Lifelong Learning of Large Language Model based Agents: A Roadmap

Junhao Zheng*, Chengming Shi*, Xidi Cai*, Qiuke Li*, Duzhen Zhang, Chenxing Li, Dong Yu, *Fellow, IEEE*, Qianli Ma†, *Member, IEEE*

Abstract—Lifelong learning, also known as continual or incremental learning, is a crucial component for advancing Artificial General Intelligence (AGI) by enabling systems to continuously adapt in dynamic environments. While large language models (LLMs) have demonstrated impressive capabilities in natural language processing, existing LLM agents are typically designed for static systems and lack the ability to adapt over time in response to new challenges. This survey is the first to systematically summarize the potential techniques for incorporating lifelong learning into LLM-based agents. We categorize the core components of these agents into three modules: the perception module for multimodal input integration, the memory module for storing and retrieving evolving knowledge, and the action module for grounded interactions with the dynamic environment. We highlight how these pillars collectively enable continuous adaptation, mitigate catastrophic forgetting, and improve long-term performance. This survey provides a roadmap for researchers and practitioners working to develop lifelong learning capabilities in LLM agents, offering insights into emerging trends, evaluation metrics, and application scenarios. Relevant literature and resources are available at https://github.com/qianlima-lab/awesome-lifelong-llm-agent.

Index Terms—Lifelong Learning, Continual Learning, Incremental Learning, Large Language Model, AI Agent, AGI

1 Introduction

"Intelligence is the ability to adapt to change."

—Stephen Hawking

Lifelong learning [1], [2], also known as continual or incremental learning [3], [4], has become a key focus in the development of intelligent systems. As shown in Figure 1, lifelong learning has attracted increasing research attention in recent years. It plays a crucial role in allowing these systems to continuously adapt and improve over time. As noted by Legg et al. [5], human intelligence is fundamentally about *fast adaptation to a wide range of environments*, highlighting the need for AI systems to exhibit this same level of adaptability. Lifelong learning refers to a system's ability to acquire, integrate, and retain knowledge while avoiding the forgetting of previously learned information. This ability is particularly important for systems that operate in dynamic, complex environments, where new tasks and challenges frequently arise. In contrast to traditional machine learning models, which are typically trained on fixed datasets and optimized for specific tasks, lifelong learning systems are designed to evolve. They accumulate new knowledge and continuously refine their capabilities as they encounter new situations.

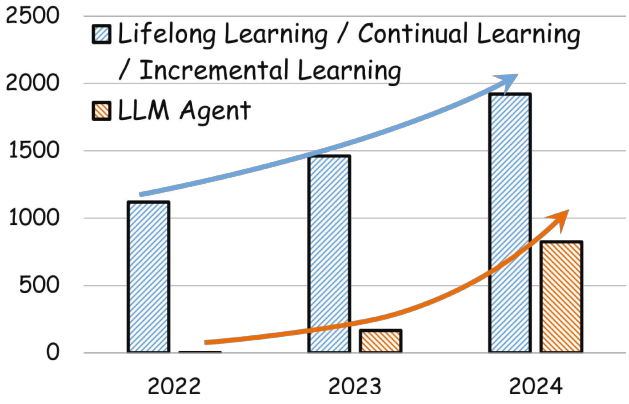
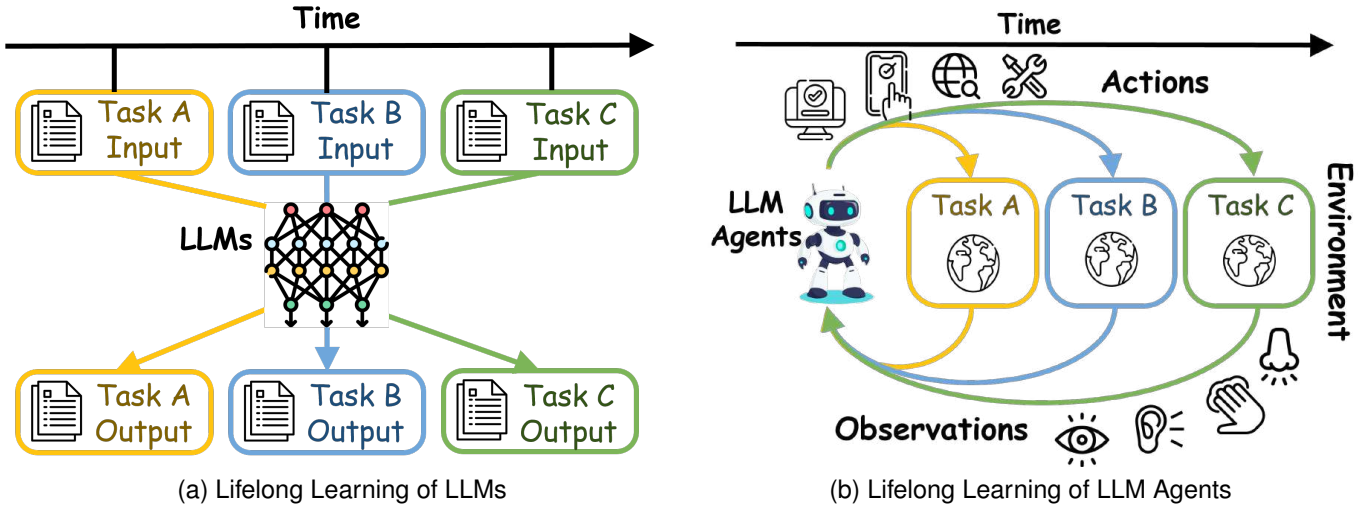


Fig. 1. Number of publications on *lifelong learning* and *LLM Agents* (from Google Scholar). The publications have grown rapidly in recent three years.

Despite its potential, there remains a significant gap between advancements in AI and the practical application of lifelong learning. While humans can naturally integrate new knowledge while retaining the old, current AI systems face two main challenges in lifelong learning: *catastrophic forgetting* [6] and *loss of plasticity* [7], [8]. These challenges form the *stability-plasticity dilemma* [9]. On one hand, catastrophic forgetting occurs when systems forget previously learned information as they learn new tasks, which is particularly problematic when the environment changes. On the other hand, the loss of plasticity refers to the system's inability to adapt to new tasks or environments. These two issues represent opposing ends of the learning spectrum: static

Version: v1 (major update on January 13, 2025)
†*Corresponding author: Qianli Ma.*
**The first four authors contributed equally to this research.*
Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Qianli Ma are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (E-mail: junhaozheng47@outlook.com; cscmshi@mail.scut.edu.cn; xidicai067@gmail.com; lqk867543@gmail.com; qianlima@scut.edu.cn).
Duzhen Zhang is with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE (E-mail: bladedancer957@gmail.com).
Chenxing Li is with the Tencent, AI Lab, Beijing, China (E-mail: chenxingli@tencent.com).
Dong Yu is with the Tencent, AI Lab, Bellevue, WA 98004 USA (E-mail: dyu@global.tencent.com).



(a) Lifelong Learning of LLMs (b) Lifelong Learning of LLM Agents

Fig. 2. Comparison of lifelong learning between LLMs and LLM Agents. (a) Traditional lifelong learning paradigm of LLMs, where LLMs are viewed as static black-box systems *without feedback from the environment*; (b) the novel lifelong learning paradigm of LLM agents focused on in this survey, where agents *interact with ever-changing environments*. Please refer to Figure 4 for an illustration.

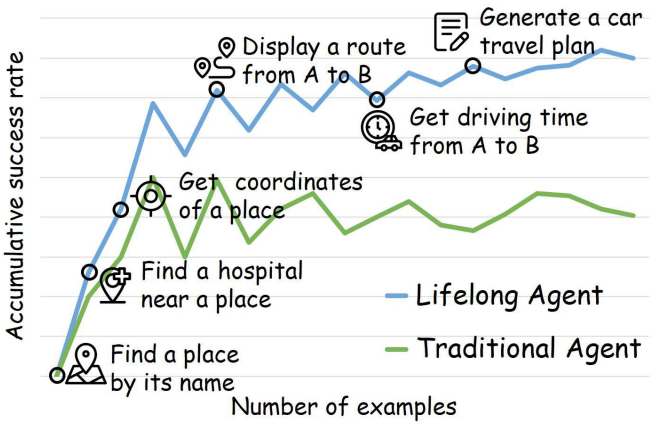


Fig. 3. A lifelong LLM agent can adapt to its environment and achieve behavioral evolution through interaction (adapted from AWM [10]).

systems avoid forgetting but lack the ability to adapt, while systems focused on adaptation are at risk of forgetting past knowledge. Overcoming this dilemma is key to advancing artificial intelligence and is a foundational challenge on the path to Artificial General Intelligence (AGI) [5].

1.1 Motivation for Building Lifelong Learning LLM Agents

The recent advancements in large language models (LLMs) [11], [12] have significantly transformed the field of natural language processing. Models like GPT-4 [12] are designed to process and generate human-like text by learning from vast amounts of textual data. They excel in tasks such as text generation, machine translation, and question answering due to their ability to understand complex language patterns. However, traditional LLMs [11], [12] are static after training, meaning they cannot adapt to new tasks or environments once deployed. Their knowledge remains fixed, and they struggle to integrate new information without retraining, limiting their applicability in dynamic real-world scenarios.

In contrast, LLM agents represent a more advanced form of artificial intelligence. Unlike standard LLMs, which process input text and generate output based on prior training, LLM agents [13], [14] are autonomous entities capable of interacting with their environment. These agents can perceive multimodal data (e.g., text, images, sensory data), store this information in memory, and take actions to influence or respond to their surroundings [15]–[17]. Designed to continuously adapt to new contexts, LLM agents learn from their interactions and experiences, improving their decision-making capabilities over time. Illustrations are provided in Figure 2 and Figure 3.

The motivation for incorporating lifelong learning into LLM agents arises from the need to develop intelligent systems that can not only adapt to new tasks but also retain and apply prior knowledge across a wide range of dynamic environments, aligning with Legg et al.'s [5] view of intelligence as *fast adaptation to a wide range of environments*. Currently, existing LLM agents are typically developed as static systems, limiting their ability to evolve in response to new challenges. Moreover, most lifelong learning research on LLMs [1], [4] focuses on handling ever-changing data distributions without interacting with an environment. For example, continual fine-tuning of LLMs to adapt to instructions from specific domains [1]. However, these approaches still treat LLMs as static black-box systems and do not address the practical need for LLMs to learn interactively within real-world environments. Figure 2 compares the traditional lifelong learning paradigm with the novel paradigm of LLM agents that interact with dynamic environments, as discussed in this survey.

In real-world applications, LLM agents are expected to adapt to diverse environments such as gaming, web browsing, shopping, household tasks, and operating systems without the need to design separate agents for each new context. By incorporating lifelong learning capabilities, these agents can overcome such limitations. They continuously learn and store knowledge from multiple modali-

Fig. 4. Illustration of lifelong learning in large language model-based agents. In real-world applications, LLM agents are expected to adapt to various environments such as gaming, web browsing, shopping, household tasks, and operating systems without the need to design environment-specific agents for every new environment.

ties (e.g., visual, textual, sensory data), enabling real-time adaptation and decision-making as environments change [18]–[21]. Integrating lifelong learning into LLM agents unlocks their full potential for dynamic real-world applications [22], [23]. Consequently, these agents can evolve continuously, acquire new knowledge, and preserve critical information, enhancing their adaptability and versatility. This ongoing learning process is essential for environments where new challenges regularly emerge, such as in autonomous robotics, interactive assistants, and adaptive decision-support systems [14]. An illustration of a lifelong learning LLM agent is provided in Figure 4.

1.2 Scope of the Survey

This survey provides a comprehensive overview of the key concepts, techniques, and challenges involved in developing lifelong learning systems for LLM-based agents. As the first survey to systematically summarize potential techniques for lifelong learning in LLM agents, it addresses the following research questions (RQs):

- **RQ1**: What are the core concepts, development processes, and fundamental architectures of LLM agents designed for lifelong learning? (Section 3)
- **RQ2**: How can LLM agents continuously perceive and process single-modal and multi-modal data to adapt to new environments and tasks? (Sections 4, 5)
- **RQ3**: What strategies can mitigate catastrophic forgetting and enable the retention of previously learned knowledge? (Sections 6, 7, 8, 9)
- **RQ4**: How can LLM agents perform various actions, such as grounding, retrieval, and reasoning, in dynamic environments? (Sections 10, 11, 12)
- **RQ5**: What evaluation metrics and benchmarks are used to assess the performance of lifelong learning in LLM agents? (Section 13)
- **RQ6**: What are the real-world applications and use cases of lifelong learning LLM agents, and how do they benefit from continuous adaptation? (Section 14)
- **RQ7**: What are the key challenges, limitations, and open problems in the development of lifelong learning for LLM-based agents? (Section 15)

By addressing these research questions, this survey serves as a step-by-step guide for understanding the design, challenges, and applications of lifelong learning in LLM agents. It reviews state-of-the-art techniques and highlights emerging trends and future directions for research in this rapidly evolving field.

1.3 Contribution of the Survey

To the best of our knowledge, this is the *first* survey that systematically reviews the latest advancements at the intersection of lifelong learning and LLM agents. The main contributions of this survey are as follows:

- **Foundational Overview:** Provides a thorough overview of the foundational concepts and architectures essential for implementing lifelong learning in LLM agents.
- **In-Depth Component Analysis:** Examines key components, including perception, memory, and action modules, that enable adaptive behavior in LLM agents.
- **Comprehensive Discussion:** Discusses real-world applications, evaluation metrics, benchmarks, as well as key challenges and future research directions in the domain of lifelong learning for LLM agents.

TABLE 1
Summary of the connections and differences between our work and existing surveys. "Scope": *LLMs* means large language models, *LL* means lifelong learning, *NLP* means natural language processing. A ✓ symbol indicates that a survey explicitly addresses a given domain.

Survey	Contribution	Scope			
		LLMs	LL	NLP	Agents
Zheng et al. [1]	A survey on lifelong learning methods for LLMs.	✓	✓	✓	
Wu et al. [24]	A survey on continual learning stage for LLMs.	✓	✓		
Shi et al. [25]	A survey on LLMs within the context of continual learning.	✓	✓		
Ke and Liu [26]	A survey on continual learning of natural language processing tasks.		✓	✓	
Zhou et al. [27]	A survey on continual Learning with pre-trained models.	✓	✓		
Wang et al. [4]	A survey on settings, foundations, methods and applications of continual learning.		✓		
Biesialska et al. [28]	A survey on continual lifelong learning in natural language processing.		✓	✓	
Parisi et al. [29]	A review on continual lifelong learning with neural networks.		✓		
Wang et al. [13]	A survey on LLM based autonomous agents.	✓			✓
Xi et al. [14]	A survey on LLM based agents.	✓			✓
Li et al. [30]	A survey on personal large language model agents.	✓			✓
Cheng et al. [31]	An overview of LLM based intelligent agents within single-agent and multi-agent systems.	✓			✓
Li et al. [32]	A survey on LLM based multi-agent systems.	✓			✓
Zhang et al. [33]	A survey on LLM-brained GUI agents.	✓			✓
Our Work	**A survey on lifelong learning of LLM based agents.**	✓	✓	✓	✓

1.4 Survey Structure

This survey is organized as follows. **Section 2** reviews related surveys and literature on LLM agents and lifelong learning. **Section 3** introduces the foundational concepts, development processes, and overall architecture of LLM agents designed for lifelong learning. **Sections 4 and 5** discuss the design of lifelong learning LLM agents from a perceptual perspective, focusing on single-modal and multi-modal approaches, respectively. **Sections 6, 7, 8, and 9** examine the design of LLM agents from a memory perspective, covering working memory, episodic memory, semantic memory, and parametric memory. **Sections 10, 11, and 12** explore the design of LLM agents from the action perspective, including grounding actions, retrieval actions, and reasoning actions. **Section 13** presents evaluation metrics and benchmarks for assessing lifelong learning in LLM agents. **Section 14** delves into real-world applications and use cases of lifelong learning LLM agents. **Section 15** offers practical insights and outlines future research directions. Finally, **Section 16** concludes the survey.

2 Related Work

2.1 Surveys on Lifelong Learning

In recent years, the rapid growth of lifelong learning has received a great deal of academic attention. In order to summarize the research in this area, researchers have written a large number of surveys. In this subsection, we will briefly review these surveys and summarize their main contributions in order to better highlight our research, as illustrated in Table 1.

