Self-evolving Agents with reflective and memory-augmented abilities

Anonymous submission

Abstract

Large language models (LLMs) have made significant advances in the field of natural language processing, but they still face challenges such as continuous decision-making, lack of long-term memory, and limited context windows in dynamic environments. To address these issues, this paper proposes an innovative framework—Self-evolving Agents with Reflective and Memory-augmented Abilities (**SAGE**). The SAGE framework comprises three agents: the User, the Assistant, and the Checker. By integrating iterative feedback, reflective mechanisms, and a memory optimization mechanism based on the Ebbinghaus forgetting curve, it significantly enhances the agents' capabilities in handling multi-tasking and long-span information. The agents, through self-evolution, can adaptively adjust strategies, optimize information storage and transmission, and effectively reduce cognitive load. We evaluate the performance of the SAGE framework on multiple benchmarks and long text tasks. Experimental results show that SAGE significantly improves model performance, achieving a 2.26X improvement on closed-source models and an improvement ranging from 57.7% to 100% on open-source models, with particularly notable effects on smaller models.

Introduction

In recent years, large language models (LLMs) have made significant progress in the field of natural language processing, demonstrating powerful performance in tasks such as dialogue and text generation (Brown et al. 2020; He et al. 2025, 2024). Recently, there has been growing interest in applying LLMs as autonomous agents (LLM agents), which use language not only for understanding and generation, but also for planning and acting in interactive environments (Yao et al. 2023b; Shinn et al. 2023; Liang et al. 2024; Li et al. 2024; Zhou et al. 2024). However, these models still face several challenges: (1) LLM Agents need to continuously make decisions in changing environments and adapt to new situations and tasks. (2) LLM Agents lack long-term memory mechanisms, which is increasingly evident in situations requiring sustained interaction with the environment (Graves et al. 2016). The limited context window also hinders the model's ability to handle information over long time spans (Rae et al. 2019).

To tackle these challenges, researchers have proposed meta-learning and multi-task learning to enhance the transferability and adaptability of LLM agents. For memory limitations, prior works like MemGPT (Packer et al. 2024) use a FIFO queue to manage forgetting, while MemoryBank employs a forgetting curve based on insertion time. However, these approaches are often task-specific, lacking a general framework to systematically improve LLM agents in complex environments. Recent innovations, such as AutoGPT (Yang, Yue, and He 2023) and BabyAGI (Nakajima 2024), leverage LLMs as core controllers, aiming to solve real-world challenges. Yet, multi-agent frameworks still face issues like communication overload, heavily relying on memory to maintain context. As interaction history grows, resource demands and latency increase, limiting efficient deployment in practical scenarios.

In this paper, we propose an innovative framework, Self-evolving Agents with reflective and memory-augmented abilities (SAGE). By enhancing agents' self-adjustment capabilities through reflection, they can more effectively utilize historical information and make efficient decisions when faced with complex and dynamic tasks. From the perspective of self-evolution, we introduce a memory optimization mechanism based on the Ebbinghaus forgetting curv (Ebbinghaus 1885). This mechanism helps agents selectively retain key information, optimize information storage and transmission, reduce unnecessary cognitive load, and enhance agents' capabilities in interaction tasks with the environment. Experimental results demonstrate that our approach consistently enhances the performance of both proprietary and open-source LLMs across a wide range of benchmarks. The improvements are especially notable in smaller models, where the gains are more pronounced. On tasks such as multi-source question answering and code generation, our method sets a new standard, outperforming existing techniques and achieving leading benchmarks (Etezadi and Shamsfard 2023), including AgentBench (Liu et al. 2023).

The main contributions of our work are as follows:

- We propose a novel framework, SAGE, which introduces a reflection mechanism to enhance agents' self-adjustment capabilities. Without any additional training, this enables agents to utilize historical information more effectively and make better decisions when faced with complex and dynamic tasks.
- We introduce a memory optimization mechanism based on the Ebbinghaus forgetting curve. This helps agents selectively retain key information, reducing the issue of

information overload in multi-agent systems.

- SAGE achieves improvements over strong baselines in multiple challenging real-world tasks and attains state-of-the-art results on benchmarks. This framework can be applied to other LLMs, with particularly strong improvements in smaller models.

Related work

Self-Improvement of Reasoning and Decision-Making

Deep learning has transformed multiple domains including NLP, time series analysis and computer vision (Qiu et al. 2025a,b, 2024). A lot of research is focused on making large language models (LLMs) better at improving themselves. Some researchers are working on using carefully crafted prompts to help models learn how to get better, although this usually only works for one-off tasks. Others are tweaking how models get feedback during tasks, which helps them get better at thinking things through (Huang et al. 2022). There's also work on using strategies like random beam searches to help models make smarter decisions and assess their own work. Most current methods rely on quick, one-off tweaks and learning strategies that need lots of resources and hands-on tech help (Tian et al. 2024). This paper introduces a self-reflection mechanism, showing that LLMs can keep getting better and produce higher quality work across different tasks, all without needing extra training.

Memory Mechanism for LLM-based Agents

In LLM-based agents, the memory module stores, processes, and retrieves task-related information, supporting knowledge accumulation, experience handling, and decision-making. To enhance the self-evolution capabilities of these agents, researchers are focused on designing and optimizing these memory modules (Raffel et al. 2020). Past research has covered various designs and implementations of memory modules. This includes integrating information from different trials to boost reasoning abilities or storing information in natural language to enhance the module's interpretability and user-friendliness (Wada, Iwata, and Matsumoto 2019). Despite progress, self-adjustment and memory management still need improvement to handle complex real-world problems more effectively.

