COMPRESSED STEP INFORMATION MEMORY FOR END-TO-END AGENT FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM) agents excel in tasks like translation, code generation, and decision-making, but consecutive tool calls in complex scenarios lead to excessively long contexts. Despite SOTA LLMs' 128K+ token context windows, unstructured data interactions easily exceed limits, harming task focus and increasing resource costs. Existing solutions have flaws: forced truncation causes information loss, external memory modules lack end-to-end optimization, and context summarization wastes KV cache and loses data. To address this, we propose Compressed Step Information Memory (CSIM), an end-to-end context management method. It compresses post-step context to minimize information loss, retells/updates plans to avoid forgetting and correct errors. Trained via SFT and RL, CSIM achieves strong performance on Gaia and Browsecomp. Our contributions: (1) CSIM boosts performance in multi-tool scenarios; (2) A data synthesis and SFT/RL framework distills SOTA agent capabilities; (3) Experiments validate the method on multiple benchmarks.

1 INSTRUCTION

Large language model (LLM) agents have demonstrated remarkable capabilities. They bring substantial convenience to users across diverse tasks, such as translation, text generation for writing, code generation for programming, providing accurate answers in information-seeking scenarios, and even assisting in making complex decisions. These agents can interact with the environment, acquire information by invoking external tools, conduct reasoning, and ultimately solve users' problems.

However, when confronted with intricate problems or scenarios, agents often need to call external tools consecutively, sometimes even dozens of times. This leads to an extremely long context for LLMs. Although state-of-the-art (SOTA) LLMs can offer a context window of 128K tokens or more, in practical scenarios, this still falls short due to some critical pain points. For example, after interacting with multiple web pages or other unstructured data, agents can generate extremely long observations which easily exceed the context limit. Even if the large model technically supports such a large context window, when the context is too long, the agent struggles to focus on the problem it currently needs to solve, resulting in performance degradation. In addition, a long context implies more tokens, which in turn means higher resource consumption.

Faced with the context-length challenge, a relatively simple and straightforward approach is to truncate the context once it exceeds a predefined threshold. However, cutting off the context forcibly will inevitably lead to information loss. Once information is lost, the agent may lose the logical connection with the previous task process, fail to understand the context of the current operation, and even make wrong decisions or repeat ineffective operations. Therefore, context engineering has become a crucial factor in enhancing the performance of LLM agents.

To address this issue, researchers have proposed various solutions. For example, some methods introduce external memory module as an aid, pre-construct a structured knowledge base and retrieve relevant information using keywords for making better decisions. However, these are usually trained separately and cannot be optimized end-to-end. Other methods summarize previous memories and newly obtained information after each interaction to help the agent understand the context. Nevertheless, this compression causes partial information loss and fails to effectively utilize the key-value(KV) cache.

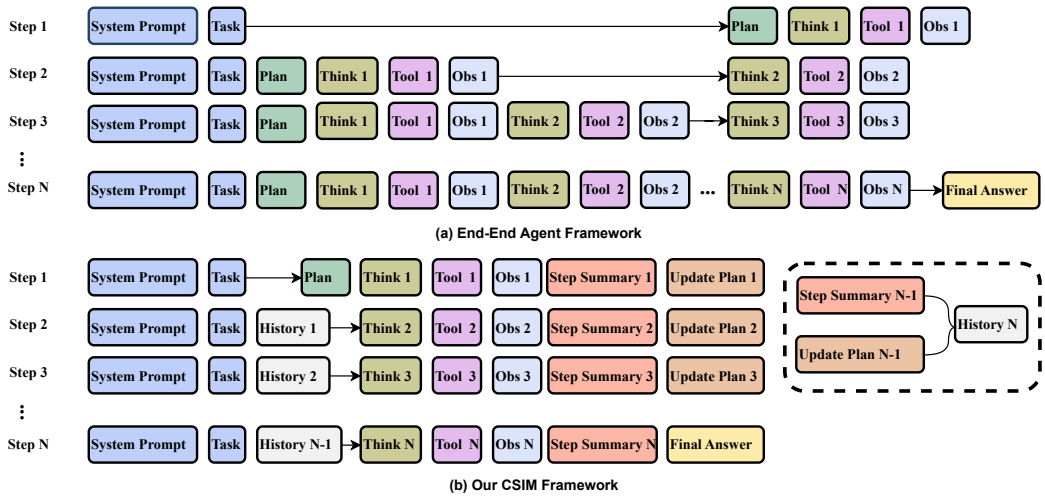

Figure 1: Overview of the training framework. (I) The SFT stage utilizes reformatted ReAct data with both short and long chains of thought for cold start. (II) The RL stage performs tool-aware rollouts on unused QA pairs and optimizes the policy.