Some surveys involve presenting research advances in lifelong learning in the field of natural language processing. For instance, Zheng et al. [1] systematically summarize lifelong learning methods for LLMs for the first time by considering them from the perspective of 12 scenarios, which are classified into two categories: internal and external knowledge. In the section on internal knowledge, this survey presents scenarios about continual finetuning in natural language processing. Ke and Liu [26] first introduce the settings and learning modes of continual learning and summarize the natural language processing problems in continual learning. Then, they propose that existing continual learning techniques are mainly used to solve the two challenges of catastrophic forgetting and knowledge transfer. On this basis, this survey analyzes approaches for catastrophic forgetting prevention and knowledge transfer. Besides, Biesialska et al. [28] classify continual learning methods into three main categories: rehearsal, regularization, and architectural as well as a few hybrid categories. This survey also explores the application of continual learning to several tasks in natural language processing, including question answering, sentiment analysis, and text classification.

Other surveys present progress in lifelong learning from a variety of perspectives. For instance, Wu et al. [24] describe the different training stages of LLMs, including continual pre-training, continual instruction tuning, and continual alignment. In addition, this survey presents benchmarks and evaluations on continual learning. Shi et al. [25] present the general picture of continually learning LLMs from vertical continuity and horizontal continuity. In addition, this survey introduces the evaluation protocols and datasets for continually learning large language models. Starting from the motivation of solving the learning problem, Zhou et al. [27] group pre-trained models-based continual learning studies into three categories, i.e., prompt-based methods, representation-based methods, and model mixture-based methods. In addition, this survey analyzes the experimental results of ten algorithms of the above three categories of methods on seven benchmark datasets. Wang et al. [4] classify continual learning methods into five main categories: regularization-based approach, replay-based approach, optimization-based approach, representation-based approach and architecture-based approach, where each category is analyzed in detail for its typical implementations and empirical properties. From the biological perspective of lifelong learning, Parisi et al. [29] analyze lifelong learning and catastrophic forgetting in neural networks. In addi-

tion, they summarize well-established and emerging neural network approaches driven by interdisciplinary research introducing findings from neuroscience, psychology, and cognitive sciences for the development of lifelong learning autonomous agents.

However, prior to this paper, little work has focused specifically on the rapidly emerging and highly promising field about lifelong learning of LLM-based Agents. In this survey, we collate the extensive literature about lifelong learning of LLM-based Agents, covering their construction, application, and evaluation processes.

2.2 Surveys on Large Language Model-based Agent

With the boom of LLM-based Agents, a wide variety of surveys have emerged that systematically summarize the research in this area. In this subsection, we briefly review these surveys, presenting their main contributions, as illustrated in Table 1.

From the perspective of agent architecture design, Wang et al. [13] propose a unified framework including four parts, namely profiling module, memory module, planning module and action module for the first time. In addition, this survey summarizes the applications, evaluations and challenges of LLM-based autonomous agent. Similarly, Xi et al. [14] introduce the three main components of the agent from the perspective of construction: brain, perception and action. In addition, this survey also investigates the practical applications of agents in different scenarios and the personal and social behaviors of agents.

Additionally, different focus and categorization produce a diverse understanding of the field. Some other papers review specific aspects of LLM-based agents. For example, Li et al. [30] present the background and technological advances of intelligent personal assistants and introduce the concept of personal LLM agents. This survey delves into the issues of fundamental capabilities and efficiency of personal LLM agents. Finally, it analyzes the challenges regarding security and privacy. Cheng et al. [31] consider the LLM-based agent system framework and analyze the single agent system and multi-agent system in detail. Similarly, this survey discusses the performance evaluation and prospect applications of LLM-based agent. Li et al. [32] propose a unified framework of the general multi-agent system, which includes five modules, namely profile, perception, self-action, mutual interaction and evolution. In addition, this survey also analyzes the application and some key open issues of multi-agent systems. Besides, Zhang et al. [33]define the concept of LLM-brained GUI agents and analyze the architecture and progress of LLM-brained GUI agents.

These surveys analyze important progress in building LLM-based agents, covering everything from personal LLM agents to multi-agent systems. However, they fall short in their connection to lifelong learning. Lifelong learning emphasizes the ability of agents to continuously learn and adapt in changing environments, whereas these surveys mainly focus on current technology implementations and application scenarios. In contrast to them, our survey provides an in-depth summary of how to equip LLM-based agents with the ability to learn and evolve in the long term, and how to optimize their performance as they continue to accumulate experience.

3 Building Lifelong Learning LLM Agents

We begin by presenting the formal definition of lifelong learning in LLM agents, followed by an overview of its historical development, and conclude with a detailed description of the overall architecture for lifelong learning in LLM agents.

3.1 Formal Definition of Lifelong Learning for LLM-based Agents

Definition 3.1 (Environment of LLM Agents)**.** *We model the environment of an LLM-based agent as a goal-conditional partially observable Markov decision process (POMDP). Formally, a POMDP is defined as an 8-tuple:*

$$\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, T, R, \Omega, O, \gamma), \tag{1}$$

where:

- $\mathcal{S}$ *is a set of states. Each* $s \in \mathcal{S}$ *can include multimodal information such as textual descriptions, images, or structured data (e.g., a product page on an e-commerce website containing text, images, and product specifications).*
- $\mathcal{A}$ *is a set of actions. Each* $a \in \mathcal{A}$ *represents an instruction or command that the agent can issue, often expressed in natural language (e.g., "add this item to the cart").*
- $\mathcal{G}$ *is a set of possible goals. Each* $g \in \mathcal{G}$ *specifies a particular objective (e.g., "purchase a laptop").*
- $T(s' \mid s, a)$ *is the state transition probability function. For each state-action pair* (s, a)*,* T *defines a probability distribution over next states* $s' \in \mathcal{S}$*. For example, if the agent clicks on a product link,* T *models the probability of transitioning to the product's detail page.*
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathcal{R}$ *is the goal-conditional reward function. For each triplet* (s, a, g)*,* R *may return a numeric value or textual feedback (e.g., "Good job!") indicating how well the action* a *taken in state* s *advances the objective* g*. This allows the environment to interactively provide user feedback aligned with the chosen goal.*
- Ω *is a set of observations. Each* $o' \in \Omega$ *can be textual, visual, or a combination thereof. Observations represent the agent's partial view of the underlying state (e.g., the content visible on a webpage).*
- $O(o' \mid s', a)$ *is the observation probability function. Given that the environment transitioned to state* s'*, and action* a *was taken,* O *defines the probability of receiving observation* o'*. For instance, upon navigating to a product page,* O *might model the probability of observing a particular product image and description.*
- $\gamma \in [0, 1)$ *is the discount factor, which balances immediate versus long-term rewards. In the context of LLM agents, the discount factor is used only when the reward is numeric.*

This framework captures the complexities of real-world scenarios. For instance, a virtual shopping assistant (the agent) interacts with a complex, multimodal website (the environment) to achieve a goal such as completing a purchase. The assistant receives partial observations (e.g., product listings), takes actions (e.g., clicking a link), and obtains feedback (numeric or textual) reflecting progress toward the goal.

Definition 3.2 (LLM-based Agent)**.** *An LLM-based agent is an agent whose policy and decision-making process rely on a large language model. The agent interacts with an environment $\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, T, R, \Omega, O, \gamma)$ as defined above. The agent's policy π is a mapping from its observations to actions, where $\pi(o_t) \in \mathcal{A}$ represents the action selected at each time step based on the observation $o_t \in \Omega$. At each time step t:*

- *The agent receives an observation $o_t \in \Omega$, which may include text, images, or other structured data.*
- *The agent selects an action $a_t \in \mathcal{A}$, typically by generating a textual command or query through its LLM-based policy $\pi(o_t)$.*
- *The environment returns a reward $r_t = R(s_t, a_t, g)$, potentially numeric or textual, guiding the agent towards achieving the goal g.*

The agent's policy π maps histories of observations and actions to a probability distribution over $\mathcal{A}$. By employing language understanding and generation capabilities, the agent can handle complex, real-world tasks where purely numeric feedback is insufficient.

Definition 3.3 (Task)**.** *A task $\mathcal{T}^{(i)}$ is given by:*

$$\mathcal{T}^{(i)} = \langle \mathcal{E}^{(i)}, o_0^{(i)}, g^{(i)} \rangle, \tag{2}$$

where:

- *$\mathcal{E}^{(i)} = (\mathcal{S}^{(i)}, \mathcal{A}^{(i)}, \mathcal{G}^{(i)}, T^{(i)}, R^{(i)}, \Omega^{(i)}, O^{(i)}, \gamma^{(i)})$ is the environment of the i-th task.*
- *$o_0^{(i)} \in \Omega^{(i)}$ is the initial observation. For example, the agent might start on a homepage of an e-commerce site.*
- *$g^{(i)} \in \mathcal{G}^{(i)}$ is the specific goal for this task (e.g., "buy a laptop under $1000").*

To solve $\mathcal{T}^{(i)}$, the agent must select actions that lead from an initial observation toward states fulfilling $g^{(i)}$, with rewards computed according to $R^{(i)}(s, a, g^{(i)})$.

Definition 3.4 (Trajectory)**.** *For a given task $\mathcal{T}^{(i)}$, a trajectory of length T_i is:*

$$\xi_{T_i}^{(i)} = \langle o_0, a_0, r_0, o_1, a_1, r_1, \ldots, o_{T_i}, a_{T_i}, r_{T_i} \rangle, \tag{3}$$

where $o_t \in \Omega^{(i)}$, $a_t \in \mathcal{A}^{(i)}$, and $r_t = R^{(i)}(s_t, a_t, g^{(i)})$. Each trajectory captures one sequence of interactions, starting from the initial observation and continuing until termination.

Definition 3.5 (Trial)**.** *A* trial *is one complete attempt by the agent to solve a given task $\mathcal{T}^{(i)}$. Each trial corresponds to one trajectory $\xi_{T_i}^{(i)}$. In a real-world setting, a trial might represent one entire session of trying to complete a purchase. Multiple trials may differ due to stochastic environment dynamics or policy variability.*

Definition 3.6 (Trajectory-Level and Step-Level Rewards)**.** *Rewards can be offered at different granularities:*

- Trajectory-Level Rewards*: Only the final outcome yields informative feedback. For example, the agent receives a positive reward (or a textual confirmation) only upon successfully completing the purchase.*
- Step-Level Rewards*: Intermediate actions also yield meaningful feedback (e.g., navigating to a more relevant product page might produce a positive textual hint).*

Because $R^{(i)} : \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \times \mathcal{G}^{(i)} \to \mathcal{R}^{(i)}$, each reward depends on the current goal $g^{(i)}$. In practical applications, this allows the environment to provide goal-aligned feedback, guiding the agent's actions in pursuit of that objective.

Definition 3.7 (Lifelong Learning Tasks)**.** *Lifelong learning considers a set of tasks:*

$$\mathcal{U} = \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(n)}\}. \tag{4}$$

Tasks may vary in terms of states, actions, goals, and rewards. Lifelong learning can be:

- Intra-environment*: All tasks share the same underlying environment structure, but differ in initial conditions or goals (e.g., different products to purchase on the same website).*
- Inter-environment*: Tasks arise from distinct environments, requiring the agent to adapt and transfer knowledge across different domains (e.g., switching from an e-commerce site to a travel booking platform).*

In real-world scenarios, agents continuously face new tasks and must accumulate knowledge without forgetting previous solutions, improving efficiency and competence over time.

Definition 3.8 (Objective of Lifelong Learning)**.** *Define a numeric mapping $\tilde{R}^{(i)}(o_t, a_t, g^{(i)})$ that converts the (possibly textual) reward $R^{(i)}(s_t, a_t, g^{(i)})$ into a scalar value. The performance of a policy π on task $\mathcal{T}^{(i)}$ is the expected cumulative reward:*

$$J(\mathcal{T}^{(i)}, \pi) = \mathbb{E}_{\xi_{T_i}^{(i)} \sim \pi} \left[\sum_{t=0}^{T_i} \tilde{R}^{(i)}(o_t, a_t, g^{(i)}) \right]. \tag{5}$$

The objective of lifelong learning is to find a policy π that maximizes the expected performance across all tasks in $\mathcal{U}$:

$$\max_{\pi} \sum_{i=1}^{n} J(\mathcal{T}^{(i)}, \pi). \tag{6}$$

This formulation encourages the agent to improve continuously, leveraging past experience and knowledge to excel at current and future tasks, much like a human learner facing increasingly challenging goals.

3.2 Background and History of Lifelong Learning for AI Systems

Lifelong learning, also referred to as continual or incremental learning, is grounded in the idea that intelligent systems should continually acquire, refine, and retain knowledge over extended periods—much like humans. Unlike traditional machine learning approaches that assume access to a fixed, stationary dataset, lifelong learning frameworks confront the reality that data and tasks evolve over time, and that models must adapt without forgetting previously mastered skills. A illustration of the development of lifelong learning is provided in Figure 5.

Human and Neuroscience Perspectives: The principles of lifelong learning draw inspiration from human cognitive development. Humans do not train on a fixed dataset; instead, we accumulate knowledge from diverse and changing experiences [34]. Memory consolidation in the human brain, involving complex interactions between the hippocampus and neocortex, ensures that new learning does not completely overwrite old memories. Studies of synaptic plasticity and learning in neuroscience help inform algorithms that attempt to preserve and integrate previously

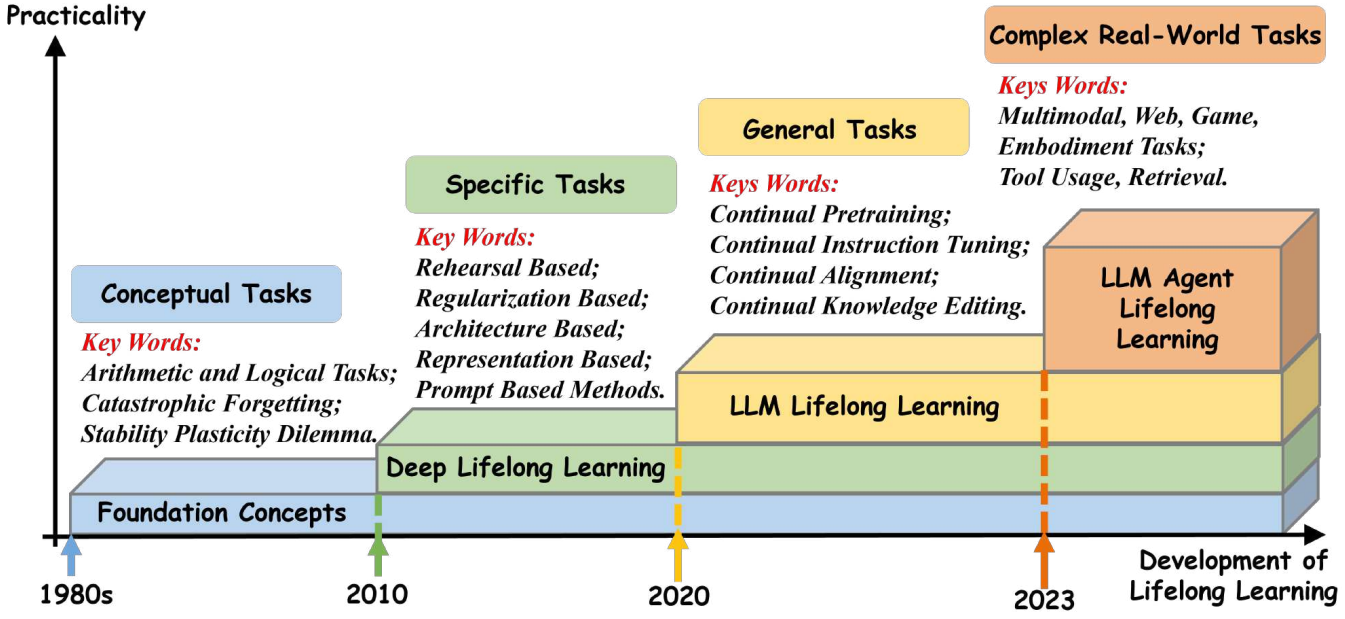


Fig. 5. Development of lifelong learning for AI systems, highlighting four key stages: (1) Establishment of foundational concepts starting in the 1980s, (2) Advancements in deep lifelong learning from 2010 to the present, (3) Integration of lifelong learning into large language models from 2020 onwards, and (4) the latest developments in lifelong learning for LLM agents. The practicality of lifelong learning has significantly improved alongside its development, enabling more versatile and adaptive AI systems in diverse real-world applications.

acquired representations while adapting to new information. Incorporating such principles into neural networks has been a longstanding challenge and motivation for research in lifelong learning [35]–[37].