Method

In this section, we present the SAGE framework, designed to improve agent performance by leveraging three core mechanisms: iterative feedback, reflection, and MemorySyntax (as shown in Figure 1). The assistant agent A iteratively updates its policy π_θ based on feedback f_t provided by the checker agent C, optimizing over successive iterations to maximize the expected reward R. The reflection mechanism allows A to incorporate historical observations $\mathcal{O}_t$ and actions $\mathbf{a}_t$, forming a self-reflection r_t, which is stored in the memory $\mathcal{M}_L$ for future decision-making. Finally, MemorySyntax combines the Ebbinghaus forgetting curve with linguistic principles to manage memory decay, dynamically updating the agent's short-term memory $\mathcal{M}_S$ and long-term memory $\mathcal{M}_L$ by prioritizing information based on its retention strength $S(I_t)$, thus improving the agent's ability to retain crucial information while discarding less relevant data. The subsequent subsections detail these components.

Iterative Feedback

The iterative feedback mechanism in the SAGE framework enables the assistant agent A to refine its policy π_θ through repeated interactions with the checker agent C. At each iteration t, the assistant receives feedback f_t based on its current output $\mathbf{o}_t$, and adjusts its policy accordingly. This process continues until the checker validates the output or the iteration cap N is reached, ensuring that A incrementally optimizes its decisions to improve task performance over successive iterations.

Initialization Phase

- Role Assignment. In the SAGE framework, three agents are introduced: the user U, the assistant A, and the checker C. The user, upon receiving prompt P_U, assumes the role of task proposer by specifying a task $\mathcal{T}_U$ and related constraints $\mathcal{C}_U$. The assistant, upon receiving prompt P_A, generates a sequence of actions $\mathbf{a}_t$ based on the observations $\mathcal{O}_t$ and environment $\mathcal{E}$. The checker C evaluates the output $\mathbf{o}_A$ produced by the assistant, providing feedback f_C based on the discrepancy between $\mathbf{o}_A$ and the expected result, updating its policy π_θ iteratively to minimize this gap.
- Task Assignment. The task $\mathcal{T}_U$ provided by the user includes an initial task description $\mathbf{d}_U$ and an instance $\mathbf{i}_U$ that serves as the reference for correct output. This forms the input set $\mathcal{I}_A = (\mathbf{d}_U, \mathbf{i}_U)$ for the assistant to initiate its generative process. The assistant then proceeds by selecting an action $\mathbf{a}_t$ at each time step t, guided by π_θ, with the goal of maximizing the reward R_t for completing $\mathcal{T}_U$.

Actual Interaction Phase Following the role assignment and task definition in the initialization phase, the assistant A transitions into the actual interaction phase to generate outputs aimed at accomplishing the task $\mathcal{T}_U$. In this phase, A iteratively produces outputs $\mathbf{o}_t$ at each time step t based on the task description $\mathbf{d}_U$ and instance $\mathbf{i}_U$ provided in the input set $\mathcal{I}_A = (\mathbf{d}_U, \mathbf{i}_U)$. At each time step t, the assistant selects an action $\mathbf{a}_t$ by following its policy π_θ, which is conditioned on the current state s_t, the reward signal R_t (the reward score for task performance), and feedback f_t^i from the checker C. This decision-making process is formalized as:

$$\mathbf{o}_t \sim \pi_\theta(\mathbf{o}_t \mid s_t, R_t, f_t^i), \tag{1}$$

where π_θ represents the assistant's policy, R_t reflects the reward signal based on task performance at time t, and f_t^i is the feedback provided by the checker during the i-th iteration.

As the interaction progresses, the checker C evaluates each output $\mathbf{o}_t$ generated by A, comparing it against the expected outcome derived from $\mathbf{i}_U$. Based on this comparison, the checker provides iterative feedback f_t^i to guide A in refining its actions $\mathbf{a}_t$ and outputs $\mathbf{o}_t$. The iterative refinement

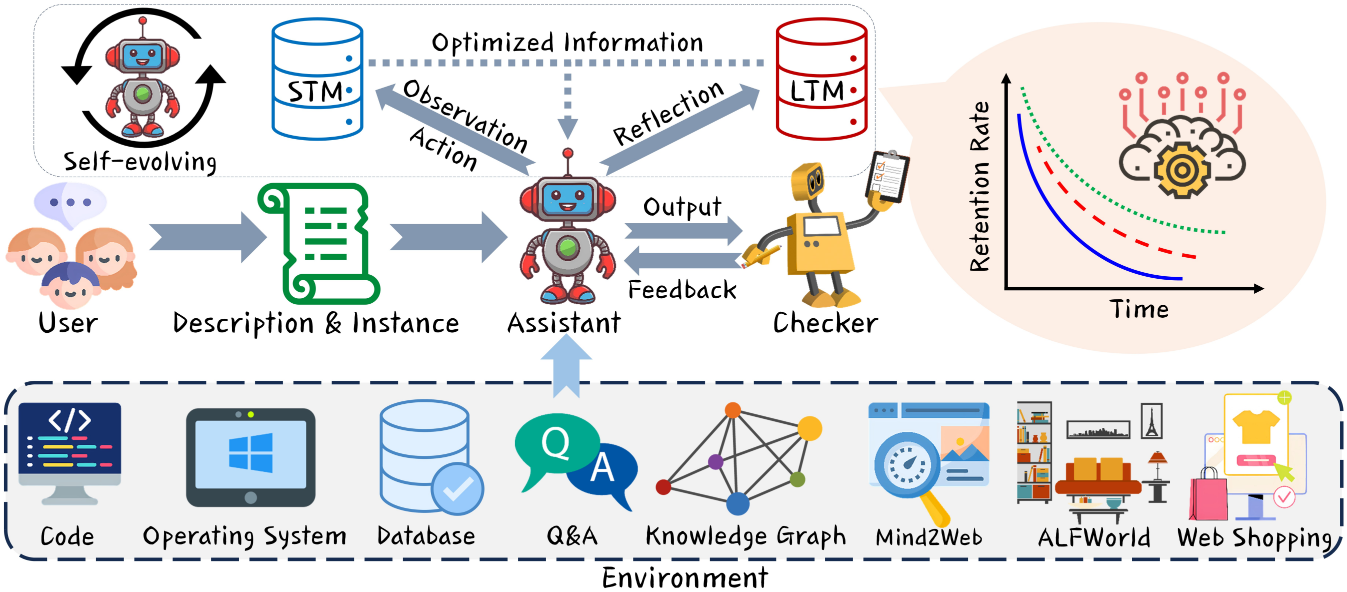

Figure 1: An illustration of the SAGE: a user provides a description and instance to the assistant with short-term (STM) and long-term (LTM) memory. The assistant performs observation, action, reflection, and output, which the checker reviews. The retention rate curve on the right illustrates memory decay over time, with a self-evolving loop guiding continued updates.