In response to these situations, we propose an end-to-end context memory management method called **Compressed Step Information Memory(CSIM)** which is based on step information compression. Our method compresses and summarizes the context memory after each step finished, minimizing information loss to the greatest extent possible. Meanwhile, it retells the plan and updates it within a certain limit. This not only prevents the agent from forgetting the original plan due to a long context, but also effectively corrects potential errors in the original plan. We train the model through end-to-end supervised fine-tuning(SFT) and reinforcement learning (RL), achieving excellent results on benchmarks such as Gaia and Browsecomp.

In summary, our core contributions are as follows:

- We propose an end-to-end context memory management method CSIM based on step information compression, which effectively improves the model's performance, especially in multiple tool-calling scenarios.
- We design a data synthesis workflow and subsequent training framework for SFT and RL, enabling to distillate the capabilities of SOTA agents into end-to-end agent models.
- We train the model using the proposed method and framework, and achieve excellent results on multiple benchmarks.

2 RELATED WORK

2.1 CONTEXT MEMORY MANAGEMENT FOR MULTI-TOOL-CALLING AGENTS

Efficient context memory management is critical for LLM agents to maintain reasoning coherence in multi-turn, tool-intensive tasks Zhou et al. (2025). Existing methods are categorized by their structure design and update mechanisms:

Static Structure Design Methods. These methods rely on fixed architectures for memory organization. MemOS unifies three memory types (plaintext, activation, parameter) into a "MemCube" to enable cross-type conversion Li et al. (2025c), but lacks dynamic compression for step-wise tool interaction history. MemoryOS adopts a three-tier structure (short-term dialogue chains, mid-term topic segments, long-term personas) with heat-based memory promotion Kang et al. (2025); similarly, MemGPT Packer et al. (2023) uses cache-like prioritization and SCM Liang et al. (2023) uses dual buffers, but all suffer limited generalization in diverse tool-calling scenarios due to fixed workflows Xu et al. (2025).

Dynamic Update Mechanisms. These methods optimize memory content via adaptive updates. MEM1 maintains a single buffer updated with prior context and environment observations Zhou et al. (2025), but risks information loss in long sequences. A-MEM links new memories to existing ones via similarity scores and evolves content Xu et al. (2025), but decouples retrieval optimization from decision-making. Agent KB uses a pre-built structured knowledge base and a teacher-student-execution pipeline for reasoning Tang et al. (2025), but the KB cannot be updated end-to-end with real-time interactions. Intrinsic Memory Agents use a unified template with agent-specific memory maintainers Yuen et al. (2025), but increase system complexity and fail to resolve context bloat in single-agent multi-tool tasks, a core target of CSIM.

2.2 Context Compression for Efficient Reasoning

To mitigate context window limits, research follows two paths; CSIM belongs to the latter and addresses gaps in dynamic tool-calling scenarios:

Architecture-Driven Long-Context Methods. These methods modify LLM architectures to handle longer sequences. RoPE extrapolation (NTK Peng & Quesnelle (2023), YaRN Peng et al. (2023)) adjusts positional embeddings Yu et al. (2025a), while linear attention Child et al. (2019) and State Space Models Gu & Dao (2023); De et al. (2024) achieve O(N) complexity. However, they require invasive architectural changes and only adapt to static texts, not dynamic tool interaction histories.

Content-Driven Compression Methods. These methods distill context without architectural modifications, focusing primarily on fixed scenarios: MemAgent Yu et al. (2025a) compresses static documents via KV cache retention, and Agent KB Tang et al. (2025) encodes long texts into structured embeddings—both aiming to preserve output consistency with full contexts. Yet they are ill-suited for multi-tool-calling agents: they target static documents rather than dynamic interaction histories (e.g., tool calls, real-time feedback Zhou et al. (2025)) and underutilize KV caches, leading to redundant computation in multi-step tasks.