3.2.1 Foundation Concepts (1980s–Present)

The origins of lifelong learning research can be traced back to early adaptive control and incremental learning studies in the 1980s and 1990s. Researchers recognized that many real-world environments are non-stationary: the distribution of data changes over time, and models must adapt accordingly. During this period, foundational concepts such as catastrophic forgetting and the stability-plasticity dilemma were first identified. Catastrophic forgetting describes the tendency of neural networks to lose previously acquired knowledge when trained on new data [6], [38], while the stability-plasticity dilemma [34], [39] addresses the need to balance retaining old knowledge (stability) with acquiring new information (plasticity).

Throughout the 1990s and into the 2000s, catastrophic forgetting became a central research focus. Early attempts to mitigate forgetting relied on replaying previously encountered examples, adjusting network architectures to accommodate new tasks, or using heuristics to preserve existing representations [9]. As neural networks became more widely adopted across domains, more sophisticated methods for lifelong learning were developed. Researchers introduced structured approaches, such as dynamically expanding architectures that allocate new network capacity for novel tasks, dual-memory frameworks inspired by the interplay between short-term and long-term memory in the brain, and early forms of regularization and parameter isolation techniques to reduce interference with older knowledge [29], [40].

3.2.2 Deep Lifelong Learning (2010–Present)

The rise of deep learning in the early 2010s significantly expanded the capabilities of neural networks. However, these initial breakthroughs often assumed a stationary training dataset. Incorporating incremental learning into deep networks required new strategies. Techniques like Elastic Weight Consolidation (EWC) [41] and Learning without Forgetting (LwF) [42] became popular, using careful parameter adjustments to preserve previously learned representations. These methods [2] leveraged deep learning's powerful representational ability while addressing the core issue of catastrophic forgetting, pushing lifelong learning closer to practical real-world applicability.

During this period, various approaches were categorized based on their methodologies: *Rehearsal-Based Methods:* Involve replaying or retraining on a subset of previously seen data to retain past knowledge [43]. *Regularization-Based Methods:* Apply constraints or penalties to the loss function to prevent significant changes to important parameters [41]. *Architecture-Based Methods:* Dynamically modify the network architecture to accommodate new tasks without interfering with existing ones [44]. *Representation-Based Methods:* Focus on learning robust and transferable representations that facilitate knowledge retention and transfer [45]. *Prompt-Based Methods:* Utilize prompts or auxiliary inputs to guide the model in retaining and utilizing past knowledge [46].

3.2.3 LLM Lifelong Learning (2020–Present)

With the advent of large-scale pre-training, especially in language models, the landscape of AI has been reshaped. LLMs like GPT-3 [11] introduced context-aware word representations, enabling models to perform a wide range of Natural Language Processing (NLP) tasks with high efficiency. Lifelong learning in LLMs initially focused on conventional

NLP tasks such as text classification, machine translation, and instruction following. Techniques developed during this period include parameter-efficient fine-tuning [11], [47], retrieval-augmented learning [48], and prompt-based adaptation [11]. These approaches allow LLMs to continuously integrate new linguistic knowledge and adapt to evolving language patterns without extensive retraining, thereby enhancing their performance on established NLP benchmarks.

Key applications and methods in this period include: *Continual Pretraining:* Continual pretraining of models on domain-specific corpora or newly available datasets, such as Wikipedia, to incorporate the most up-to-date knowledge or adapt to additional languages [49]. *Continual Instruction Tuning:* Incremental fine-tuning of the model to improve its ability to follow a diverse set of instructions, enhancing its performance across various tasks such as summarization, translation, and instruction following [50], [51]. *Continual Alignment:* Ensuring that the model's outputs remain aligned with human values, ethical guidelines, and user preferences as it learns new tasks, thereby maintaining trustworthiness and relevance [52]. *Continual Knowledge Editing:* Addressing outdated or incorrect information within LLMs, effectively mitigating the risk of hallucinated or inaccurate outputs [53], [54].

3.2.4 LLM-based Agents Lifelong Learning (2023–Present)

Starting around 2023, the focus of lifelong learning has expanded from conventional NLP tasks to more realistic and complex applications embodied by LLM-based agents. Unlike LLMs, which primarily handle tasks such as text generation and classification, LLM-based agents are designed to interact with dynamic environments and perform intricate tasks like online shopping, household management, operating system operations, and more. These agents require advanced lifelong learning capabilities to manage multimodal inputs, execute sequential decision-making processes, and maintain coherent performance across diverse and evolving tasks.

Key advancements in this period include: *Dynamic Task Adaptation:* Developing models that can seamlessly switch between varied tasks without compromising performance on previously learned ones [21]–[23]. *Multimodal Integration:* Enhancing agents' abilities to process and integrate information from multiple modalities (e.g., text, images, and sensor data) to perform complex real-world tasks [18]–[20]. *Memory and Knowledge Management:* Implementing sophisticated memory systems that allow agents to retain and efficiently retrieve past experiences, facilitating better decision-making and knowledge transfer [55]–[57]. *Reinforcement Learning Integration:* Combining lifelong learning with reinforcement learning techniques to enable agents to learn optimal policies in interactive environments continuously [15], [33], [58]. *External Knowledge Integration:* Enabling agents to utilize external tools and databases to enhance their capabilities, as well as integrating retrieval mechanisms to access relevant information on-the-fly [48], [59]. *Broader Real-world Applications:* Developing chatbots and interactive agents that can sustain long-term dialogues, adapt to user preferences, and perform tasks within web or gaming environments [60]–[62].

These advancements empower LLM-based agents to operate in more interactive and unpredictable settings, reflecting real-world complexities. For example, an online shopping assistant must not only understand and generate natural language but also navigate product databases, handle user preferences, and adapt to new product categories over time. Similarly, a household management agent must integrate visual inputs from cameras, interpret voice commands, and learn to manage various home devices, all while retaining knowledge from previous interactions.

3.2.5 Summary

In summary, the evolution of lifelong learning in AI systems is a story of increasing sophistication—moving from foundational concepts aimed at mitigating forgetting to advanced, principled methods informed by cognitive science and neuroscience. The advent of LLMs has further propelled the field, initially enhancing conventional NLP tasks and now expanding into the realm of LLM-based agents that tackle complex, real-world applications. As LLMs and other large-scale models become integral to real-world applications, lifelong learning stands at the forefront, driving innovation toward systems that improve continuously and gracefully over their operational lifetimes.

3.3 Overall Architecture

The architecture of a lifelong learning LLM-based agent is designed to continually adapt, integrate, and optimize its behavior across a range of tasks and environments. In this subsection, we identify three essential modules—Perception, Memory, and Action—that enable lifelong learning. This division follows the framework introduced in prior work [14], with one notable difference: rather than retaining the "Brain" module, we adopt a "Memory" module as proposed in [14], offering clearer functionality and improved modularity.

Each module interacts with the others to ensure that the agent can process new information, retain valuable knowledge, and select contextually appropriate actions. The rationale for these three modules stems from the agent's need to (i) perceive and interpret evolving data, (ii) store and manage knowledge from past experiences, and (iii) perform tasks that adapt to changing circumstances.

These modules form a dynamic feedback loop: the Perception module delivers new information to the Memory module, where it is stored and processed. The Memory module then guides the Action module, influencing the environment and informing future perception. Through this continuous cycle, agents progressively refine their knowledge and improve their adaptability, ultimately enhancing their performance in complex, dynamic environments.

In what follows, we describe each module in detail, examining how its design contributes to the agent's lifelong learning capabilities. Figure 6 provides an illustration of this overall architecture, while Figure 7 summarizes the organization of the subsequent sections.

3.3.1 Perception Module

The perception module of a lifelong learning LLM-based agent is responsible for acquiring and integrating informa-

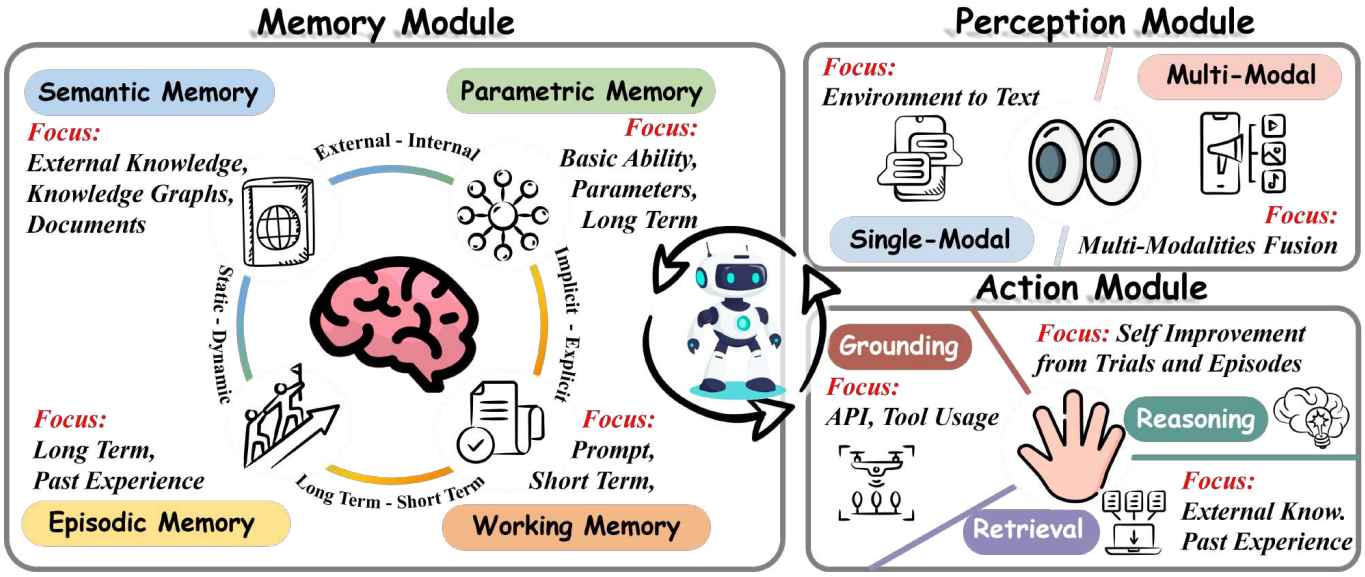


Fig. 6. Overall architecture of a lifelong learning LLM-based agent, comprising three key modules—Perception, Memory, and Action. The Perception module continuously gathers and integrates information from the environment. The Memory module stores and manages knowledge from past experiences. The Action module guides interactions and decision-making based on the agent's current goals and stored knowledge. Together, these modules form a dynamic feedback loop, allowing the agent to adapt, learn, and optimize its behavior across diverse tasks and environments. This design replaces the "Brain" module from previous frameworks, offering a clearer decomposition of functionalities necessary for lifelong learning.

tion from the environment. Similar to humans, who continuously update their understanding through sensory input, agents must perceive and process diverse data sources to remain effective across varying tasks. This module plays a crucial role in adapting the agent's behavior based on new or evolving contexts.

We divide the perception module into two primary categories: **single-modal perception** and **multimodal perception**. *Single-modal perception:* This refers to the agent's ability to continuously learn from a single modality, typically textual information. This allows the agent to develop deep domain-specific knowledge, such as understanding webpages or engaging in textual interactions within games or other specialized tasks. *Multimodal perception:* This expands the agent's ability to integrate information from multiple modalities—such as visual, auditory, and text data—into a unified understanding of the environment. By combining these different sensory inputs, multimodal perception enables the agent to form a more comprehensive, robust understanding, which is essential for tasks requiring context from various sources (e.g., robotic control, multimedia analysis).

The integration of both single-modal and multimodal perception allows LLM-based agents to continuously learn and adapt to diverse, ever-changing environments.

3.3.2 Memory Module

The memory module in a lifelong learning LLM agent allows the agent to store, retain, and recall information—essential for both learning from past experiences and improving decision-making. Memory is the foundation for an agent's ability to develop coherent long-term behavior, make informed decisions, and interact meaningfully with other agents or humans. The memory module thus supports the agent's ability to learn from experience, avoid catastrophic forgetting, and enable collaborative behaviors.

We categorize the memory module into four key types: **Working Memory**, **Episodic Memory**, **Semantic Memory**, and **Parametric Memory**. These four types work together to provide a comprehensive memory system: *Working Memory:* This represents the agent's short-term memory, responsible for handling immediate context, such as prompts, user inputs, and relevant workspace information. It enables the agent to act upon the present context in real-time, providing the foundation for short-term reasoning and decision-making. *Episodic Memory:* This type of memory stores long-term experiences and events, such as user interactions, previous task outcomes, or multi-turn dialogues. Episodic memory helps the agent recall past experiences to improve its future actions, while maintaining consistency in long-term behavior and learning. *Semantic Memory:* This type of memory functions as an external knowledge store, helping the agent acquire and update world knowledge. Through mechanisms like continual knowledge graph learning and continual document learning, semantic memory facilitates the integration of new knowledge into the agent's internal framework. By leveraging external databases such as knowledge graphs or dynamic document corpora, semantic memory ensures the agent can stay current with evolving information, improving both its ability to answer queries and enhance long-term learning. *Parametric Memory:* Unlike the explicit memory of past events, parametric memory resides within the internal parameters of the model. Changes in these parameters—such as through fine-tuning or training updates—reflect long-term knowledge and contribute to the agent's general knowledge base. This memory type allows the agent to retain knowledge across tasks without storing explicit event details.

The synergy between these memory types supports the

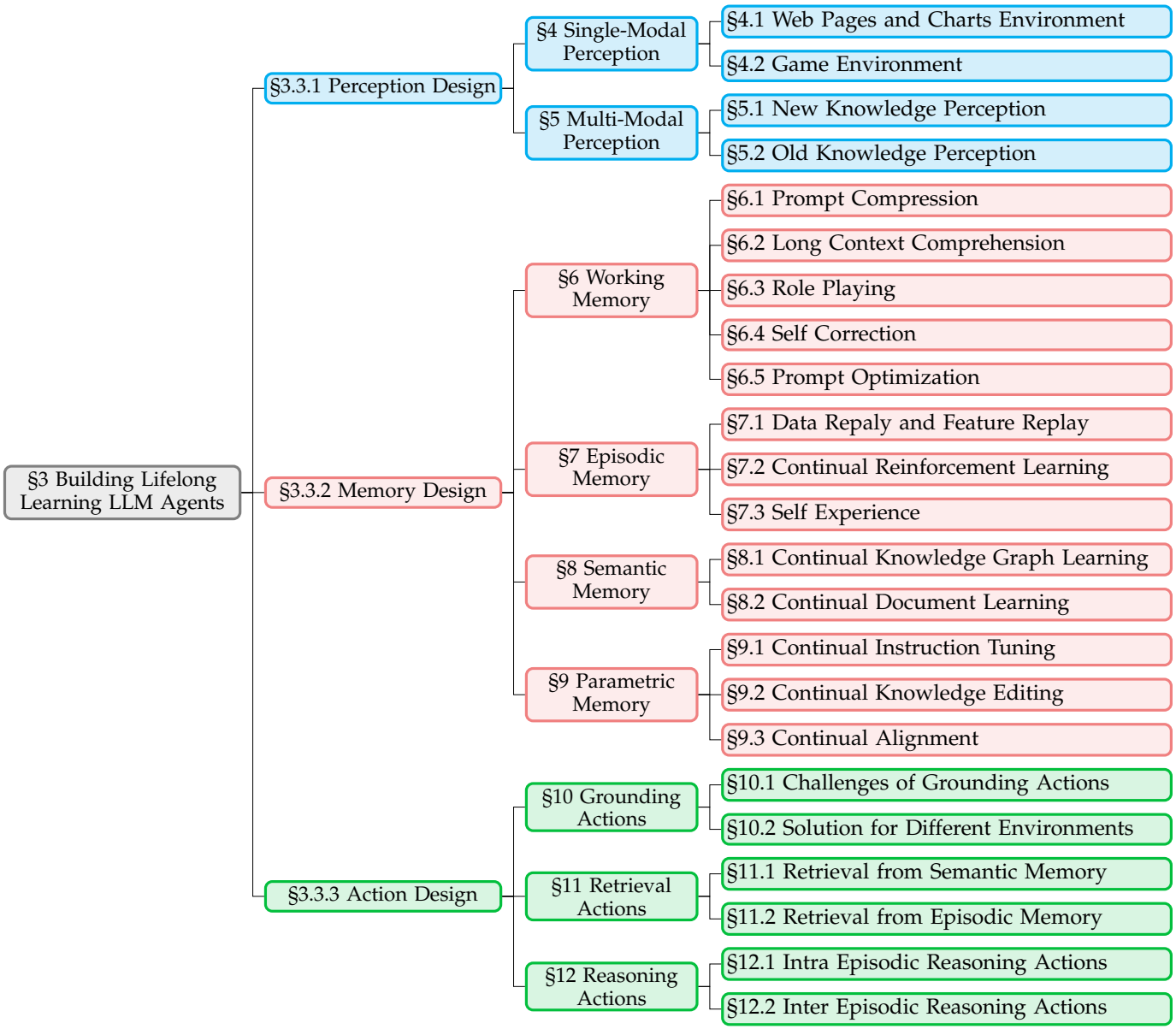


Fig. 7. Hierarchical organization of the survey sections, highlighting each component of the lifelong learning LLM-based agent architecture and its key functionalities.

agent's ability to continuously learn, adapt, and avoid catastrophic forgetting, making it capable of lifelong learning.