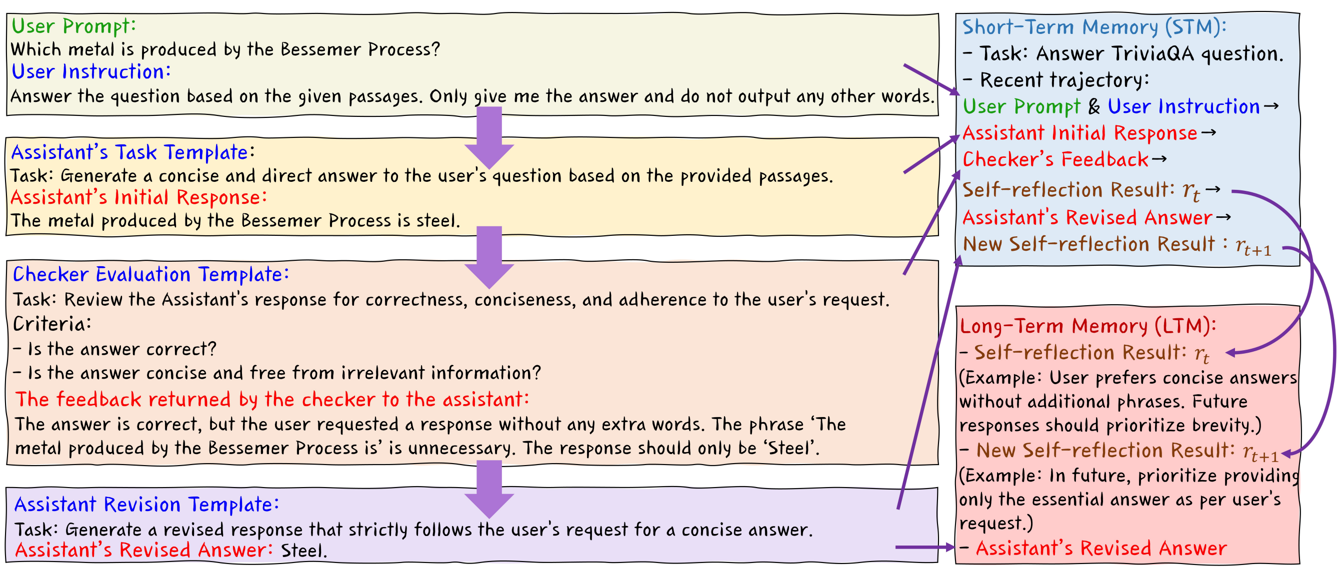

Figure 2: An example of the assistant's iterative workflow, including checker evaluation, prompt templates for feedback, and reflection processes integrating short-term and long-term memory.

continues until either the checker validates the output as correct or the iteration limit N is reached.

Theoretical optimality of iterative feedback mechanism. In the SAGE framework, the assistant repeatedly updates its policy through this checker feedback, enabling the outputs to be incrementally refined until either the result is validated or a specified iteration limit is reached. The assistant's utility R_A reflects task performance, and the checker's utility R_C depends on its feedback. The following theorem indicates that this iterative feedback mechanism leads to strategy stability in the sense of a Nash equilibrium (Fudenberg and Tirole 1991).

Theorem 0.1 (Theory for the multi-agent iterative feedback system)**.** *Let $\mathcal{U}, \mathcal{A}, \mathcal{C}$ denote the compact, convex strategy spaces of the user (U), assistant (A), and checker (C), respectively. Assume that the utility functions*

$$R_U : \mathcal{U} \times \mathcal{A} \times \mathcal{C} \to \mathbb{R}, \quad R_A : \mathcal{U} \times \mathcal{A} \times \mathcal{C} \to \mathbb{R}, \quad \text{and} \quad R_C : \mathcal{U} \times \mathcal{A} \times \mathcal{C} \to \mathbb{R}. \tag{2}$$

are continuous in each player's strategy. Then, by the Debreu-Glicksberg-Fan fixed-point theorem, there exists a Nash equilibrium

$$(s_U^*, s_A^*, s_C^*) \in \mathcal{U} \times \mathcal{A} \times \mathcal{C}. \tag{3}$$

Furthermore, suppose that the assistant's policy π_θ is updated via policy gradient methods and that the checker's strategy is refined through convex optimization. Then, the iterative update procedures yield sequences

$$\{\pi_\theta^{(k)}\}_{k\geq 0} \quad and \quad \{f^{(k)}\}_{k\geq 0}, \tag{4}$$

which converge to a stable strategy profile (π_θ^*, f^*), and has:

$$R_A(\pi_\theta^*, f^*) \geq R_A(\pi_\theta, f^*), \quad R_C(\pi_\theta^*, f^*) \geq R_C(\pi_\theta^*, f). \tag{5}$$

This result demonstrates that the iterative feedback mechanism enhances the model's strategy stability by converging to a Nash equilibrium in the three-player game. It provides a stronger justification for the three-agent system versus simpler alternatives (such as **two-agent systems**).