In particular, relevant memory management works incorporate partial compression-like mechanisms, but are not tailored for dynamic agent scenarios. MEM1 Zhou et al. (2025) iteratively summarizes context, queries, and observations into a single buffer to maintain continuity, but this design may lead to gradual information dilution in extended tool-calling chains. A-MEM Xu et al. (2025) uses similarity-based pruning to structure memory, yet its focus on retrieval optimization means it does not prioritize retention of task-critical KV patterns. Such limitations highlight the need for a compression strategy tailored to multi-tool-calling agents, which CSIM addresses: its step-wise mechanism targets full interaction histories (tool calls, feedback, reasoning logs), selectively preserving KV-relevant details (e.g., tool output validity, plan adjustment logic) to balance information retention and efficiency—resolving gaps in existing methods.

3 Preliminary

Target scenario and notation: Let $\mathcal{T}$ denote a space of texts. We focus on the agentic task scenario, aiming to predict the output $O(t) \in \mathcal{T}$ based on the input $I(t) \in \mathcal{T}$ and the accumulated context $C(t) = [c(1), \ldots, c(t)]$ for reasoning step $t \in \{1, \ldots, T\}$, where $T \in \mathbb{N}$ represents the maximum number of reasoning steps. Here, $c(t) \in \mathcal{T}$ denotes a newly integrated context at reasoning step t, which comprises the interaction results from the preceding reasoning step t-1, including $I(t\text{-}1)$, $O(t\text{-}1)$, and any additional tool feedback. We represent the dataset with multiple identities as $\mathcal{D} = \{(C_i(t), I_i(t), O_i(t)) \mid i \in \mathcal{I}, t \in \{1, \ldots, T\}\}$, where $\mathcal{I}$ denotes an index set of identities.

Table 1: Illustrative instances of online inference scenarios.

Application	Dataset	Context $C(t)$	Input $I(t)$	Output $O(t)$
Agentic task	GAIA Mialon et al. (2023)	State history	Current action	Next action

Context compression: Let us consider a large language model $f_\theta : \mathcal{T} \to \mathbb{R}^+$, which models the probability distribution over the text space $\mathcal{T}$. A typical approach for predicting output $O(t)$ involves

using the full context $C(t)$ as $\hat{O}(t) \sim f_\theta(\cdot \mid C(t), I(t))$. However, this approach requires increasing memory and computation costs over time for maintaining and processing the entire context $C(t)$. One can employ context compression techniques to mitigate this issue, compressing contexts into a shorter sequence of attention key/value pairs or soft prompts (Mu et al., 2023; Ge et al., 2023). Given the compression function g_{comp}, the inference with compressed contexts becomes $\hat{O}(t) \sim f_\theta(\cdot \mid g_{\text{comp}}(C(t)), I(t))$, where $|g_{\text{comp}}(C(t))| \ll |C(t)|$. It is worth noting that existing context compression methods mainly focus on compressing a fixed context $\bar{C}$ that is repeatedly used as a prompt (Mu et al., 2023; Ge et al., 2023). The objective of the compression is to generate outputs for a given input I that are similar to the outputs generated when using the full context: $f_\theta(\cdot \mid g_{\text{comp}}(\bar{C}), I) \approx f_\theta(\cdot \mid \bar{C}, I)$.

4 METHODS: CSIM

We focus on efficiently understanding long contexts while utilizing short context windows, thereby avoiding the quadratic complexity associated with attention mechanisms in long sequences. This approach imposes specific requirements on memory design: it must not only ensure smooth transitions between windows but also retain critical information from all previous windows.

4.1 SFT FOR COLD START

Compressed Context Agent Knowledge Distillation: Our approach leverages agent-level knowledge distillation to transfer capabilities from state-of-the-art multi-agent systems into chain-of-agents trajectories. This method extends sequence-level knowl- edge distillation principles [24] to the multi-agent domain, where we distill the sequential decision-making patterns of expert multi-agent systems rather than word-level distributions.

Progressive Quality Filtering: filtering high trajectory for SFT.

SFT Training: To avoid interference from external feedback during learning, we mask out loss contributions from tokens representing external feedback. Given the task context $\mathbf{tc}$ and the complete decision-making trajectory $x = [x_1, x_2, \ldots, x_n]$, where each $x_i \in \{\langle\text{think}\rangle, \langle\text{action}\rangle, \langle\text{answer}\rangle\}$, the loss function at this stage is computed as follows:

$$L = \frac{1}{|x|} \sum_{i=1}^{|x|} I_{x_i \neq \text{o}} \left[\log \pi_\theta(x_i \mid \mathbf{tc}, x_{<i})\right]$$

Here, $I_{x_i \neq \text{o}}$ indicates tokens that do not correspond to external feedback, and we only consider the numerical terms associated with these tokens in the loss calculation.

