W.; and Yu, D. 2025. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations (ICLR)*.
- Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; and Zhang, Y. 2025. A-MEM: Agentic Memory for LLM Agents. In *Advances in Neural Information Processing Systems*.
- Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.
- Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2022. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Yoon, C.; Lee, T.; Hwang, H.; Jeong, M.; and Kang, J. 2024. CompAct: Compressing Retrieved Documents Actively for Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yu, Y.; Yao, L.; Xie, Y.; Tan, Q.; Feng, J.; Li, Y.; and Wu, L. 2026. Agentic Memory: Learning Unified Long-Term and Short-Term Memory Management for Large Language Model Agents. arXiv:2601.01885.
- Zheng, Y.; Fu, D.; Hu, X.; Cai, X.; Ye, L.; Lu, P.; and Liu, P. 2025. DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments. arXiv:2504.03160.
- Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2023. MemoryBank: Enhancing Large Language Models with Long-Term Memory. arXiv:2305.10250.
- Zhou, Z.; Qu, A.; Wu, Z.; Kim, S.; Prakash, A.; Rus, D.; Zhao, J.; Low, B. K. H.; and Liang, P. P. 2025. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. arXiv:2506.15841.