Evolutionary Goals and Directions Leveraging the feedback f_t^i obtained at each iteration t, the assistant A formulates new evolutionary objectives:

$$\mathcal{G}^{t+1} = (\mathcal{A}^{t+1}, \mathcal{D}^{t+1}), \tag{6}$$

$$\mathcal{D}^{t+1} = \arg\min_{\mathcal{D}_t \in \Delta} \sum_{i \in I_t} L_D\left(\mathcal{D}_t; f_t^i, \pi_\theta^t\right) \tag{7}$$

where $\mathcal{A}^{t+1}$ represents the updated memory optimization mechanisms, and $\mathcal{D}^{t+1} \in \Delta$ refers to the model's self-adjustments to make the RL algorithm converge. These evolutionary objectives guide the assistant in updating its policy π_θ for the subsequent iteration. The policy update is governed by the function ψ, which integrates the current policy π_θ^t with the new evolutionary objectives $\mathcal{G}^{t+1}$:

$$\begin{aligned}\theta^{t+1} &= \phi\big(\theta^t, \mathcal{G}^{t+1}\big) \\ &= \theta^t + \alpha \nabla_\theta \Big[\lambda_A L_A\big(\theta^t, \mathcal{A}^{t+1}\big) + \lambda_D L_D\big(\theta^t, \mathcal{D}^{t+1}\big)\Big]\end{aligned} \tag{8}$$

Here $L_A(\theta, \mathcal{A})$ and $L_D(\theta, \mathcal{D})$ are MSE loss functions corresponding to the memory-optimization and self-adjustment aspects, respectively, and $\lambda_A, \lambda_D \geq 0$ are weighting coefficients. The iterative policy refinement enables the assistant A to continuously adapt its strategies based on cumulative feedback and evolving task requirements, thereby improving its overall performance in dynamic environments.

Memory Management

The SAGE framework implements a dual-memory system, consisting of Short-Term Memory (STM) and Long-Term Memory (LTM), to manage task-relevant information and enhance the agent's reasoning and decision-making capabilities (see Figure 2 for a visual representation of this process).

Short-Term Memory (STM). STM is responsible for storing immediate, task-specific data with limited capacity. It updates rapidly with new observations ($\mathcal{O}_t$) and actions ($\mathbf{a}_t$), maintaining a recent trajectory history $\mathcal{T}_t = (\mathcal{O}_t, \mathbf{a}_t)$. This allows the agent to make real-time decisions and respond quickly to dynamic changes in the environment (Mnih et al. 2015).

Long-Term Memory (LTM). LTM retains critical information and self-reflections (r_t) over extended periods, enabling the agent to accumulate knowledge from past interactions and apply it to future tasks. Stored as $\mathcal{M}_L = \{r_t \mid t \in T\}$, this memory mechanism allows the agent to use prior experiences to improve task performance, particularly in complex environments that require long-span information (Graves et al. 2016).

By integrating STM and LTM, the SAGE framework allows the agent to balance immediate task demands with the ability to draw from accumulated knowledge, thereby enhancing its overall decision-making efficiency.

Reflection Figure 3 illustrates an example of the reflection mechanism applied to a HotpotQA task (Yang et al. 2018b). The reflection mechanism equips the assistant A with sparse reward signals, such as binary success/failure states, trajectory $\mathcal{T}_t$, and its stored memory $\mathcal{M}_L$. The assistant processes these inputs, deriving insights from past performance and storing self-reflections $\mathbf{r}_t$ for future decision-making. These self-reflections, richer than scalar rewards, enhance the assistant's learning capacity and are incorporated into long-term memory:

$$\mathbf{r}_t = \text{ref}(\mathbf{o}_{1:t}, \mathbf{R}_{1:t}), \tag{9}$$

where $\text{ref}(\cdot)$ denotes the reflection function based on the output sequence $\mathbf{o}_{1:t}$ and rewards $\mathbf{R}_{1:t}$. The derived reflection $\mathbf{r}_t$ is then added to $\mathcal{M}_L$:

$$\mathcal{M}_L \leftarrow \mathcal{M}_L \cup \{\mathbf{r}_t\}. \tag{10}$$

The process gradually enhances the agent's decision-making, allowing it to adapt effectively through accumulated experience.

MemorySyntax Building upon the reflection mechanism, the MemorySyntax method integrates the Ebbinghaus forgetting curve with linguistic principles to emulate human-like memory processes within the agent's memory management system. Let I_t denote the information received at time t, and let $R(I_t, \tau)$ represent its retention rate after a time interval τ. According to the Ebbinghaus forgetting curve, the retention rate is modeled as:

$$R(I_t, \tau) = e^{-\frac{\tau}{S}}, \tag{11}$$

where S is the strength of the information, reflecting its importance and complexity.

To enhance retention, MemorySyntax applies linguistic optimization to I_t, producing an optimized version I_t^* with increased strength $S^* > S$. The retention rate for I_t^* is defined as:

$$R(I_t^*, \tau) = \begin{cases} e^{-\frac{\tau}{S^*}}, & \text{if } I_t^* \in \mathcal{M}_S, \\ e^{-\frac{\tau}{S}}, & \text{if } I_t^* \in \mathcal{M}_L, \end{cases} \tag{12}$$

where $\mathcal{M}_S$ and $\mathcal{M}_L$ represent short-term memory and long-term memory, respectively.

The agent updates its memory state $\mathcal{M}_t$ based on the retention rate of I_t^* using predefined thresholds θ_1 and θ_2, with $\theta_1 > \theta_2$. The memory update rule is formalized as:

$$\mathcal{M}_{t+1} = \begin{cases} \mathcal{M}_t \cup \{I_t^*\}, & \text{if } R(I_t^*, \tau) \geq \theta_1, \\ \mathcal{M}_t \setminus \{I_t^*\}, & \text{if } R(I_t^*, \tau) < \theta_2, \\ \mathcal{M}_t, & \text{otherwise.} \end{cases} \tag{13}$$

User: Answer the question based on the given passages. Only give me the answer and do not output any other words. The following are given passages:
The "trusty system" (sometimes incorrectly called "trustee system") was a penitentiary system of discipline and security enforced in parts of the United States until the 1980s ......
Answer the question based on the given passages. Only give me the answer and do not output any other words.
Question: Which case was brought to court first Miller v. California or Gates v. Collier ?
Answer:

Assistant: Gates v. Collier

Checker:
Incorrect answer: Neglecting specific years: Assistant did not accurately confirm the specific years of each case, leading to an incorrect judgment of the chronological order.
Insufficient understanding of the question: When answering the question, Assistant may not have fully understood or carefully analyzed the provided information, resulting in a misjudgment.