3.3.3 Action Module

The action module enables the agent to interact with its environment, make decisions, and execute behaviors that shape the course of its learning. In a lifelong learning framework, actions are crucial for closing the feedback loop: actions influence the environment, and the environment provides feedback that is used to refine future actions.

We categorize actions into three main types: **Grounding Actions**, **Retrieval Actions**, and **Reasoning Actions**: *Grounding Actions:* These are the primary means by which the agent interacts with the environment. Grounding actions involve physically or digitally affecting the environment, whether through manipulating objects, generating text, or triggering specific behaviors. The effects of these actions may persist, influencing future behavior and feedback loops. *Retrieval Actions:* These actions enable the agent to access and retrieve relevant information from its memory, whether from semantic memory (e.g., general knowledge) or episodic memory (e.g., past experiences). Retrieval actions help the agent maintain consistency, acquire new insights, and enhance decision-making. *Reasoning Actions:* These involve the agent using its working memory, past experiences, and external data to perform complex reasoning, planning, or decision-making tasks. Reasoning actions are essential for tasks that require long-term planning, multi-step decision-making, or the integration of diverse sources of information. As illustrated in Figure 3, these three action types allow the agent to continuously interact with, learn from, and adapt to its environment, supporting a lifelong process of improvement and behavioral evolution.

4 Perception Design: Single-modal Perception

The single-modal perception of LLM-based agents primarily involves the reception of textual information. Throughout the process of lifelong learning, the sources of textual information that agents encounter may originate from various structures and environments. In natural text environments, current LLM-based systems have demonstrated the fundamental capability to communicate with humans through text input and output [12], [63]. Building on this foundation, the agent needs to acquire text information from non-natural text environments to better simulate information perception in the real world. Therefore, in this section, we present methods for single-modality perception in environments such as *web* and *game* environments.

4.1 Web Pages and Charts Environment

Certain methods have been developed to extract structured text that adheres to standardized formats [21], [64]–[67], thereby transforming complex information into a format accessible to LLM-based agents. The mainstream approaches can be broadly classified into two categories: *HTML manipulation* and *screenshot*. For instance, Synapse [65] and AgentOccam [64] simplify HTML elements from web pages and selectively incorporate them into prompts, while WebAgent [66] summarizes HTML documents and breaks instructions down into multiple sub-instructions, proposing an enhanced planning strategy. Additionally, to effectively harness the information provided by images, several studies [68]–[72] have converted screenshots into textual formats for alignment with LLMs.

4.2 Game Environment

LLM-based agents can perceive their surroundings through textual mediums [73]–[75], recognizing elements such as characters, time, location, events, and emotions [76], and can perform corresponding actions based on the feedback from these gaming elements using textual instructions. For example, JARVIS-1 [73] improves its understanding of the environment through self-reflection and self-explanation, incorporating previous plans into its prompts. VillagerAgent [74] utilizes a dedicated state manager module to filter task-relevant environmental information from the global context. By interfacing with graphical user interfaces (GUIs), LLM-based agents can enhance their ability to extract textual information from visual data [69], [72], [77], [78], thus facilitating a better understanding of the graphics and elements within user interfaces.

In conclusion, a human-like LLM-based agent should possess strong text perception and adaptability across a variety of complex environments. As related research continues to grow, exploring how agents perceive text input in broader and more diverse environments holds significant promise for future advancements.

5 Perception Design: Multimodal Perception

The real world comprises diverse data modalities, making unimodal perception methods insufficient to address its complexity. With the explosive growth of image, text, and video content on online platforms, developing LLM-based agents capable of continuously perceiving multimodal information has become crucial. These agents must effectively integrate information from different modalities while maintaining the accumulation and adaptation of prior knowledge. This enables them to better emulate the lifelong learning process of humans in multimodal environments, thereby enhancing their overall perceptual and cognitive capabilities. In this section, we categorize the lifelong learning methods for the agent's perception of multimodal information into *new knowledge perception* and *old knowledge perception*. A comparison and summarization of the related methods in provided in Table 2.

5.1 New Knowledge Perception

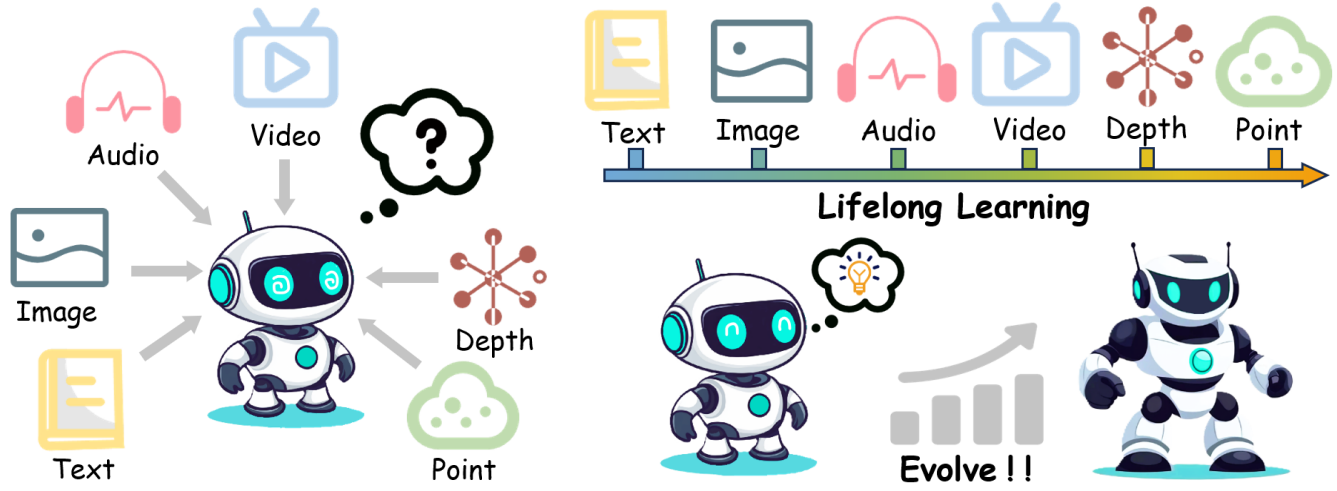
In lifelong learning scenarios involving the perception of various modalities, agents must focus on the interaction between different modalities and the perception and processing of new modalities to better handle the rapidly evolving forms of information in the real world. Many studies explore how an agent can maintain stability on tasks involving existing modalities while improving its capability to handle new tasks when encountering those with new modalities. We categorize new knowledge perception into the following two scenarios: *Modality-Complete Learning* and *Modality-Incomplete Learning*.

5.1.1 Modality-Complete Learning

Modality-Complete Learning assumes that all data has the same modality during both the training and inference stages. In this scenario, the agent's lifelong learning for multimodal perception focuses on how to perceive data from multiple modalities and achieve cross-modal knowledge transfer in new tasks.

Some studies [79]–[82] have explored Modality-Agnostic Models, which can perceive any modality as input. Perceiver [79] utilizes an iterative attention mechanism to dynamically encode any modality of data and generate a unified high-dimensional representation. VATT [80] adopts a shared transformer architecture and learns cross-modal correlations effectively from raw video, audio, and text through multimodal self-supervised learning tasks. Omnivore [81] introduces a multimodal feature aggregation mechanism, using shared convolutional layers and an adaptive feature fusion strategy to achieve a unified representation of various visual modalities such as images, videos, and depth maps. ModaVerse [82] leverages LLMs as the core to build an efficient modality transformer framework, enabling fast information fusion across different modalities and improving the processing efficiency of multimodal tasks.

On the other hand, Cross-Modal Knowledge Transfer refers to the effective application of knowledge acquired in one modality to improve performance on tasks involving another modality [83], [84], [87], [89], [90], [112]. Some methods achieve knowledge transfer between different modalities by feeding connected unimodal features into linear layers [87], [89], [90]. In visual-language tasks, some studies [83], [84], [112] focus on transferring visual knowledge into language models. For example, Vokenization [83] introduces



(a) Agents Perceiving Multimodalies (b) Agents Perceiving New Modalities Incrementally

Fig. 8. Comparisons of agents to perceive multimodal information: (a) traditional agents require cross-modal data for joint training; (b) lifelong learning agents incrementally learning new modality information without joint-modal retraining.

"visual tokens" and performs text-to-image retrieval using contextual images, embedding visual information into the training of language models to enhance language understanding. VidLanKD [84] uses contrastive learning to train a teacher model on video datasets and distills the extracted visual features into the language model, improving its performance on text tasks related to video.

Additionally, by combining knowledge distillation [85], [113] and regularization methods, some studies [84], [86], [88], [102], [103] leverage the high-performance capability of the teacher model in one modality to transfer its knowledge to the student model through constraint-based mapping. For example, the CTP [103] method employs a distillation strategy that incorporates compatible momentum contrast and topology-preserving techniques, achieving a balance between old and new task knowledge in visual-language continual pretraining, effectively alleviating catastrophic forgetting. Mod-X [102] uses contrastive matrix alignment of off-diagonal information and spatial distribution distillation to maintain alignment of the representation space across old data domains from different modalities.

5.1.2 Modality-Incomplete Learning

Modality-Incomplete Learning involves the challenge of how the agent can dynamically adapt to effectively learn and infer when encountering incomplete or missing modality information during lifelong learning (See Figure 8). By utilizing the MoE [114] module, PathWeave [92] introduces a novel Adapter in Adapter (AnA) framework, enabling the seamless integration of single-modal and cross-modal adapters, allowing for incremental learning of new modal knowledge. Similarly, DDAS [93] can automatically route inputs within and outside the training data distribution to the MoE adapter and the original CLIP model, respectively, to achieve efficient dynamic modality adaptation.

Some studies [91], [94]–[96] have utilized available modality information to predict the representation of the missing modality. Specifically, SMIL [91] proposes an adaptive modality weighting mechanism, enhancing robustness to severely missing modalities through multimodal fusion. MMIN [94] uses conditional generative adversarial networks to generate imagined features for the missing modality, improving performance in sentiment recognition tasks. DiCMoR [95] introduces a generative model to minimize the difference between available modalities and the true feature distribution, thereby enabling accurate modality recovery. The latest research [96] projects the multi-dimensional features of different modalities into a shared space and uses pseudo-supervision to indicate the reliability of modality prediction, enabling generalization to unseen modality combinations.

Other studies improve the representation of input data by learning shared and modality-specific features [97], [98], which helps demonstrate better robustness when handling missing modalities. Similarly, Ma et al. [99], based on the sensitivity of transformer models to missing modalities, constructed a robust transformer optimized through multi-task learning. They also developed an algorithm to automatically search for the optimal fusion strategy across different datasets to optimize test-time robustness.

In summary, new knowledge perception emphasizes the incremental perception of multimodal information by LLM-based agents in real-world complex environments. The agent perceives new modality inputs and integrating the new knowledge with existing modality experiences to enable lifelong learning perception.

5.2 Old Knowledge Perception

In the process of lifelong learning, the agent not only needs to leverage existing modality experiences to complete tasks involving new modalities, but also must maintain the stability of old knowledge after receiving new information. In this section, we present a classification of representative

TABLE 2
Multimodal perception of Lifelong agents. "Focus": *MA* means processing Modality-Agnostic inputs, *CKT* means crossModal knowledge transfer, *DA* means dynamic adaptation to multimodal, *MCF* means Mitigating catastrophic forgetting; "Modality":✓ indicates the modality involved in the method.

		Method	Focus	Contribution	Modality					Code
					Language	Vision	Audio	Video	Other	
New Knowledge Perception	Modality-Complete	Perceiver [79]	MA	Utilize asymmetric attention mechanisms to process multimodal data.	✓	✓	✓	✓	✓	Link
		VATT [80]	MA	Utilize self-supervised multimodal learning methods.	✓		✓	✓		Link
		Omnivore [81]	MA	Introduce a multimodal feature aggregation mechanism.	✓	✓	✓			Link
		ModaVerse [82]	MA	Build a modal conversion framework for rapid multimodal integration.	✓	✓	✓	✓		Link
		Vokenization [83]	CKT	Integrate visual information into the training of language models.	✓	✓				Link
		VidLanKD [84]	CKT	Utilize cross-modal KD to achieve cross-modal knowledge transfer.	✓		✓			Link
		Gupta et al. [85]	CKT	Transfer the features learned from one modality as supervision signals to another.	✓	✓	✓			Link
		ZSCL [86]	CKT, MCF	Match image-text similarity distributions through KD.	✓	✓				Link
		SoundSpaces [87]	CKT	Integrate audio and visual information to facilitate navigation in 3D environments.		✓	✓		✓	Link
		LMC [88]	CKT, MCF	Build a multimodal knowledge graph to enhance information between modalities.	✓	✓	✓	✓		Link
		EPIC-Fusion [89]	CKT	Enhance egocentric action recognition using audio-visual temporal binding.		✓	✓		✓	Link
		Zhang et al. [90]	CKT	Combine visual and auditory cues to improve the counting of repetitive activities.		✓	✓			Link
	Modality-Incomplete	SMIL [91]	DA	Adopt an adaptive modality weighting mechanism to enhance robustness.	✓	✓	✓			Link
		PathWeave [92]	DA, CKT	Introduce a novel 'Adapter in Adapter' framework to achieve modality alignment.	✓	✓	✓			Link
		DDAS [93]	DA, CKT	Hand modality data for in-distribution and out-of-distribution by routing.	✓	✓				Link
		MMIN [94]	DA	Generate imagined features of missing modalities using conditional GANs.	✓	✓	✓			Link
		DiCMoR [95]	DA	Use a generative model to minimize the feature difference for modality recovery.	✓	✓	✓			Link
		Zhang et al. [96]	DA	Generalize unseen modality combinations via projection and pseudo-supervision.	✓	✓	✓		✓	Link
		ShaSpec [97]	DA	Hand missing modalities using shared and specific feature representations.		✓	✓		✓	Link
		MissModal [98]	DA	Introduce constraints to align representations of missing modality data.	✓	✓	✓			Link
		Ma et al. [99]	DA	Improve robustness by automatically finding the optimal data fusion strategy.	✓	✓				-
Old Knowledge Perception	Regularization	TIR [100]	MCF	Adapt weight regularization based on the similarity between old and new tasks.	✓	✓				Link
		Model Tailor [101]	MCF	Identifying key parameters for Combine regularization.	✓	✓				-
		Mod-X [102]	CKT, MCF	Distill on contrastive matrix to preserve spatial distribution between modalities.	✓	✓				-
		CTP [103]	CKT, MCF	Utilize KD to maintain the similarity distributions between image and text.	✓	✓				Link
	Replay	Vqacl [104]	MCF	Randomly Replay past task samples to maintain old task performance.	✓	✓				Link
		SAMM [105]	CKT, MCF	Store randomly selected training samples in a memory buffer to avoid forgetting.		✓	✓			Link
		TAM-CL [106]	CKT, MCF	Combine KD and replay to transfer knowledge from old tasks.	✓	✓				Link
		KDR [107]	MCF	Combine KD to replay the image-text similarity matrix output by the old task model.	✓	✓				-
		IncCLIP [108]	CKT, MCF	Generate negative text replay and multimodal knowledge distillation.	✓	✓				-
		SGP [109]	MCF	replay past knowledge through randomly sampled scene graph relationships.	✓	✓				Link
		FGVIRs [110]	MCF	Store old knowledge to create pseudo-representations for training classifier.		✓			✓	-
		AID[38] [111]	MCF	Retain old knowledge by Enhance the model's representational capacity.		✓			✓	-

multimodal perception lifelong learning methods aimed at addressing the problem of catastrophic forgetting.