4.2 REINFORCEMENT LEARNING FOR COMPRESSED CONTEXT

RL Training Data: Given the heterogeneous quality distribution across our integrated diverse data sources, we implement a multi-stage filtering protocol to ensure query quality. This curation strategy addresses data variance through quality filter and strategic sampling.

Quality filter: We employ `Qwen-2.5-72B-Instruct` Qwen et al. (2025) to evaluate question solvability without tool assistance. For each query $\mathbf{q}$ in the QA dataset:

$$r_q = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\text{EM}(a_i, y_{\text{gt}}) = 1\right] \tag{1}$$

where $N = 32$ is the number of model predictions, a_i denotes the i-th prediction, y_{gt} represents the ground truth, and $\text{EM}(\cdot)$ computes the exact match score between two inputs. This pass rate r_q quantifies parametric knowledge contamination risk. Queries with $r_q > 0.3$ are excluded as they either represent: 1) Trivially solvable cases requiring no tool usage, or 2) Highly contaminated samples vulnerable to parametric recall. This threshold ensures genuine tool engagement.

Strategic sampling: We adopt a random selection strategy to sample queries from the remaining challenging ones (with $r_q \leq 0.3$), which are ultimately used for RL training:

$$\mathcal{Q}_{\text{RL}} = \{q_j \mid r_{q_j} \leq 0.3\}_{j=1}. \tag{2}$$

The sampled subset, which is excluded from the SFT dataset, forms the final RL dataset. This composition focuses on queries where tool-based reasoning offers substantial value. By design, the strategic sampling ensures that the RL training emphasizes challenging cases in which effective tool coordination is critical, while reducing the influence of trivial or potentially useless samples.

Reward Function Design: Reward signals are critical for shaping RL dynamics in open-ended web agent tasks. Our framework adopts a streamlined design, built on two key considerations: Format consistency is inherently ensured through high-quality supervised fine-tuning and effective cold-start, obviating the need for explicit format validation rewards (e.g., prior $score_{format}$). For evaluating answer correctness, traditional rule-based metrics (F1, EM) fail to capture the nuance of diverse valid outputs in open-ended tasks. Instead, we use LLM-as-JudgeZheng et al. (2023), where judge model M_j provides binary assessments. Our reward function is:

$$\mathcal{R}_{\text{web}}(\tau) = score_{\text{answer}} \tag{3}$$

where $score_{\text{answer}} \in \{0, 1\}$ is 1 if M_j judges the final prediction correct. This design prioritizes core correctness, avoids instability from fragmented rewards, mitigates reward hacking via binary signals, and enables flexible evaluation of diverse outputs through LLM judgment.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks: To assess performance on complex information retrieval tasks, we further evaluate on three specialized benchmarks: GAIA Mialon et al. (2023) (103 text-only examples for fair comparison with Li et al. (2025b); Wu et al. (2025)), BrowseComp Wei et al. (2025), and HLE Phan et al. (2025). These benchmarks collectively enable systematic assessment across diverse task typologies and complexity levels.

- **GAIA** Mialon et al. (2023) is a benchmark for General AI Assistants that evaluates multi-step reasoning and tool-use proficiency through real-world questions. While conceptually simple for humans (92% solve rate), these questions are challenging for AI systems. We use its text-only subset (103 validation samples) to ensure fair comparison with prior work Li et al. (2025b); Wu et al. (2025), requiring fundamental abilities including web browsing and tool orchestration.
- **BrowseComp** Wei et al. (2025) assesses advanced web navigation capabilities through deliberately obscure yet verifiable questions. It requires persistent, creative search strategies to locate hard-to-find information that cannot be discovered via simple queries or brute-force methods, with verification through short, factual answers. We evaluate on the full benchmark (1,266 examples).
- **HLE** Phan et al. (2025) is a frontier academic benchmark at the limits of human knowledge, featuring 2,500 multi-modal questions across mathematics, humanities, and natural sciences. These questions require expert-level reasoning and cannot be resolved through simple internet retrieval. For methodological consistency, we evaluate exclusively on its text-only subset (500 samples), which exposes significant capability gaps in state-of-the-art systems.

Metrics: Model performance is evaluated using the LLM-as-Judge method, with Qwen-2.5-72B serving as the judge Zheng et al. (2023); Sun et al. (2025); Wu et al. (2025). The judge provides binary correctness assessments for each prediction, yielding accuracy scores per dataset. The standardized judging prompt is detailed in xxx.