Reflection:
When answering the question about which case, "Miller v. California" or "Gates v. Collier," was brought up first, I mistakenly thought "Gates v. Collier" preceded "Miller v. California." In reality, "Miller v. California" came first. This indicates that I made an error in judging the chronological order of events.
Improvement measures: Carefully verify information: When answering questions involving time sequences, it is essential to carefully check the specific timing of each event to ensure accuracy.
Timeline tools: When dealing with multiple events or cases, use timeline tools to help clarify the chronological order and avoid oversights.
Review mechanism: Before submitting the final answer, establish a self-review mechanism to recheck all relevant information and details to ensure the accuracy of the response.
Conclusion: This mistake serves as a reminder for me to be more careful and cautious when handling questions related to time and sequence. Through reflection and improvement, I hope to provide more accurate and reliable answers in the future and prevent similar errors from occurring.

Assistant with SAGE response:
Miller v. California

Figure 3: The illustration of an example HotpotQA with SAGE.

This update rule operates under the following conditions:

- **Retention in Short-Term Memory ($\mathcal{M}_S$):** If $R(I_t^*, \tau) \geq \theta_1$, the information I_t^* is retained in short-term memory $\mathcal{M}_S$.
- **Discarding Information:** If $R(I_t^*, \tau) < \theta_2$, the information I_t^* is considered insignificant and is discarded.
- **Transfer to Long-Term Memory ($\mathcal{M}_L$):** If $\theta_2 \leq R(I_t^*, \tau) < \theta_1$, the information I_t^* is transferred to long-term memory $\mathcal{M}_L$.

By simulating human memory dynamics, MemorySyntax enables the agent to prioritize essential information, retain critical data in short-term memory $\mathcal{M}_S$, store important but less frequently used information in long-term memory $\mathcal{M}_L$, and discard irrelevant data. The mechanism addresses memory capacity limitations and enhances the agent's ability to perform complex tasks requiring efficient memory management.

Experiment

To demonstrate the capabilities and performance of SAGE in coordinating autonomous agent groups to work together on tasks, we conduct extensive quantitative experiments on benchmark tasks. We use a public benchmark, AgentBench (Liu et al. 2023), which is a multidimensional evolutionary benchmark from which we select six tasks. These tasks test the reasoning and decision-making abilities of LLMs acting as agents in multi-turn open-ended generation settings. To further assess the agents' long-context understanding, we select four widely adopted tasks related to long-text processing. These tasks reflect the agents' programming abilities(LCC(Guo et al. 2023), RepoBench-P (Liu, Xu, and McAuley 2023)) and reasoning abilities(HotpotQA (Yang et al. 2018b), TriviaQA (Joshi et al. 2017b)).

Evaluation on AgentBench

Task Description AgentBench includes scenarios from CODE (Knowledge Graph, OS, DB), GAME (ALFWorld) (Shridhar et al. 2021), and WEB (WebShop (Yao et al. 2023a), Mind2Web (Deng et al. 2023)). For more details for the datasets and benchmarks, see Appendix **??**.

Baselines We evaluate commercial models GPT-3.5 (Brown et al. 2020) and GPT-4 (OpenAI et al. 2024), and open-source models Llama2 (Touvron et al. 2023), Codellama (Rozière et al. 2024), Qwen (Bai et al. 2023), and ChatGLM2 (GLM et al. 2024). Dialogue history exceeding the model length limit is truncated, using greedy decoding.

Results As shown in Table 1, our method significantly improves model performance on AgentBench, especially for smaller models. GPT-3.5 and GPT-4, despite already high scores, also show notable improvements with SAGE, up to 2.26x in the Database task. Llama2-7b is notably enhanced, showing the method's impact on weaker models. CodeLlama-7b and Qwen-1.8B also see substantial gains. Qwen-1.8B, after applying our method, performs close to GPT-3.5, highlighting its potential as a general agent. Llama2, previously error-prone, shows a significant reduction in basic errors through feedback and memory optimization, proving that our method not only enhances agent capabilities but also reduces fundamental errors in complex tasks.

Table 1: Baseline and SAGE Framework Performance on AgentBench

LLM Type	Model	VER		OS		DB		KG		ALF		WS		M2W	
		Base	SAGE	Base	SAGE	Base	SAGE	Base	SAGE	Base	SAGE	Base	SAGE	Base	SAGE
API	GPT-4	42.4	**49.7**	32.0	**39.8**	57.4	**63.1**	78.0	**82.0**	67.1	**67.8**	27.0	32.0	27.0	**32.0**
	GPT-3.5	31.6	**38.3**	15.7	**35.6**	25.9	**37.6**	17.0	**23.0**	64.1	**72.1**	16.0	**28.0**	16.0	**28.0**
OSS	Llama2-7B Chat	0.0	**8.4**	0.0	**10.2**	0.0	**25.0**	0.0	**5.0**	4.4	**10.4**	0.0	**15.0**	0.0	**15.0**
	CodeLlama-7B Instruct	5.7	**18.4**	2.6	**19.2**	0.0	**27.0**	0.0	**12.5**	16.3	**40.2**	0.0	**15.0**	15.0	15.0
	Qwen1.8B Chat	2.7	**18.7**	1.4	**15.1**	6.8	**45.3**	0.0	**10.5**	6.6	**11.4**	0.6	**13.6**	13.6	13.6
	Qwen-7B Chat	5.6	**22.2**	4.8	**18.0**	0.0	**48.0**	34.0	**38.5**	0.0	**13.6**	0.0	**15.0**	15.0	15.0
	ChatGLM2-6B v1.1	0.0	**15.2**	0.0	**16.3**	0.0	**17.0**	0.0	**5.0**	0.3	**10.3**	4.9	**14.9**	14.9	14.9