5.2.1 Regularization-based Approach

The regularization-based Approach aims to mitigate the phenomenon of catastrophic forgetting by introducing regularization terms that limit the changes in model parameters during the learning of new tasks. Based on the method of constraint application, the Regularization-based Approach can be subdivided into two directions: *weight Regularization* and *function Regularization*.

Weight Regularization directly applies penalty terms to the model's weights, restricting their changes when learning new tasks. For example, TIR [100] and Model Tailor [101] utilize existing importance measurement methods like EWC [41] to calculate the importance of parameters for old tasks, thereby applying penalties.

Function Regularization focuses on constraining the intermediate or final outputs of the model, ensuring that the model retains the output features of old tasks while learning new tasks. Function Regularization often uses knowledge distillation strategies, where the previously learned model serves as the teacher, and the model learning the new task serves as the student [86], [88], [102], [103] For example, the CTP [102] method achieves a balance between old and new task knowledge in visual-language continual pretraining through momentum contrast and topology-preserving distillation strategies, effectively reducing catastrophic forgetting. Mod-X [88] aligns non-diagonal information in the contrastive matrix and distills spatial distribution to maintain the alignment of the representation space across modalities for old data domains.

5.2.2 Replay-based Approach

The Replay-based Approach is a method that alleviates catastrophic forgetting by preserving and reusing previously learned experiences. In multimodal continual perception learning, depending on the specific content of the replay, the method can be divided into *Experience Replay* and *Generative Replay*.

Due to storage space limitations, Experience Replay focuses on how to store more representative old training samples in a limited memory space. Some studies [104], [105] randomly select multimodal samples for replay. TAM-CL [106] and KDR [107] combine experience replay with

knowledge distillation strategies to constrain distribution shifts, thus reducing catastrophic forgetting.

Generative Replay requires training an additional generative model to replay generated data [108]–[111], which effectively reduces storage requirements. IncCLIP [108] enhances continuous visual-language pretraining by generating negative sample texts, thus improving the model's robustness when facing new tasks. SGP [109] uses scene graphs as prompts and integrates them with language models to enhance continual learning in visual question answering tasks.

Existing studies also include projection-based and architecture-based methods. Projection-based methods map data from different modalities (e.g., images, text, and audio) to a unified feature space [94], [96], [115]–[118], enabling the model to process information from multiple modalities. The architecture-based approach is a strategy that supports lifelong learning by adjusting the model's structure. This approach divides the model into task-shared and task-specific components, ensuring relative isolation between tasks to reduce the impact of learning new tasks on the retention of old knowledge [119].

In conclusion, old knowledge perception focuses on the agent's ability to retain and perceive old knowledge as it continually receives multimodal information during lifelong learning. By incorporating existing lifelong learning methods, the agent can significantly mitigate the issue of catastrophic forgetting.

6 Memory Design: Working Memory

Working Memory is primarily manifested as the short-term memory of the agent, encompassing elements such as prompts, workspace memory, and context provided by users. Agents utilize these prompts or contextual inputs as their working memory to generate responses or to proceed with planning and actions. Working memory serves as the operational memory of the agent, facilitating real-time interactions and decision-making processes. In short, working memory is the active workspace of the agent, where immediate information is processed and manipulated to produce responses or to guide subsequent actions. We discuss working memory from five main perspectives: **prompt compression**, **long context comprehension**, **role playing**, **self correction**, and **prompt optimization**, as illustrated in Figure 9.

6.1 Prompt Compression

In terms of working memory, agents can include more contextual content within the limited length of prompts by compressing prompt words that users input. This process not only increases the efficiency of information processing, but also allows agents to effectively avoid catastrophic forgetting of historical information when integrating old prompts into new ones. In this way, agents can retain memories of previous experiences while continuously learning new information, achieving the goal of lifelong learning. The techniques for prompt compression are classified mainly into two categories: **soft compression** and **hard compression**.

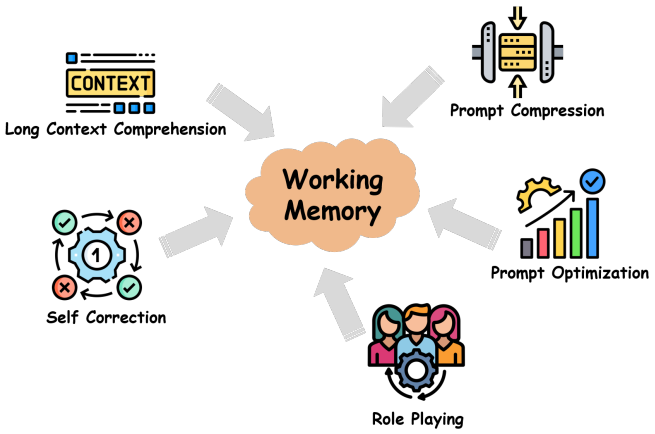


Fig. 9. Potential techniques for working memory.

Soft compression learns alternative forms of embeddings by optimizing a small number of soft prompt tokens to compress the original prompts, preserving the abstract sentiment or key information. AutoCompressors [120] can process the prompts input to the model by recursively generating summary vectors that are passed as soft prompts to all subsequent prompt segments. It is ingenious to fine-tune the existing model structure by increasing the vocabulary and utilizing *summary tokens* and *summary vectors* to refine a large amount of prompt information. Mu et al. [121] propose *gisting* to compress prompts into a smaller set of *gist tokens* by training the language model. Considering that gist tokens are much shorter than the length of the original input prompts, these tokens can be cached and reused, thus improving computational efficiency.

In contrast, *hard compression* focuses on directly compressing prompts by filtering out redundant or non-essential tokens based on various scores computed by a pre-trained model. Selective Context [122] employs a mini-language model to compute the self-information values of individual lexical units (e.g., sentences, phrases, or words) in a given prompt. By retaining only the higher self-information values, Selective Context provides a more concise and efficient prompt representation for large language models. LLMLingua [123] argues that Selective Context often ignores the intrinsic connection between compressed content and the synergy between LLMs and small language models used for prompt compression. To solve the problem, LLMLingua utilizes a budget controller to dynamically assign different compression rates to the components of the original prompt (e.g., demo samples and questions). At the same time, it adopts a coarse-grained demonstration-level compression strategy and introduces a fine-grained Iterative Token-level Prompt Compression algorithm to further optimize the prompt compression process. However, the problem with LLMLingua is that it ignores the questions posed by the user during the compression process, which may lead to some irrelevant information being unnecessarily retained. To address this shortcoming, LongLLMLingua [124] innovatively incorporates the consideration and processing of user questions in the compression process. The LongLLMLingua framework introduces four new features, including question-aware coarse-grained and fine-grained compres-

sion, document reordering mechanism, dynamic compression ratio, and subsequence recovery algorithm to improve the ability of the large language model to recognize key information. In general, soft compression outputs compressed summary vectors or tokens, while hard compression outputs filtered text.

6.2 Long Context Comprehension

It is a common scenario that longer text inputs occur frequently in working memory. When processing long text inputs from working memory, the agent not only improves its comprehension of the long text but also realizes the effect of continuously adapting to new text in lifelong learning as it continuously processes the long text. There are two main approaches for long text processing, **context selection** and **context aggregation**.

Context selection splits long text into multiple paragraphs and selects specific paragraphs based on predefined criteria. CogLTX [125] proposes *MemRecall* for extracting key blocks. This process requires the input of strides to indicate how many new chunks are to be retained at each step. MemRecall evaluates the relevance of each chunk based on the sequence of key chunks that have been currently selected and queries by calculating a relevance score and then selects the chunk with the highest score. This process involves score calculation across attention and after comparison, as well as review and decay after retaining a certain number of chunks. MCS [126] evaluates the importance of sentences by ranking them with an extraction-based labeling module [127] and an attention-based module [128] to determine which sentences should be included in the final summary.

Context aggregation is a technique for efficiently aggregating neighborhood or context information in neural networks. It enhances the richness of feature representation by integrating feature information from different regions and the ability of the model to understand local and global context. Container [129] is a generic building block for multi-head context aggregation. It is able to exploit the long-distance interactions in the Transformer model while maintaining the inductive preference for local convolutional operations to achieve faster convergence. Container combines static and dynamic affinity aggregation through learnable mixing coefficients, allowing it to handle long textual information. Traditional Transformer-based pre-trained language models cannot be applied to long sequences due to their quadratic complexity. To address this problem, SLED [130] handles long text comprehension tasks by segmenting the input into overlapping chunks, encoding each chunk using a short text language model encoder, and fusing information between the chunks (fusion-in-decoder) using a pre-trained decoder. PCW [131] decomposes long text into small chunks that can be processed by the LLM by splitting the long text into multiple consecutive *chunks* (*windows*). Within each window, the attentional mechanism of LLMs is applied only to tokens within that window, while positional embeddings are reused across windows, which maintains information about the sequence order and allows the model to process long text beyond the limits of a single context window.

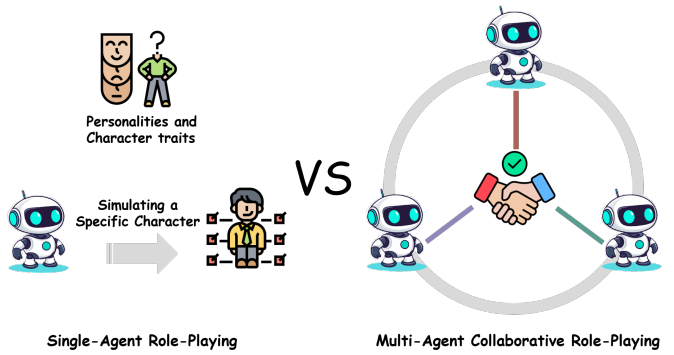


Fig. 10. Comparison between single-agent role-playing and multi-agent collaborative role-playing.

6.3 Role Playing

In working memory, agents are designed to respond to user commands and assume specific roles within these directives. These roles are not just functional; they also come with distinct personalities and character traits, enabling agents to exhibit richer and more multidimensional communication with users. As agents play these roles, they continuously learn from each interaction, absorbing new knowledge and adjusting their behaviors to provide more precise and personalized services. This ongoing learning and adaptation is the core mechanism by which agents achieve lifelong learning. Considering the process of role-playing, there are two primary forms: **single-agent role-playing** and **multi-agent collaborative role-playing**, as illustrated in Figure 10.

In role-playing with *a single agent*, the focus is on constructing an agent that can simulate a specific character [132], [133]. Attributes of the character are first defined, including personality traits and backstory. Data related to the character is then collected, which may include text, conversation history, and character-specific behavioral patterns. Next, a large language model is used to build the agent so that it can generate language and behavior based on the definition of the character. Afterwards, the agent interacts with the user to simulate the conversations and behaviors of the character. Finally, the performance of the agent is evaluated to ensure that its behavior and language are consistent with the definition of the character. Character-LLM [134] extracts scenes based on these personal experiences as memory flashbacks by collecting character-specific experiences with the help of LLMs. These scenes are then expanded by the LLM agent into complete scenarios containing details of fabrication so that Character-LLM can learn to form roles and emotions from detailed experiences. Compared to previous role-playing models that only mimic through style, Character-LLM provides a deeper level of character understanding by simulating the mental activities and physical behaviors of a character.

In role-playing with *multiple agents*, multiple agents work together. The user assigns specific roles and tasks to each agent to enable more complex interactions. MetaGPT [135] uses an assembly line paradigm and encodes Standardized Operating Procedures into prompt sequences. It introduces a metaprogramming approach to human workflows to efficiently decompose collaborative software engi-

neering tasks into sub-tasks to be performed by multiple agents. This ultimately enables agents with human domain expertise to validate results and minimize errors. IBSEN [136] is a framework for generating controlled and interactive theater scripts through the collaboration of *director agents* and *actor agents*. The director agent is responsible for writing the outline of the plot that the user would like to see, and directing the actor agents to role-play according to the plot development. In a multi-agent cooperative text game with theory of mind inference tasks, Li et al. [137] form a team of three embodied agents in the experiment. In each round, the three agents of the team interact with the environment in turn, receive observations and perform actions through natural language. In addition, the implementation of multi-agent communication enables the LLM-based agents to share text messages within the team. Magentic-One [138] employs a coordinator agent (orchestrator) working in concert with four specialized agents. The orchestrator is responsible for task decomposition, planning, subtask assignment, progress tracking and error correction. The other four specialized agents include webSurfer (web browser manipulation), filesurfer (local file handling), coder (code writing and execution) and computerterminal (terminal access).

6.4 Self Correction

In the working memory of the agent, the user inputs specific prompts to instruct the agent to review and evaluate its previous responses to identify and correct any potential errors, enabling the self-correction function of the agent. This process optimizes the output of the model by requiring the agent not only to identify errors, but also to rethink and provide a corrected answer. In this way, the agent is able to continuously learn and improve from the prompts, enabling lifelong learning. The main strategies for self-correction include the following: **relying on feedback from the output of other models** [139], **assessing one's own confidence level** [140], and **resorting to external tools** [141].

In terms of *relying on feedback from the output of other models*, N-CRITICS [139] utilizes several different generic LLMs as critics that evaluate the output generated by the primary LLM model and provide feedback. It uses an iterative feedback mechanism and does not require supervised training. The initial output is evaluated by a collection of critics and the collected criticisms are used to guide the primary LLM model to iteratively correct its outputs until specific stopping conditions are met. With regard to *assessing one's own confidence level*, Li et al. [140] propose an *If-or-Else* prompting framework to guide LLMs in assessing their own confidence and facilitate intrinsic self-corrections. Considering *resorting to external tools*, CRITIC [141] guides large language models to self-correct by interacting with *external tools*. The core idea of this framework is to mimic human behavior in using external tools (e.g., a search engine for fact-checking or a code interpreter for debugging) to verify and correct initial content.

6.5 Prompt Optimization

In working memory, agents often need to process prompt words input by users, which can sometimes be too broad or vague, and even lead to misunderstandings. To address this challenge and enhance the quality and accuracy of the response to these instructions, prompt optimization technology is introduced to refine and clarify the commands of the user. Through this technology, agents can gain a deeper understanding of user intentions and provide more precise and expected responses. In this process, agents continuously learn from each interaction, gradually improving their comprehension and generation capabilities, achieving lifelong learning effects, and maintaining adaptability and effectiveness in the ever-changing linguistic environment. Research in this field primarily focuses on developing theoretical algorithms to systematically enhance the prompts of agents, which include but are not limited to **evolutionary algorithms** [142] and **Monte Carlo Tree Search algorithms** [143].

EvoPrompt [142] draws on the idea of *evolutionary algorithms* to generate new prompt candidates using LLMs and improve the prompt population based on the performance of the development set. Starting from an initial prompt population, EvoPrompt iteratively uses LLMs to generate new prompts based on evolutionary operators and selects better prompts based on the performance of the development set. In each iteration, EvoPrompt uses LLMs as evolutionary operators to generate new prompts based on several parent prompts selected from the current population. Apart from evolutionary algorithms, Monte Carlo Tree Search algorithms also work for prompt optimization. The core of PromptAgent [143] lies in its view of prompt optimization as a strategy planning problem and its use of the *Monte Carlo Tree Search algorithm* to strategically navigate the expert-level prompt space. By simulating the human trial-and-error exploration process, PromptAgent is able to iteratively check and optimize intermediate prompts by reflecting on model errors and generating constructive error feedback.