Implementation Details: Our experimental framework is implemented using the `Qwen-2.5` model family as the backbone architecture. Specifically, we evaluate the `32B-Instruct` variants Qwen et al. (2025) to analyze performance across different model scales. All models are configured with a maximum sequence length of 32768 tokens to support complex reasoning chains and

the integration of lengthy retrieved content. During inference, we set the generation temperature to 1.0, the top-p sampling threshold to 0.9, and the top-k sampling parameter to 20.

For SFT, we use a batch size of 256 for 2.5 epochs with a learning rate of 1.4e-5 and AdamW optimizer with cosine decay. The fine-tuning procedure is implemented using the LLaMA-Factory framework Zheng et al. (2024). Following established practice in prior work Jin et al. (2025); Sun et al. (2025), we mask external tool call outputs during fine-tuning to preserve the integrity of the learning process by excluding extraneous external knowledge. RL stage employs Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) Yu et al. (2025b) with the following protocol: Each training iteration processes 64 prompts, generating 8 rollouts per prompt through environment interaction. Each rollout permits up to 24 steps and 32k tokens followed by final answer generation. We use the VeRL framework Sheng et al. (2024) for DAPO training.

Baselines: For GAIA, WebWalker, BrowseComp, and HLE benchmarks, we compare against:

- Direct Inference: For complex web tasks, we evaluate against more advanced baseline LLMs, including Qwen2.5-32B-Instruct Qwen et al. (2025), QwQ-32B Team (2024), and Deepseek-R1-671B Guo et al. (2025).
- Agent Framework: We additionally compare against two SOTA agent frameworks: OAgents Zhu et al. (2025) and OWL Hu et al. (2025), which are widely recognized for their strong performance in web agent tasks.
- Tool-integrated Frameworks: We compare against specialized web agents including: Search-o1 Jin et al. (2025), R1-Searcher Song et al. (2025), WebThinker Li et al. (2025b), SimpleDeepSearcher Sun et al. (2025), WebDancer Wu et al. (2025), WebSailor Li et al. (2025a), and WebShaper Tao et al. (2025).

All baselines utilize publicly available implementations with performance reported for their optimal configurations. To ensure fair comparison while isolating architectural contributions, we use the same backbone models (Qwen-2.5-7B/32B-Instruct or QwQ-32B) across all methods where applicable.

5.2 Experimental Results

Our result is as 2.

5.3 ABLATIONS

Complementary roles of step summary and update plan: xxx

Impact of update plan agent: Impact of update plan agent.

Impact of step summary prompt: Impact of step summary prompt.

Different compression methods: The impact of different compression methods.

6 CONCLUSION

References

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Table 2: Results on agentic benchmarks including GAIA, WebWalker, BrowseComp and HLE. We port Pass@1 metric for all tasks. Gray-highlighted values represent our reproduced results.

Method	Backbone	GAIA				WebWalker	BrowseComp	HLE
		Level 1	Level 2	Level 3	Avg.	Avg.	Avg.	Avg.
Model Inference								
Qwen2.5-32B-Instruct	-	12.8	3.8	0.0	6.8	3.1	0.6	5.4
QwQ-32B	-	30.8	15.4	25.0	22.3	4.3	0.5	9.6
Deepseek-R1-671B	-	43.6	26.9	8.3	31.1	10.0	2.0	8.6
Agent Frameworks								
OWL	GPT-4.1	71.0	50.0	28.6	53.6	10.2	-	6.4
OAgents	GPT-4.1	66.7	57.7	33.3	58.3	-	13.7	20.2
DeepResearch	-	74.3	69.1	47.6	67.4	-	51.5	26.6
Tool-integrated Methods								
R1-Searcher	Qwen-2.5-7B-Instruct	28.2	19.2	8.3	20.4	-	-	-
WebDancer	Qwen-2.5-7B-Instruct	41.0	30.7	0	31.0	36.0	-	-
WebSailor	Qwen-2.5-7B-Instruct	-	-	-	37.9	-	6.7	-
Ours-SFT	Qwen-2.5-7B-Instruct	36.5	33.3	16.7	34.0	-	-	-
Ours-RL	Qwen-2.5-7B-Instruct	53.8	32.7	33.3	40.8	55.6	5.8	15.6
Search-o1	QwQ-32B	53.8	34.6	16.7	39.8	34.1	-	10.8
WebThinker-Base	QwQ-32B	53.8	44.2	16.7	44.7	41.9	-	13.0
WebThinker-RL	QwQ-32B	56.4	50.0	16.7	48.5	46.5	2.8	15.8
SimpleDeepSearcher	QwQ-32B	-	-	-	50.5	-	-	-
WebDancer	QwQ-32B	61.5	50.0	25.0	51.5	47.9	3.8	7.2
WebShaper	QwQ-32B	69.2	50.0	16.6	53.3	49.7	-	-
Search-o1	Qwen-2.5-32B-Instruct	33.3	25.0	0.0	28.2	-	-	-
WebDancer	Qwen-2.5-32B-Instruct	46.1	44.2	8.3	40.7	38.4	-	-
SimpleDeepSearcher	Qwen-2.5-32B-Instruct	-	-	-	40.8	-	-	-
WebShaper	Qwen-2.5-32B-Instruct	61.5	53.8	16.6	52.4	51.4	-	-
WebSailor	Qwen-2.5-32B-Instruct	-	-	-	53.2	-	10.5	-
Ours-SFT	Qwen-2.5-32B-Instruct	56.4	51.9	25.0	50.5	61.5	10.0	16.3
Ours-RL	Qwen-2.5-32B-Instruct	69.2	50.0	33.3	**55.3**	**63.0**	**11.1**	**18.0**