Table 2: Evaluation of SAGE and Baseline Models on Three Different Tasks

Agent	Task Completion Time (min)	Answer Accuracy (QA) (%)	Dialog Coherence (%)	Step Completion Accuracy (%)
GPT-3.5 (Baseline)	Long-form QA (HotpotQA)	54.1%	48.5%	62.7%
GPT-4 (Baseline)	Long-form QA (HotpotQA)	61.2%	53.8%	68.2%
Llama2-7b (Baseline)	Multi-turn Dialog (MultiWOZ)	55.9%	50.1%	64.8%
Codellama-13b (Baseline)	Multi-turn Dialog (MultiWOZ)	58.4%	52.3%	66.7%
Mistral-7b (Baseline)	Sequential Task (ALFWorld)	56.5%	51.5%	65.1%
SAGE-GPT-3.5	Long-form QA (HotpotQA)	**74.9% (+20.8%)**	**68.3% (+19.8%)**	**80.6% (+17.9%)**
SAGE-GPT-4	Long-form QA (HotpotQA)	**78.4% (+17.2%)**	**73.4% (+19.6%)**	**83.9% (+15.7%)**
SAGE-Llama2-7b	Multi-turn Dialog (MultiWOZ)	**72.2% (+16.1%)**	**67.9% (+17.8%)**	**78.5% (+13.7%)**
SAGE-Codellama-13b	Multi-turn Dialog (MultiWOZ)	**74.7% (+16.3%)**	**71.2% (+18.9%)**	**81.2% (+14.5%)**
SAGE-Mistral-7b	Sequential Task (ALFWorld)	**73.8% (+17.3%)**	**70.5% (+19.0%)**	**79.9% (+14.8%)**

Complex Problem-Solving Tasks Evaluation

We evaluated SAGE against baseline models on three tasks: long-form QA (Akash et al. 2023), multi-turn dialog (Cui et al. 2020), and sequential task completion (Stephens, Cho, and Ballard 2012). As shown in Table 2, SAGE outperforms all baselines with significant gains, such as a 20.8% increase in answer accuracy for GPT-3.5 on HotpotQA (Yang et al. 2018b) and a 17.3% improvement in task completion for Mistral-7b on ALFWorld (Shridhar et al. 2021). Across all tasks, SAGE notably enhances answer accuracy, dialog coherence, and step completion.

Evaluation of Long-Context Tasks

We evaluated the agent's code generation and reasoning on four long-text tasks: **LCC Dataset** (Mohler et al. 2016) focuses on predicting the next line of code from a few initial lines, with Precision, Recall, and F1 as metrics. **RepoBench-P** (Liu, Xu, and McAuley 2024) tests retrieval of relevant code snippets from cross-file and within-file contexts to predict the next line, also evaluated with Precision, Recall, and F1. **HotPotQA** (Yang et al. 2018a), a Wikipedia-based dataset with 113k question-answer pairs, challenges the agent to reason across multiple documents, evaluated by answer F1. **TriviaQA** (Joshi et al. 2017a) is a reading comprehension dataset with question-answer pairs and evidence paragraphs (filtered to over 1,000 words), also using answer F1 for evaluation.

We compared two self-refinement methods: **Beam Search** (Kool, van Hoof, and Welling 2019), which integrates self-assessment through stochastic beam search, and **Reflexion** (Shinn et al. 2023), which uses past trial experience in a verbal form.

Evaluation Results:

Code Completion Task: On the LCC dataset (Table 3), SAGE shows a slight improvement in F1 score (79.29) compared to Beam Search and Reflexion. Its memory mechanisms help refine code predictions, but the performance difference is not substantial in simpler tasks like code completion.

Reasoning Tasks: SAGE significantly outperforms Reflexion and Beam Search on HotPotQA and TriviaQA, with

Table 3: Comparison of Performance Across Different Methods

Models	LCC			RepoBench-P			HotpotQA	TriviaQA
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**F1**	**F1**
Reflexion	77.72	81.00	79.28	78.73	81.86	80.25	11.26	11.23
Beam search	78.98	79.32	79.12	78.75	81.02	79.87	10.26	12.13
SAGE	78.76	79.88	79.29	79.27	83.28	81.22	22.06	22.76

Table 4: Evaluation of different RAG Agents on Different Tasks and Datasets

Agent	**Accuracy (QA) (%)**	**Latency (ms)**	**Memory Usage (MB)**
Task 1: Multi-Document QA (HotpotQA)			
RAG (BM25)	60.8	121	613
RAG (DPR)	66.3	129	542
RAG (OpenAI Retrieval)	67.4	108	494
TART	63.2	144	477
FiD (Fusion-in-Decoder)	70.1	153	456
ChatGPT-4 - Sage	**74.8 (+4.7)**	128	**231 (-50%)**
Task 2: Document Retrieval for Contextual Answering (Natural Questions)			
RAG (BM25)	59.9	125	605
RAG (DPR)	65.5	131	561
RAG (OpenAI Retrieval)	66.8	113	484
TART	62.4	146	455
FiD (Fusion-in-Decoder)	69.8	156	443
ChatGPT-4 - Sage	**73.6 (+3.8)**	131	**227 (-49%)**
Task 3: Open-Domain QA with Multiple Contexts (TriviaQA)			
RAG (BM25)	62.1	124	615
RAG (DPR)	67.8	129	530
RAG (OpenAI Retrieval)	68.9	117	494
TART	64.7	148	462
FiD (Fusion-in-Decoder)	71.9	155	456
ChatGPT-4 - Sage	**75.5 (+3.6)**	134	**243 (-47%)**

F1 scores of 22.06 and 22.76 (Table 3). SAGE's ability to effectively integrate multi-document information through reflection leads to better reasoning accuracy, while Reflexion and Beam Search face challenges in handling complex reasoning tasks.