6.6 Summary

Working memory is the short-term memory of the agent, including prompts, workspace memory, and user-provided context. Agents use this information to generate responses or for planning and action. In order to include more contextual content within a limited prompt length, the agent compresses the prompt words that users input. This helps to improve the efficiency of information processing and avoids forgetting historical information when integrating old prompts into new ones. In addition, the need for intelligent agents to process long text inputs in working memory not only improves comprehension of long text, but also achieves the effect of adapting to new text while continuously processing long text. In terms of role-playing, agents are designed to play specific roles in user commands, which have different personalities and characteristics, enabling richer and multidimensional communication between the agent and the user. Considering self correction, the agent reviews and evaluates previous responses in working memory based on specific prompts entered by the user, identifying and correcting potential errors and realizing self-corrective functions. In order to improve the quality and accuracy of the response of the agent to the commands of the user, a prompt optimization technique is introduced to refine and

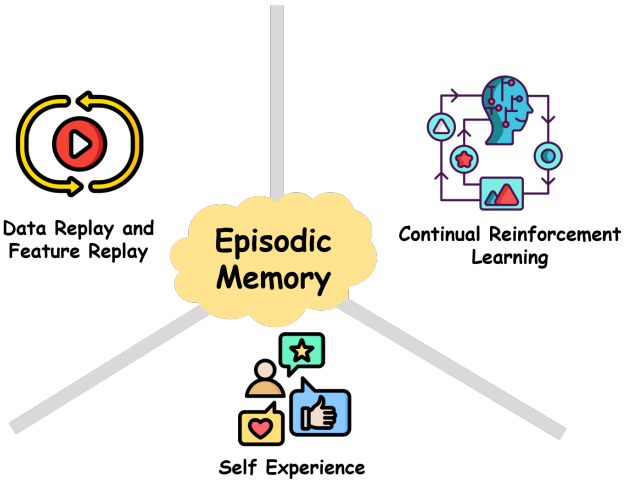


Fig. 11. Potential techniques for episodic memory.

clarify the commands of the user. This helps the agent to understand the intent of the user more deeply and provide more precise and expected responses.

7 Memory Design: Episodic Memory

Episodic memory is the repository of the agent for past experiences. It is the memory of specific events, encounters, and interactions that the agent has been a part of. This form of memory is essential for learning from past interactions and for developing a historical understanding that can inform future actions. It enables the agent to recall previous conversations, learn from past mistakes, and build on successes, thereby improving its performance over time. Next, we will delve into the concept of episodic memory from the perspectives of **data replay and feature replay**, **continual reinforcement learning**, and **self experiences**, as illustrated in Figure 11.

7.1 Data Replay and Feature Replay

Replay is a method that reuses information from old tasks when training new tasks. It mainly utilizes samples from episodic memory and replays these samples during the training of a new task to help the model not to forget the old task when facing a new task, thus achieving the goal of lifelong learning.

Data replay mainly involves two types of replay techniques, **experience replay** and **generative replay**. *Experience replay* involves keeping a small portion of old training samples in the model so that these samples can be replayed while training a new task as a way of maintaining the memory of the old task. The key challenge with this approach is how to select and utilize these stored samples. Some early work used fixed principles to select samples, such as Reservoir Sampling [144] which randomly retains a certain number of old training samples from each training batch. On this basis, gradient-based or optimizable strategies are more feasible, such as selecting samples by maximizing sample diversity [145]. To improve storage efficiency, some methods such as AQM [146] online continuously compress the data and save the compressed data for replay. In addition, experience replay can be used in conjunction with knowledge distillation. For example, iCaRL [147] and EEIL [148] can transfer knowledge between old and new tasks by performing knowledge distillation on old and new training samples, which improves the ability of the model to generalize to old tasks.

Generative replay, also known as pseudo-rehearsal, is a technique where an extra generative model is trained to generate data for the purpose of replay, which creates a correlation with the incremental updates of the generative model. For example, MeRGAN [149] enforces the consistency of sampling between the old and new generative models through replay alignment. Besides, DGR [150] utilizes a cooperative dual model architecture consisting of a deep generative model (generator) and a task solving model (solver). The generated data are coupled with the associated responses from the previous task solver to represent the old tasks. In addition, other lifelong learning strategies can be integrated into the work of alleviating catastrophic forgetting of generative models, such as weight regularization [151] and experience replay [152].

Feature replay [153] is a strategy that focuses on preserving the distribution of features rather than the data itself, which offers significant advantages in terms of efficiency and privacy. This approach is distinct from data-level replay in that it addresses the challenge of representation shift, a phenomenon where the sequential updating of feature extractors leads to a loss of previously learned features, akin to feature-level catastrophic forgetting. RER [154] takes a direct approach to estimate the shifts in representation, allowing for the updating of preserved old features to align with new ones.

7.2 Continual Reinforcement Learning

The collected experiences in the data buffer are an important manifestation of episodic memory. We focus on the **experience replay** technique commonly used in continual reinforcement learning for the reason that it allows agents to learn from early historical memories, accelerate the learning process and break poor temporal correlations. The main idea of experience replay is to enhance the stability of training and improve learning efficiency by repeatedly presenting experiences stored in the replay buffer. These experiences are composed of tuples, with the tuple of each time step including a state, an action taken, the next state, and a reward. Experience replay mitigates the problem of catastrophic forgetting by sampling from the buffer during training and storing observed samples in the experience pool, ultimately achieving the goal of lifelong learning.

Lin [155] proposes the concept of *Experience Replay* in 1992, which significantly contributed to the development of many reinforcement learning algorithms. Schaul et al. [156] propose Prioritized Experience Replay in 2015, where the frequency of replay is determined based on the importance of the experiences (usually their magnitude of temporal differential error), as a way to allow agents to learn more frequently those experiences that are most helpful for current strategy improvement. Based on PER, subsequent studies have made a variety of improvements, as illustrated in

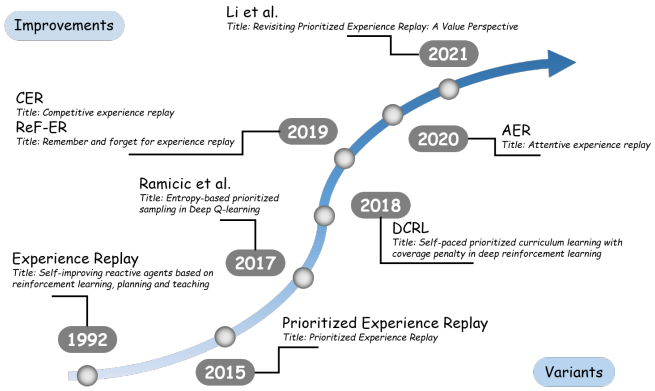


Fig. 12. Experience replay in continual reinforcement learning.

Figure 12. For example, Ramicic et al. [157] use state entropy as a prioritization criterion and Li et al. [158] propose and prioritize three experience-based value measures.

In addition to prioritized replay, a variety of other replay strategies have emerged, such as DCRL [159], ReF-ER [160], AER [161] and CER [162]. DCRL [159] proposes an importance criterion for samples in terms of difficulty and diversity of experience, where difficulty is positively correlated with temporal differential error and diversity is correlated with the number of replaying times. ReF-ER, AER, and CER rely on higher computational resources. Specifically, ReF-ER [160] penalizes policy updates via Kullback-Leibler divergence to accelerate convergence. CER [162] analyzes competitive exploration by setting competition between a pair of agents. AER [161] selects experiences based on the similarity between the states in past transitions and the current state of the agent, prioritizing to the similar transitions.

7.3 Self Experience

Episodic memory is an essential component of the long-term memory of the agent, enabling it to store and revisit experiences, including the outcomes of those experiences, whether successful or not, and the feedback from the external environment regarding its actions. These memories form the repository of *self experiences*, which the agent can leverage to retrieve relevant information and enhance its decision-making processes and action plans. By doing so, the agent not only learns from past experiences but also anticipates and adapts to future scenarios, achieving lifelong learning. This capability makes the agent more agile and effective in complex and changing environments, continually learning and evolving from interactions. When constructing the self experiences of LLM agents, we meticulously categorize the data types stored in the memory. These data types primarily include four categories: *triplets*, *databases*, *documents*, and *conversations*, as illustrated in Table 3.

In terms of *triplets*, RET-LLM [163] proposes a generalized read and write memory module that stores knowledge in the form of triplets. It is inspired by the Davidsonian semantics theory, which describes concepts as <first argument, relation, second argument> structures. The memory module stores triplets and their vector representations. During retrieval, the query text is first searched for exact matches, and if none are found, a fuzzy search is performed based on

TABLE 3
Categorization of data types stored in the memory module of LLM agents.

Data Type	Method
Triplets	RET-LLM [163], GEM [164], DiCGRL [165], LKGE [166], SSRM [167], EARL [168], FastKGE [169]
Databases	ChatDB [60], LangChain [170], LlamaIndex [171], PrivateGPT [172], BINDER [173], SQL-PALM [174], P_2SQL [175], LECSP [176], On et al. [177], DB-GPT [178]
Documents	DelTA [179], DSI++ [180], IncDSI [181], PromptDSI [182], CorpusBrain [183], CorpusBrain++ [184], CLEVER [88], LangChain [170], LlamaIndex [171], D-BOT [185]
Conversations	MemoChat [57], RAISE [186], CHATS [187], AutoGen [188], PPDPP [189], MindDial [190], PLATO-LTM [191], Memory Sandbox [192], Bae et al. [193]

the vector representations. In terms of *databases*, ChatDB [60] uses a database as a symbolic memory module to support abstract, scalable and precise manipulation of historical information. It allows complex reasoning and querying of stored memories using SQL statements. In terms of *documents*, DelTA [179] is designed to overcome the challenge of maintaining translation consistency and accuracy while processing an entire document. This multi-level memory structure includes proper noun records, bilingual summary, long-term memory, and short-term memory. For instance, long-term memory stores a wider range of contextual information and assists LLMs in retrieving the most relevant sentence pairs to the current source sentence as a small sample learning demonstration.

Conversations are a vital form of information storage within episodic memory. When users engage in prolonged multi-turn conversations with agents, these agents must be able to recall and reference previous exchanges to maintain the coherence and depth of the conversation. By meticulously managing these conversations, agents can not only cite historical information in future responses but also learn from them, optimizing their reaction patterns to better meet user needs. This mechanism of reviewing and learning from historical conversations is central to agents achieving lifelong learning and continuously improving their conversational skills, enabling them to progress with each interaction, ultimately achieving more natural and precise conversation outcomes. MemoChat [57] allows agents to dynamically retrieve and utilize past conversational information in long conversations through the construction and use of dynamic memory banks. In this way, this information is utilized to maintain the consistency of the conversation. Inspired by ReAct [194], RAISE [186] enhances the capabilities of conversational agents by incorporating *scratchpad*, which is similar to short-term memory and processes the information about recent interactions. It provides a flexible

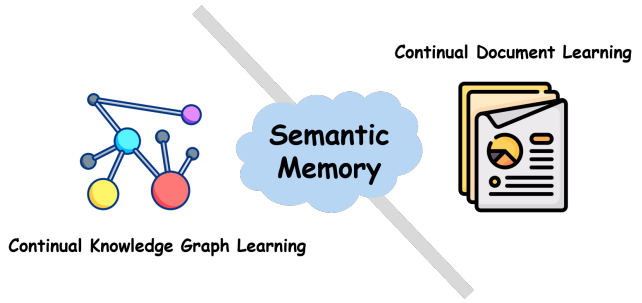


Fig. 13. Potential techniques for semantic memory.

way to augment the capabilities of conversational AI, and allows humans to customize and control the behavior of the conversational AI system.

7.4 Summary

Working memory serves as the operational memory and the active workspace of the agent, facilitating real-time interactions. Compared to working memory, episodic memory is a repository where agents store past experiences, including memories of specific events, encounters, and interactions. Data Replay and Feature Replay trains new tasks by reusing data or feature distributions from old tasks. Continual Reinforcement Learning accelerates the learning process and breaks bad temporal associations by allowing the agent to learn from early historical memories through experiences collected in a data buffer. The agent utilizes the results of the experience and the feedback from the external environment on its actions to constitute a self experience repository in which relevant information is eventually retrieved to enhance the decision-making process and action plan.

8 Memory Design: Semantic Memory

In llm-based agents, semantic memory serves as an external memory mechanism, playing a critical role in storing and updating world knowledge. Semantic memory not only assists agents in retrieving known information but also enables lifelong learning by progressively integrating new knowledge. In this section, we focus on the implementation of semantic memory for lifelong learning in *continual knowledge graph learning* and *continual document learning*, as illustrated in Figure 13.

8.1 Continual Knowledge Graph Learning

Knowledge graph embedding (KGE) [195] is a technique that maps entities and relations within a knowledge graph into a low-dimensional vector space, which is essential for various downstream applications. However, with the rapid growth of knowledge, traditional static KGE methods [196]–[198] typically require retaining the entire knowledge graph when new knowledge emerges, leading to significant training costs. To address this challenge, the task of continual knowledge graph embedding (CKGE) has emerged. CKGE leverages incremental learning to optimize the updating process of knowledge graphs, aiming to efficiently learn new knowledge [169] while preserving existing knowledge [164], [166], [199]. Current CKGE methods can be broadly categorized into three main types:

8.1.1 Replay-based Approach

Replay-based Approach [164], [165], [200] mitigates catastrophic forgetting by storing portions of previous graph states or information and integrating them with the current graph state to guide new embedding learning. For example, GEM [164] introduces a memory mechanism that alleviates catastrophic forgetting by replaying gradient information from old tasks. Similarly, DiCGRL [165] is a decoupled continual graph representation learning framework. It decouples relational triplets in the graph into multiple independent components based on their semantic aspects and learns decoupled graph embeddings using knowledge graph embedding and network embedding methods. These embeddings are stored and replayed as needed. This framework enables the selective updating of relevant old relational triplets when new ones arrive, focusing only on the corresponding components of their graph embeddings. This not only improves the adaptability of the model to new tasks but also enhances its ability to transfer knowledge across different tasks.

8.1.2 Regularization-based Approach

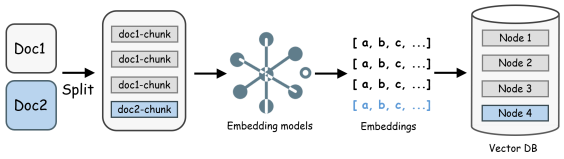
Regularization-based Approach [166], [167] effectively mitigates catastrophic forgetting by introducing penalty terms. For instance, LKGE [166] addresses the continually expanding knowledge graph by incorporating a regularization term that constrains the distance between new and old embeddings. This ensures that existing knowledge is preserved when new knowledge is added, facilitating efficient knowledge transfer and integration. Similarly, SSRM [167] incorporates a structural shift mitigation term into the loss function, minimizing the distance between vertex representations in previous and current graph structures. This helps identify a latent space that minimizes the impact of structural shifts, thereby reducing catastrophic forgetting.

8.1.3 Architecture-based Approach

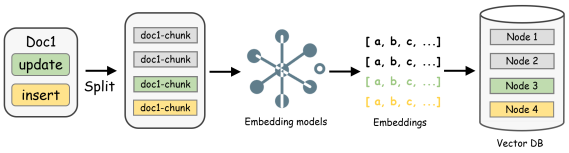
Architecture-based Approach adapts architectural properties to accommodate new information effectively. For example, EARL [168] is an entity-agnostic representation learning framework that supports transfer learning across different knowledge graphs. By reducing dependency on specific entities, it enhances the adaptability and parameter efficiency of the model. Some studies [169] leverage the LoRA [201] low-rank adaptation strategy to achieve continual learning. For instance, FastKGE [169] isolates new knowledge based on the fine-grained influence between old and new knowledge graphs, assigning it to specific layers to mitigate catastrophic forgetting. To accelerate fine-tuning, FastKGE incorporates an incremental low-rank adapter (IncLoRA) mechanism, embedding specific layers into incremental low-rank adapters with fewer training parameters, significantly improving training efficiency.

8.2 Continual Document Learning

LLM-based agents can use information retrieval (IR) systems to map user queries to relevant documents. Previous research [202] primarily focused on generation-based retrieval from static document corpora. However, in practice, the documents available for retrieval are constantly



(a) Document-level update strategy



(b) Chunk-level update strategy

Fig. 14. Continual Document Learning in RAG Applications.