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, Zeyu Zhang, Yifeng Wang, Qianshuo Ye, Bernard Ghanem, Ping Luo, and Guohao Li. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation, 2025. URL https://arxiv.org/abs/2505.23885.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. *ArXiv*, abs/2506.06326, 2025. URL https://api.semanticscholar.org/CorpusID:279250574.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating superhuman reasoning for web agent, 2025a. URL https://arxiv.org/abs/2507.02592.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025b.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan

Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Ke-Rong Li, Zhenzhen Tao, Junpeng Ren, Huayi Lai, Haoze Wu, Bo Tang, Zhenren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Ming-Zhou Yang, Tong Xu, Wei Xu, Huajun Chen, Haofeng Wang, Hongkang Yang, Wentao Zhang, Zhikun Xu, Siheng Chen, and Feiyu Xiong. Memos: A memory os for ai system. *ArXiv*, abs/2507.03724, 2025c. URL https://api.semanticscholar.org/CorpusID:280093879.

Xinnian Liang, Bing Wang, Huijia Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. Scm: Enhancing large language model with self-controlled memory framework. 2023. URL https://api.semanticscholar.org/CorpusID:258331553.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph Gonzalez. Memgpt: Towards llms as operating systems. *ArXiv*, abs/2310.08560, 2023. URL https://api.semanticscholar.org/CorpusID:263909014.

Bowen Peng and Jeffrey Quesnelle. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

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

Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, et al. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*, 2025.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. Agent kb: Leveraging cross-domain experience for agentic problem solving. *ArXiv*, abs/2507.06229, 2025. URL https://api.semanticscholar.org/CorpusID:280047833.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025. URL https://arxiv.org/abs/2504.12516.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *ArXiv*, abs/2502.12110, 2025. URL https://api.semanticscholar.org/CorpusID:276421617.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025a.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025b.

Sizhe Yuen, Francisco Gomez Medina, Ting Su, Yali Du, and Adam J. Sobey. Intrinsic memory agents: Heterogeneous multi-agent llm systems through structured contextual memory. *ArXiv*, abs/2508.08997, 2025. URL https://api.semanticscholar.org/CorpusID:280635601.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *ArXiv*, abs/2506.15841, 2025. URL https://api.semanticscholar.org/CorpusID:279465470.

He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, Yeyi Guan, Jinxiang Xia, Yi Yao, Hanhao Li, Ningning Wang, Pai Liu, Tianhao Peng, Xin Gui, Xiaowan Li, Yuhui Liu, Yuchen Eleanor Jiang, Jun Wang, Changwang Zhang, Xiangru Tang, Ge Zhang, Jian Yang, Minghao Liu, Xitong Gao, Wangchunshu Zhou, and Jiaheng Liu. Oagents: An empirical study of building effective agents, 2025. URL https://arxiv.org/abs/2506.15741.

A APPENDIX

You may include other additional sections here.

B RELATED WORKS

C DISCUSSIONS

D DATASET DETAILS

E EXPERIMENT SETUP

F ADDITIONAL EXPERIMENTAL RESULTS