Evaluation of RAG Agents

Table 4 compares classical lexical retrieval (RAG with BM25) (Robertson and Zaragoza 2009), dense passage retrieval (RAG with DPR) (Reichman and Heck 2024), a commercial retrieval solution (RAG with OpenAI Retrieval) (OpenAI 2023), the TART (Eisenschlos et al. 2022) sequence-to-sequence retrieval model, and the FiD (Fusion-in-Decoder) method (Izacard and Grave 2021), all tested on multi-document and open-domain QA tasks (HotpotQA, Natural Questions (Kwiatkowski et al. 2019), and TriviaQA). RAG with BM25 relies on term-based matching, while RAG with DPR uses learned dense embeddings. TART adopts a transformer-based approach to produce relevant contexts, and FiD fuses multiple retrieved passages through an encoder-decoder design. In contrast, ChatGPT-4 (SAGE) employs a structured reasoning workflow for retrieval and generation, which leads to steady accuracy improvements of 3.6% to 4.7% and cuts memory consumption nearly 50% on some tasks, all without increasing latency.

Error analysis

As shown in Figure 4, the SAGE framework significantly enhances agent performance across tasks, especially in the WS task for AgentBench, due to its iterative feedback mechanism, which refines outputs through continuous assistant-checker interaction. In OS and DB tasks, Context Limit Exceeded and invalid format errors are nearly eliminated, with a notable reduction in invalid action errors, attributed to the reflection mechanism that helps the assistant learn and reduce logical mistakes.

Ablation Study

We conducted ablation experiments on Qwen-1.8B and CodeLlama-7B to evaluate memory optimization (Table 5). Without memory optimization, both models perform weakly, especially Qwen-1.8B, which improves from 6.8 to 48.0 in

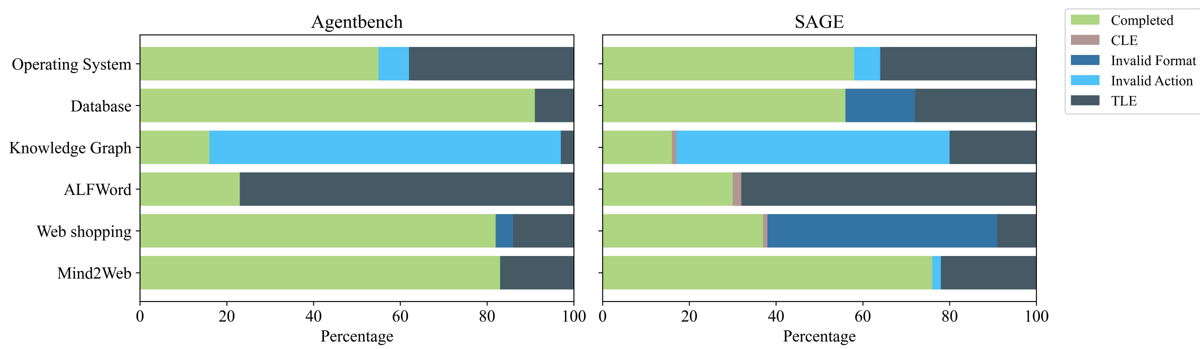

Figure 4: Execution results across six tasks (CLE: Context Limit Exceeded, TLE: Task Limit Exceeded). Task limits are the main cause of incomplete tasks, highlighting LLM agents' limitations under time constraints.

KG and from 0.0 to 10.5 in ALF after optimization. Similarly, CodeLlama-7B shows substantial gains, particularly in DB (2.7 to 41.3) and WS (14.3 to 58.7). Overall, CodeLlama-7B performs better than Qwen-1.8B, highlighting the stronger adaptability of models with more parameters in handling complex tasks.

Table 5: Ablation study for memory optimization on the task of AgentBench

Models	OS	DB	KG	ALF	WS	M2W
Qwen-1.8B (w/o memo)	10.4	22.6	6.8	0.0	26.6	5.0
Qwen-1.8B (w memo)	**18.7**	**28.3**	**45.3**	**10.5**	**31.4**	**25.1**
Codellama-7B (w/o memo)	9.7	2.7	0.0	0.0	14.3	5.0
Codellama-7B (w memo)	**23.4**	**41.3**	**48.0**	**12.5**	**58.7**	**15.0**

Conclusion

In this paper, we propose the SAGE framework, which enhances agents' self-adjustment and memory management in complex tasks through reflective mechanisms and memory optimization. Experimental results show significant performance improvements across benchmarks, especially in smaller models. In the AgentBench test, SAGE boosts the performance of strong baselines like GPT-3.5 and GPT-4, while also significantly improving open-source models. It effectively reduces basic errors and logical mistakes, particularly enabling smaller models to handle complex tasks.

References

Akash, P. S.; Roy, K. K.; Popa, L.; and Chang, K. C.-C. 2023. Long-form Question Answering: An Iterative Planning-Retrieval-Generation Approach. arXiv:2311.09383.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. arXiv:2309.16609.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Cui, L.; Wu, Y.; Liu, S.; Zhang, Y.; and Zhou, M. 2020. MuTual: A Dataset for Multi-Turn Dialogue Reasoning. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1406–1416. Online: Association for Computational Linguistics.

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. arXiv:2306.06070.

Ebbinghaus, H. 1885. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot.

Eisenschlos, J.; Zemlyanskiy, Y.; Phang, J.; Dhingra, B.; and Cohen, W. W. 2022. TART: Retrieval-Augmented Generation for Task-Oriented Dialog. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1866–1881.

Etezadi, R.; and Shamsfard, M. 2023. The state of the art in open domain complex question answering: a survey. *Applied Intelligence*, 53(4): 4124–4144.