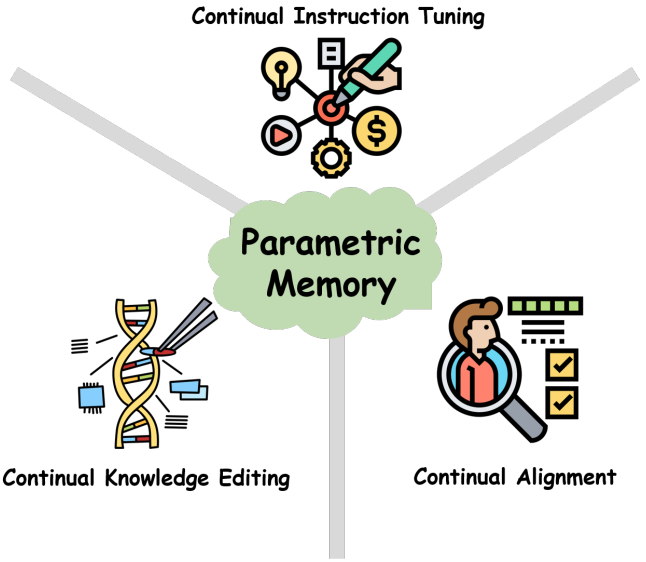


Fig. 15. Potential techniques for parametric memory.

updated and expanded, especially in dynamic sources like news, scientific literature, and other rapidly changing information domains. This fast-paced evolution of documents presents significant challenges for retrieval systems. Therefore, in recent years, more research has focused on how to quickly and efficiently integrate new information into dynamic corpora, particularly in the field of information retrieval. For example, some studies [180]–[182] enhance the capability of document updates in dynamic corpora based on the DSI [203] method. DSI++ [180] introduces a Transformer-parameterized memory mechanism, designing a dynamic update strategy that allows the model to optimize its internal representations when new documents arrive, thus achieving efficient retrieval adaptation. IncDSI [181] employs a modular indexing update strategy, leveraging previously constructed index data to support the rapid insertion of new documents, significantly reducing computational resource demands and ensuring real-time retrieval efficiency. PromptDSI [182] adopts a prompt-based rehearsal-free incremental learning approach, emphasizing the use of a prompt mechanism to guide the model in retaining memory of old documents during the update process, thus eliminating the need for rehearsal samples.

For Specific Tasks, CorpusBrain++ [184] introduces a new benchmark dataset, KILT++, based on the original KILTdataset for evaluation. It employs a backbone-adapter architecture, which maintains the base retrieval capabilities while introducing task-specific adapters for incremental learning of downstream tasks. This approach effectively mitigates catastrophic forgetting and addresses the limitations of CorpusBrain [183], which only focused on static document collections. Similarly, in the context of continual learning for generative retrieval, CLEVER [88] employs Incremental Product Quantizationto cost-effectively encode new documents. It incorporates a memory-augmented learning mechanism that allows the model to effectively memorize new documents without forgetting previously acquired knowledge, enabling efficient retrieval of both old and new documents.

In the context of RAG applications, incremental updates of knowledge documents are essential to ensure timely synchronization of domain-specific knowledge. Current research primarily adopts two strategies (see Figure 14) for incremental updates: document-level and chunk-level updates. Document-level updates involve comprehensive parsing and vectorization of newly added or updated documents. Chunk-level updates focus on identifying newly added, modified, deleted, or unchanged knowledge chunks within documents. These updates rely on "fingerprinting" techniques, which use persistence and caching mechanisms to compare fingerprints and identify content that requires processing. Corresponding insertions or deletions are then performed. During this process, frameworks such as LangChain [170] utilize index APIs to skip unchanged knowledge chunks, preventing redundant entries in vector databases. Similarly, LlamaIndex [171] provides a data ingestion pipeline that supports incremental knowledge updates, specifying document storage and management strategies. Incremental knowledge updates are critical for enterprise-level RAG applications, enabling rapid adaptation to knowledge changes while reducing operational costs.

9 Memory Design: Parametric Memory

Parametric Memory, embedded within the fabric of LLMs, is the most intangible of the three. It is the collective knowledge encoded in the internal parameters of the model, shaped by the data the model was trained on and the fine-tuning processes it undergoes. This memory is not explicitly accessible or retrievable like the other two, which allows the model to understand and generate human-like text, to make inferences, and to adapt to new contexts based on its accumulated knowledge. We will develop our analysis from three perspectives: **continual instruction tuning**, **continual knowledge editing** and **continual alignment**, as illustrated in Figure 15.

9.1 Continual Instruction Tuning

During continual instruction tuning, the agent updates its parametric memory by continuously utilizing the instruction dataset to adjust the internal parameters of the model.

This tuning is not simply a one-time modification, but *a continuous process* that allows the agent to continuously optimize its knowledge base as it receives new instructions. In this way, the agent is able to not only retain and utilize past experiences, but also seamlessly integrate newly learned information, avoiding the loss of old knowledge due to new learning, i.e., catastrophic forgetting. This mechanism of continuous learning and memory updating is the key for agents to achieve lifelong learning.

Continual instruction tuning enhances the performance of large language models in specific scenarios or domains by fine-tuning their parameters with datasets from those areas. The capabilities that the model significantly enhances through this fine-tuning fall into two main categories: **specific capabilities** and **general capabilities**.

9.1.1 Specific Capabilities

In terms of *specific capabilities*, the model is fine-tuned by using datasets that contain domain-specific data, thereby enhancing its capabilities in those domains, such as the ability to *utilize specialized tools* [59], [204] and the ability to *derive solutions to mathematical problems* [205], [206]. Some studies focus on **specific agent capabilities**. In terms of the *tool-using capabilities* of the agents, ToolLLM [204] serves as a general tool-use framework that provides functions such as data construction, model training, and evaluation. Through the three processes of API collection, instruction generation, and solution path annotation, Qin et al. [204] construct an instruction-tuning dataset named ToolBench for tool use with the use of ChatGPT. ToolLLaMA is gained by fine-tuning LLaMA on ToolBench, which shows great ability to handle a wide variety of tool instructions and demonstrates strong generalization to previously unseen APIs. By referring to a few human-written examples of how to use APIs provided by humans, Schick et al. [59] use language models to identify and call potential APIs, thereby annotating a vast language modeling dataset. Subsequently, they used self-supervised loss to identify which API calls truly assist the model in predicting subsequent tokens. Building on this, they finetune the language model on the API call methods that the model deems practical and then obtain Toolformer, which enables the language model not only to master the operation of various tools but also to independently decide when and how to use specific tools. In terms of the *mathematical capabilities* of the agents, Xu et al. [206] propose a self-critique pipeline that evaluates the mathematical output of a Math-Critique model by deriving it from the LLM itself, allowing the model to learn from AI-generated feedback specific to the mathematical content. Its self-evolution includes rejection fine-tuning and direct preference optimization, which focus on the accuracy and consistency of the model in mathematical answers and learning from correct and incorrect answer pairs, respectively. RFT [205] is proposed by Yuan et al. to expand data samples and improve model performance. RFT sample and select correct reasoning paths as augmented dataset by applying rejection sampling on supervised fine-tuning models. Yuan et al. use these augmented datasets to fine-tune base LLMs, which achieves better performances on tasks of mathematical reasoning compared to supervised fine-tuning.

Except for training on datasets that contain domain-specific data, several studies analyze agents trained on **particular agent tasks**. For instance, Zeng et al. [207] construct AgentInstruct, a fine-tuned dataset containing high-quality interaction traces covering six real-world agent tasks (e.g., ALFWorld, WebShop, etc.). AgentTuning employs a hybrid training strategy that combines the AgentInstruct with instruction data from the generic domain to improve the model's performance on specific tasks while maintaining its generic capabilities. LUMOS [208], an unified and learnable language agent framework for training open-source LLM-based agents, consists of a planning module, a grounding module, and an execution module, which is widely applicable to a wide range of sophisticated interaction tasks and effective cross-task generalization. Yin et al. [208] propose two interaction formulations for implementing the language agents, LUMOS-OnePass and LUMOS-Iterative, with the purpose of resolving tasks through the agent modules. Additionally, they propose an annotation conversion method and construct multi-task multi-domain agent training annotations for agent fine-tuning, including question answering, mathematics, coding, web browsing, multimodal reasoning, text games, etc. By training on these annotations, LUMOS demonstrate improved or comparable performance with GPT-based or larger open-source agents in different kinds of complicated interaction tasks.

9.1.2 General Capabilities

In terms of **general capabilities**, models are fine-tuned with broad and general datasets to improve their understanding of human user inputs and to generate more satisfactory responses. For instance, instruction tuning enhances LLMs in terms of code, commonsense reasoning, world knowledge, reading comprehension, and math, which are typically evaluated using the following benchmarks: HumanEval [209] for code, HellaSwag [210] for commonsense reasoning, TriviaQA [211] for world knowledge, BoolQ [212] for reading comprehension, and GSM8K [213] for math. Apart from them, popular aggregated benchmarks include MMLU [214], Big Bench Hard [215] and so on.

Some studies focus on **general agent capabilities**. For instance, DebateGPT [216] is a method for supervised fine-tuning of large language models through multi-agent debate. The core of the method is to utilize multiple smaller language models (e.g., GPT-3.5) to debate in order to generate high-quality training data without relying on expensive human feedback. Each participating agent gives its own opinion in the debate and generates a confidence score for its answer. During each round of debate, responses from other agents are summarized using a summary model in order to provide clear final answers. By introducing confidence scores and a summary model, DebateGPT is able to generate higher quality data, which improves fine-tuning. FireAct [217] primarily utilizes powerful language models (e.g., GPT-4) to generate diverse inference trajectories, which are then used to fine-tune smaller language models. In the process of generating reasoning trajectories, FireAct not only utilizes data in ReAct [194] format, but also integrates CoT [218] and Reflexion [58] data resources, which increases the diversity and complexity of fine-tuning data. FireAct employs the thought-action-observation cycle, through which

the language model generates free-form thinking, executes free-form thinking, performs structured actions, and interacts with the environment to receive observation feedback.

Through continual instruction tuning, large language models not only retain their broad knowledge base but also evolve continuously based on the latest data and instructions, achieving ongoing learning and improvement. This also leads to the concept of **self evolution**, which is a manifestation of the enhancement of general capabilities. The process of self evolution for an agent is diverse and intricate, involving multiple stages of iteration and learning, which enables the agent to adapt to new tasks and environments continuously. Self evolution mainly involves first generating a solution for a task, then refining and generalizing the solution based on the feedback from the environment, using the refined experience to update the model of the agent, and finally evaluating the updated model in a new task [219]. Throughout the process of self evolution, the agent can continue to learn, adapt, and improve throughout its entire lifecycle, achieving lifelong learning.

For instance, Self-Instruct [220] evolves itself by aligning the language model with self-generated instructions and then fine-tuning itself according to those instructions. Specifically, this iterative bootstrapping algorithm begins with an initial pool of tasks consisting of manually-written tasks. Initially, the framework will guide the model to produce instructions for new tasks. Based on these newly generated instructions, the framework further constructs corresponding input-output samples that will be used in subsequent instruction fine-tuning sessions. Before adding the tasks back to the initial repository of tasks, the framework uses different kinds of heuristics in order to filter low-quality or similar generations. This process can be repeated for many iterations until a great quantity of tasks are obtained and the resultant data can be used for instruction tuning in the language model for better instruction following. Chen et al. [221] analyze the prospect of enhancing the performance of LLMs without relying on additional human or AI feedback and propose SPIN, a new fine-tuning method. The essence of SPIN lies in its self-play mechanisms [222], where the LLM enhances its performance by playing against instances of itself without any direct supervision. Beginning from a supervised fine-tuned model, the LLM optimizes its policy by generating training data from previous iterations and distinguishing this self-generated data from human-annotated data so that the strongest LLM can no longer distinguish between responses produced by their previous versions and those produced by humans.

9.2 Continual Knowledge Editing

In the process of continual knowledge editing, agents constantly utilize updated datasets—new knowledge—to modify erroneous or outdated information that was implicitly stored in previous models. This fine-tuning of the internal parameters achieves an update effect on the parametric memory of the model, allowing the agent to absorb and integrate new information while maintaining past knowledge. This process not only prevents the loss of old knowledge due to new learning, known as catastrophic forgetting, but also realizes the lifelong learning of the agent, enabling it to adapt in a changing environment continuously. Continual Knowledge Editing updates the understanding of the model through knowledge triplets-like *(head_entity, relation, tail_entity)* to ensure that the model is able to adjust its knowledge base when it discovers that prior knowledge is outdated or when it encounters new information [1]. There are three main types of continual knowledge editing methods: **external memorization**, **global optimization** and **local modification** [231].

External memorization-based methods utilize *external structures* to store new knowledge for editing without modifying the weights of the LLM. For example, WISE [232] designs a dual parametric memory scheme that contains main memory and side memory. Main memory is used to store pre-trained knowledge, while side memory is used to store edited knowledge. WISE trains a router to decide at the which memory (main memory or side memory) to process a given query through, thus allowing the model to edit and update new knowledge without destroying the original pre-trained knowledge. GRACE [233] implements model editing by adding an adaptor to a specific layer of a pre-trained model, where the adaptor contains a discrete codebook and a deferral mechanis. The keys of the codebook cache the activations passed in from the previous layer, and each key maps to a corresponding value. When new editing requirements arise, GRACE adapts to the new changes by updating the keys and values in the codebook without adjusting the weights of the model.

Guided by the new knowledge, **Global optimization**-based methods incorporate the new knowledge into LLMs by *updating all parameters*. To preserve original knowledge, PPA [234] utilizes Low-Rank Adaptation (LoRA) in feed forward layers of the transformer decoder. It makes use of plug-in modules that are trained with the constrained optimization of LoRA and evaluate if the input content is relevant to in-scope input space using the K-adapter [235]. Similarly, ELDER [236] utilizes a series of Mixture-of-LoRA components, which dynamically assign LoRAs to continual editing tasks, ensuring a continual link between the data and the adapters. In the mixture-of-LoRA module, each edit is routed to top-k LoRAs with the highest scores based on its query vector.

Local modification-based methods *locate relevant parameters* for specific knowledge in LLMs and update them. To address the problem of toxicity buildup and toxicity flash during editing, WilKE [237] assesses the degree of pattern matching between different layers for editing knowledge, and then selects the most appropriate layer for knowledge editing based on the degree of pattern matching. The essence of PRUNE [238] lies in its approach to managing the condition number of a matrix during continual editing processes. By diminishing the large singular values [239], [240] of the edit update matrix, it effectively decreases the upper bound on perturbation to the matrix, which maintains the general capabilities of the model while integrating new editing knowledge.

9.3 Continual Alignment

In terms of continual alignment for agents, they fine-tune their internal model parameters by continuously absorbing

TABLE 4
Comparison between preference optimization techniques for one-step continual alignment.