Fudenberg, D.; and Tirole, J. 1991. *Game Theory*. Cambridge, MA: MIT Press.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Rojas, D.; Feng, G.; Zhao, H.; Lai, H.; Yu, H.; Wang, H.; Sun, J.; Zhang, J.; Cheng, J.; Gui, J.; Tang, J.; Zhang, J.; Li, J.; Zhao, L.; Wu, L.; Zhong, L.; Liu, M.; Huang, M.; Zhang, P.; Zheng, Q.; Lu, R.; Duan, S.; Zhang, S.; Cao, S.; Yang, S.; Tam, W. L.; Zhao, W.; Liu, X.; Xia, X.; Zhang, X.; Gu, X.; Lv, X.; Liu, X.; Liu, X.; Yang, X.; Song, X.; Zhang, X.; An, Y.; Xu, Y.; Niu, Y.; Yang, Y.; Li, Y.; Bai, Y.; Dong, Y.; Qi, Z.; Wang, Z.; Yang, Z.; Du, Z.; Hou, Z.; and Wang, Z. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv:2406.12793.

Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626): 471–476.

Guo, D.; Xu, C.; Duan, N.; Yin, J.; and McAuley, J. 2023. LongCoder: A Long-Range Pre-trained Language Model for Code Completion. arXiv:2306.14893.

He, Y.; Wang, J.; Li, K.; Wang, Y.; Sun, L.; Yin, J.; Zhang, M.; and Wang, X. 2024. Enhancing Intent Understanding for Ambiguous Prompt: A Human-Machine Co-Adaption Strategy. *Available at SSRN 5119629*.

He, Y.; Wang, J.; Li, K.; Wang, Y.; Sun, L.; Yin, J.; Zhang, M.; and Wang, X. 2025. Enhancing Intent Understanding for Ambiguous Prompts through Human-Machine Co-Adaptation. *arXiv preprint arXiv:2501.15167*.

Huang, J.; Gu, S. S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2022. Large Language Models Can Self-Improve. arXiv:2210.11610.

Izacard, G.; and Grave, E. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*, 874–880.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017a. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017b. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. arXiv:1705.03551.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. arXiv:1903.06059.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; 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.

Li, Z.; He, Y.; He, L.; Wang, J.; Shi, T.; Lei, B.; Li, Y.; and Chen, Q. 2024. FALCON: Feedback-driven Adaptive Long/short-term memory reinforced Coding Optimization system. *arXiv preprint arXiv:2410.21349*.

Liang, X.; Tao, M.; Xia, Y.; Shi, T.; Wang, J.; and Yang, J. 2024. Cmat: A multi-agent collaboration tuning framework for enhancing small language models. *arXiv preprint arXiv:2404.01663*.

Liu, T.; Xu, C.; and McAuley, J. 2023. RepoBench: Benchmarking Repository-Level Code Auto-Completion Systems. arXiv:2306.03091.

Liu, T.; Xu, C.; and McAuley, J. 2024. RepoBench: Benchmarking Repository-Level Code Auto-Completion Systems. In *The Twelfth International Conference on Learning Representations*.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; Zhang, S.; Deng, X.; Zeng, A.; Du, Z.; Zhang, C.; Shen, S.; Zhang, T.; Su, Y.; Sun, H.; Huang, M.; Dong, Y.; and Tang, J. 2023. AgentBench: Evaluating LLMs as Agents. *arXiv preprint arXiv: 2308.03688*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Mohler, M.; Brunson, M.; Rink, B.; and Tomlinson, M. 2016. Introducing the LCC Metaphor Datasets. In Calzolari, N.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 4221–4227. Portorož, Slovenia: European Language Resources Association (ELRA).

Nakajima, Y. 2024. BabyAGI: An AI Agent That Can Achieve Goals and Execute Tasks.

OpenAI. 2023. OpenAI Cookbook. https://github.com/openai/openai-cookbook. Accessed: 2025-03-27.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.;

Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

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.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; and Yang, B. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. In *Proc. VLDB Endow.*, 2363–2377.

Qiu, X.; Li, X.; Pang, R.; Pan, Z.; Wu, X.; Yang, L.; Hu, J.; Shu, Y.; Lu, X.; Yang, C.; Guo, C.; Zhou, A.; Jensen, C. S.; and Yang, B. 2025a. EasyTime: Time Series Forecasting Made Easy. In *ICDE*.

Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2025b. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*, 1185–1196.

Rae, J. W.; Potapenko, A.; Jayakumar, S. M.; and Lillicrap, T. P. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Reichman, B.; and Heck, L. 2024. Dense Passage Retrieval: Is it Retrieving? arXiv:2402.11035.

Robertson, S.; and Zaragoza, H. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; Rapin, J.; Kozhevnikov, A.; Evtimov, I.; Bitton, J.; Bhatt, M.; Ferrer, C. C.; Grattafiori, A.; Xiong, W.; Défossez, A.; Copet, J.; Azhar, F.; Touvron, H.; Martin, L.; Usunier, N.; Scialom, T.; and Synnaeve, G. 2024. Code Llama: Open Foundation Models for Code. arXiv:2308.12950.

Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366.

Shridhar, M.; Yuan, X.; Côté, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Stephens, K. K.; Cho, J. K.; and Ballard, D. I. 2012. Simultaneity, sequentiality, and speed: Organizational messages about multiple-task completion. *Human Communication Research*, 38(1): 23–47.

Tian, Y.; Peng, B.; Song, L.; Jin, L.; Yu, D.; Mi, H.; and Yu, D. 2024. Toward Self-Improvement of LLMs via Imagination, Searching, and Criticizing. arXiv:2404.12253.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wada, T.; Iwata, T.; and Matsumoto, Y. 2019. Unsupervised multilingual word embedding with limited resources using neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3113–3124.

Yang, H.; Yue, S.; and He, Y. 2023. Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions. arXiv:2306.02224.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018a. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380. Brussels, Belgium: Association for Computational Linguistics.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018b. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. arXiv:1809.09600.

Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2023a. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. arXiv:2207.01206.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023b. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

Zhou, Z.; Zhang, J.; Zhang, J.; He, Y.; Wang, B.; Shi, T.; and Khamis, A. 2024. Human-centric Reward Optimization for Reinforcement Learning-based Automated Driving using Large Language Models. *arXiv preprint arXiv:2405.04135*.