Research	Loss Function	Note
DPO [223]	$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)\right]$	-
IPO [224]	$\mathcal{L}_{\text{IPO}}(r_\pi,x,y_w,y_l,\pi_t) = \left(r_\pi(x,y_w) - r_\pi(x,y_l) - \tau^{-1}\right)^2$	For each iteration $t \in [T]$, it designates the reference policy as the policy generated from the previous iteration, denoted by π_t. $r_\pi(x,y) = \beta\log\frac{\pi(y\|x)}{\pi_t(y\|x)}$. τ is a regularization parameter.
EPO [225]	$\mathcal{L}_{\text{EPO}}(\theta) = \mathbb{E}_{(T,p_w,p_l)\sim\mathcal{D}}\left[-p_w\log(\pi_\theta(\hat{p}\mid T)) + \mathcal{L}_{\mathcal{D}}\right]$	$\mathcal{L}_D = -\mathbb{E}_{(T,p_w,p_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(p_w\|T)}{\pi_{\text{sup}}(p_w\|T)} - \beta\log\frac{\pi_\theta(p_l\|T)}{\pi_{\text{sup}}(p_l\|T)}\right)\right]$. π_{sup} denotes the LLM learned from the annotated dataset. $\hat{p}$ denotes the logits of the model output tokens.
KTO [226]	$\mathcal{L}_{\text{KTO}}(\pi_\theta,\pi_{\text{ref}}) = \mathbb{E}_{x,y\sim\mathcal{D}}[\lambda_y - v(x,y)]$	λ_y denotes $\lambda_D(\lambda_U)$ when y is desirable(undesirable) respectively. $r_\theta(x,y) = \log\frac{\pi_\theta(y\|x)}{\pi_{\text{ref}}(y\|x)}$. $z_0 = \text{KL}(\pi_\theta(y'\|x)\|\pi_{\text{ref}}(y'\|x))$. If $y \sim y_{\text{desirable}}\|x, v(x,y) = \lambda_D\sigma(\beta(r_\theta(x,y) - z_0))$. If $y \sim y_{\text{undesirable}}\|x, v(x,y) = \lambda_U\sigma(\beta(z_0 - r_\theta(x,y)))$.
DMPO [227]	$\mathcal{L}_{\text{DMPO}} = -\mathbb{E}_{(s_0,\tau^w,\tau^l)\sim D}\log\sigma\left[\sum_{t=0}^{T_w-1}\beta\phi(t,T_w)\log\frac{\pi_\theta(a_t^w\|s_t^w)}{\pi_{ref}(a_t^w\|s_t^w)} - \sum_{t=0}^{T_l-1}\beta\phi(t,T_l)\log\frac{\pi_\theta(a_t^l\|s_t^l)}{\pi_{ref}(a_t^l\|s_t^l)}\right]$	The discount function $\phi(t,T) = (1-\gamma^{T-t})/(1-\gamma^T)$.
ORPO [228]	$\mathcal{L}_{\text{ORPO}} = \mathbb{E}_{(x,y_w,y_l)}\left[\mathcal{L}_{SFT} + \lambda\cdot\mathcal{L}_{OR}\right]$	$\mathcal{L}_{SFT}$ denotes supervised fine-tuning loss. $\mathcal{L}_{OR} = -\log\sigma\left(\log\frac{\textbf{odds}_\theta(y_w\|x)}{\textbf{odds}_\theta(y_l\|x)}\right)$.
ROPO [229]	$\mathcal{L}_{\text{ROPO}} = \frac{4\alpha}{(1+\alpha)^2}\cdot\mathcal{L}_{\text{DPO}} + \frac{4\alpha^2}{(1+\alpha)^2}\cdot\mathcal{L}_{\text{reg}}$	$\mathcal{L}_{\text{reg}} = \sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_2\|\mathbf{x})}{\pi_{ref}(\mathbf{y}_2\|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_1\|\mathbf{x})}{\pi_{ref}(\mathbf{y}_1\|\mathbf{x})}\right)$.
CDPO [230]	$\mathcal{L}_{\text{CDPO}} = -\mathbb{E}_{(x,c,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\hat{R}_\theta(c,x,y_w) - \hat{R}_\theta(c,x,y_l)\right)\right]$	$\hat{R}_\theta(c,x,y_w) = \beta\log\frac{\pi_\theta(y_w\|c,x)}{\pi_{\text{ref}}(y_w\|c,x)}$. $\hat{R}_\theta(c,x,y_l) = \beta\log\frac{\pi_\theta(y_l\|c,x)}{\pi_{\text{ref}}(y_l\|c,x)}$

human feedback and preferences. The *alignment tax* [241] refers to the trade-off between aligning models with human values and the potential compromise in their overall performance, leading to a reduction in the general capabilities of the model and making it a crucial factor to consider in this process. This fine-tuning not only enhances the responses of the agents to new commands but also allows them to absorb and integrate new information without forgetting previous knowledge through continuous learning. This dynamic parameter adjustment process is essentially the manifestation of the parametric memory updating, enabling them to learn and adapt in each interaction, thus achieving lifelong learning and avoiding catastrophic forgetting.

Traditional alignment goes through a *one-step* process [52] in which the dataset usually consists of a fixed set of static examples. This alignment usually uses *preference optimization* techniques, as illustrated in Table 4. This one-step process allows the model to learn the nuances of a particular task in depth, but may lack the ability to adapt to new situations.

Compared to it, *multi-step* alignment demands that the model needs to adapt to new tasks without forgetting previously learned tasks, which is the challenge of continual alignment. For example, in the previous task, a model is aligned to provide helpful and harmless responses. When asked "Are people with mental illness crazy?", the model answers no, emphasizing that mental illness is a medical condition that affects the thinking, feelings, and behavior of an individual. This indicates that the model emphasizes avoiding offensive or inaccurate statements at this stage. In the current task, the model is aligned to provide concise and organized responses. When asked "Why is investigating lifelong learning of large language models important?", the model lists several key reasons such as adaptability, efficiency, robustness and personalization. This shows that the model has been adapted at this point to focus more on the structure and organization of the responses. In the next task, the model is aligned to express positive sentiment. When asked "Do AI technologies pose a threat to humanity?", the model responds that when developed and implemented thoughtfully, AI technologies offer incredible opportunities, not threats. This suggests that the response of the model takes a positive angle, highlighting the potential and benefits of AI and avoiding the negativity of overemphasizing threats. Overall, the process involves gradual adjustments to the model to ensure that its outputs are appropriate in terms of ethics and social responsibility. Each task has different alignment standards for the model, and at each step the model learns how to better align with human values, ultimately achieving lifelong learning.

In scenarios of continual alignment, datasets reflecting human preferences are in a state of constant flux, typically spanning across multiple tasks or domains. In order to solve the problem that full retraining of RLHF-based language models consumes a lot of time and computational resources, COPR [242] computes the sequence of optimal policy distributions without relying on the partition function and subsequently regularizes the current policy according to the historically optimal distribution, which finetunes the policy model and reduces the occurrence of catastrophic forgetting. By maintaining a scoring module, COPR offers adaptability in lifelong learning scenarios where human preferences evolve continuously without the need for human feedback. Similarly, to address the constraints of time costs and data privacy concerns, CPPO [243] introduces a weighting strategy that distinguishes between the rollout samples used to augment strategy learning and to consolidate past experience. CPPO classifies the samples into five types based on their reward and generation probabilities, and then assigns different policy learning weights and knowledge retention weights, which continually aligns language models with dynamic human preferences.

9.4 Summary

Considering the differences between the four types of memory modules, working memory is the short-term memory of the agent, while episodic memory stores long-term experiences and semantic memory stores external world knowledge. Unlike the other three types of memory modules that explicitly store experiences or knowledge, parametric memory is the most abstract form of memory in LLMs, which is shaped by training data and fine-tuning processes through knowledge encoded in the internal parameters of the model. This memory is not explicitly accessible or retrievable like other memories, but it enables the model to understand and make inferences, and to adapt to new situations based on accumulated knowledge.

In the process of continual instruction tuning, the parametric memory of LLMs is updated through the continuous use of instruction datasets to adjust the implicit parameters

of the model. This is an ongoing process that allows the model to continuously improve general or specific capabilities for lifelong learning. In the process of continual knowledge editing, the agent continuously uses the updated dataset to modify errors or outdated information implicit in the previous model. This process not only prevents the loss of old knowledge due to new learning (catastrophic forgetting), but also realizes lifelong learning for the agent, allowing it to adapt to changing environments. Continual alignment involves the agent fine-tuning its internal model parameters by continuously absorbing human feedback and preferences. In contrast to traditional one-step alignment, multi-step alignment requires the model to adapt to new tasks in the presence of changing datasets reflecting human preferences, which is the challenge of continual alignment.

10 Action Design: Grounding Actions

In Table 5, we provide an overview of research aimed at improving the quality of LLM agent actions. This subsection focuses on grounding actions, which involve perceiving the environment through textual descriptions and generating text to determine the most suitable subsequent actions [274]. We begin by outlining the key challenges lifelong LLM agents face in executing these grounding actions, then explore corresponding solutions under different environmental contexts.

10.1 Challenges of Grounding Actions

As the brain of agent, LLM is responsible for taking textual observations from the environment as input and generating textual actions as output. Similar to [64], we define *input grounding actions* as the process of perceiving and understanding textual environment description, and *output grounding actions* as generating text that can be parsed into actions by environment. The generation process of a grounding action at time t can be formulated as:

$$ra_t, a_t = \mathcal{G}(\mathcal{E}, g, t) = \begin{cases} \mathcal{G}(\mathcal{E}, g, o_0), & \text{if } t = 0 \\ \mathcal{G}(\mathcal{E}, g, \xi(t)), & \text{otherwise.} \end{cases} \tag{7}$$

where:

- ra_t is the rationale corresponding to the generated action.
- a_t is the action generated at time t, which can be viewed as the result of output grounding action.
- $\mathcal{G}$ is the generative LLM backbone.
- g is the goal defined in definition 3.2.
- $\mathcal{E}$ is the environment defined in definition 3.3. Here we omit the task index for simplification. The environment is presented in textual form, serving as the primary focus for input grounding actions to interpret and process.
- o_0 is the initial observation.
- $\xi(t)$ is the trajectory defined in definition 3.5 up to time t.

10.1.1 Input Grounding Actions

For *input grounding actions*, there are notable differences between the textual formats encountered in the pretraining corpus and those used for environment descriptions. While the pretraining corpus primarily consists of well-structured paragraphs, environment descriptions often take the form of brief sentences, short phrases, or structured text formats such as JSON strings or HTML tags. As a result, LLM must adapt from the familiar input formats of its pretraining data to the diverse and specialized formats found in the agent's environment. In rapidly changing environments, the agent needs to adapt to the updating descriptions continuously to get better understanding of the environment.

10.1.2 Output Grounding Actions

For *output grounding actions*, there are significant differences in the types of content LLM is required to generate. During pretraining, LLM is primarily trained for simple text completion, but in an agent's environment, it must generate text that adheres to specific patterns, representing actions or environment-specific elements. LLM must learn to perform complex actions by producing outputs tailored to the requirements of the environment, rather than merely describing actions or intents in free-form natural language. Furthermore, in complex environments, the requirements for output grounding actions may change based on the agent's previous actions, necessitating continuous adaptation to align with the evolving demands of the environment.

10.1.3 Summary

The differences between input and output grounding actions highlight the critical importance of lifelong learning in LLM agents. LLM needs to adapt its input and output grounding actions, originally designed for the pretraining phase, to align with the specific requirements of different environments. Furthermore, it also needs to continuously adjust these actions to effectively respond to ever-changing environment.

10.2 Solutions for Different Environments

LLM agents with lifelong learning capabilities can not only adapt their grounding actions from the pretraining phase to suit specific environments but also continuously evolve through interactions with their surroundings. However, the variability across environments presents unique challenges, prompting the development of diverse solutions in existing research. To offer a clear and comprehensive overview of these solutions, we classify commonly used environments into three categories, ranging from simple to complex: **tools**, **web**, and **game**.

10.2.1 Tool Environment

Tool refers to an external functionality or resource that agent can interact with to enhance its capabilities. Calculator, calendar, search engine [253], and Application Programming Interface (API) [245] can all be viewed as tools. Tools can enhance LLM agent expertise in specific domains and provide better interpretability to agent decision process. In tool environment, an LLM agent must first understand the tools' functionalities and then invoke the appropriate tools in the correct order based on user intents.

The *input grounding actions* of an LLM agent in a tool environment primarily involve understanding tool documentation. Complex tools are often encapsulated as APIs

TABLE 5
Summary of research that focus on improving the quality of the LLM agent's actions. Focus: Categories of actions targeted by the research. Grounding: Grounding Actions. Retrieval: Retrieval Actions. Reasoning: Reasoning Actions.

Research	Contribution	Focus		
		Grounding	Retrieval	Reasoning
GEAR [244]	Use small language models to select tools for LLM.	✓		
ToolLLM [245]	Introduce a tool learning dataset and a tool searching algorithm.	✓	✓	✓
EASYTOOL [246]	Employ ChatGPT to transform tool documents into concise instructions.	✓		✓
ART [247]	Revise tool calling trajectories manually and store them for future use.	✓	✓	
Xu et al. [248]	Introduce a tool learning dataset.	✓	✓	
STE [249]	Generate tool learning data that can fully represent the usage of tools.	✓	✓	
Confucius [250]	Employ curriculum learning to help LLM understand difficult tools.	✓		
LATM [251]	Employ GPT-4 to make tools to solve similar problems.	✓	✓	
ToolkenGPT [252]	Enable LLM to use tools by only modifying its output embedding layer.	✓		
Toolformer [253]	Introduce the concept of tool learning.	✓		
ToolNet [254]	Organize tools into a directed graph to guide LLM's tool selection.			✓
α-UMi [255]	Split the tool calling process into three parts, each handled by an LLM.			✓
SteP [256]	Compose handcrafted policies dynamically to solve web tasks.	✓		✓
AgentOccam [64]	Modify web LLM's action and observation space to improve performance.	✓		
WebPilot [257]	Propose a multi-agent system combining MCTS for web environments.	✓		✓
RLEM [258]	Propose a framework allowing LLM to use experience from multiple tasks.	✓		✓
LASER [259]	Model the interactive web tasks as state-space exploration.	✓		✓
ICAL [260]	Construct multimodal memory from sub-optimal demonstrations.	✓	✓	✓
Synapse [65]	Simplify webpages to include full trajectories in prompts.	✓	✓	
DEPS [75]	Propose an LLM based planning framework in Minecraft.	✓		✓
JARVIS-1 [73]	Propose an multimodal agent in Minecraft.	✓	✓	
VillagerAgent [74]	Introduce a multi-agent benchmark and framework in Minecraft.	✓		✓
STEVE [261]	Propose a visionary agent in Minecraft.	✓	✓	✓
Cradle [262]	Propose a VLM framework interacting with software via a unified interface.	✓	✓	✓
Voyager [61]	Propose the first lifelong LLM agent in Minecraft.	✓	✓	✓
GITM [62]	Propose framework integrating LLMs with text memory in Minecraft.	✓	✓	✓
Huang et al. [263]	Employ demonstrations to translate plans to admissible actions.		✓	✓
MemoryBank [55]	Propose a LLM memory system allowing continuous memory updates.		✓	
Tree of Thoughts [264]	Arrange problem solving intermediate steps generated by LLM in a tree.			✓
ReAct [194]	Prompt LLM to generate interleaved verbal reasoning traces and actions.			✓
Reflexion [58]	Reinforce LLM reasoning through linguistic feedback from previous trials.			✓
RAP [265]	Propose a framework using LLMs as world model and reasoning agent.			✓
LLM-MCTS [266]	Employ LLM as both world model and policy that acts on it.			✓
SwiftSage [267]	Propose an efficient LLM reasoning framework imitating human cognition.		✓	✓
API-Bank [268]	Introduce a tool learning dataset and a data generation method.			✓
ADaPT [269]	Propose an LLM reasoning algorithm decomposing sub tasks recursively.			✓
Tree-planner [270]	Employ tree structure to reduce token consumption of LLM reasoning.			✓
ToolEVO [271]	Inprove the adaptability of LLM in dynamic tool environment.			✓
DEER [272]	Enable LLM to invoke tools only when appropriate tools are available.			✓
Raman et al. [273]	Employ error information to enhance LLM planning ability.			✓

and described using JSON strings [245], [248], [268], [275]–[277]. To fully understand these tools, LLM must adapt to the unique formats of tool documentation.

One approach to address this challenge is simplifying tool documentation directly within the prompt, enabling LLM to focus on essential information. For instance, EASYTOOL [246] uses ChatGPT to transform verbose and redundant tool documentation into concise instructions for easier comprehension. Similarly, GEAR [244] employs a carefully designed framework composed of smaller language models

to select tools for LLM, narrowing its focus to specific tool documentation. Another common strategy involves leveraging tool calling trajectories for fine-tuning or as in-context learning demonstrations, helping LLM better understand tools [245], [248], [249].

Furthermore, ART [247] proposes manually revising tool calling trajectories generated by LLM and storing them in a database for future use. Confucius [250] also utilizes tool calling trajectories to fine-tune LLM, iteratively updating the training dataset based on LLM's performance with various tools. Both approaches adopt a lifelong learning perspective to refine LLM's output grounding actions. By utilizing previously generated tool calling trajectories, they enhance the agent's performance in subsequent tool interactions, enabling continuous improvement over time.

LLM also need to adapt its *output grounding actions* to tool environments so that it can output text in particular format to use tool and pass parameters. Almost all works force LLM to generate text in particular format by fine-tuning [253] or few-shot learning [246]. ToolkenGPT [252] stands out among these works by introducing a special token in the output embeddings of LLMs, enabling the models to invoke tools. By fine-tuning only the output embeddings corresponding to this token, it can call tools while maximizing the preservation of its language modeling capabilities.

The works mentioned above focus on improving LLM's tool-calling abilities in static tool environments. Recent studies, however, explore how to enhance LLM's tool-calling capabilities in lifelong tool environments. A lifelong tool environment is the tool environment where LLM must continually adapt its grounding actions to changing tools. For example, STE [249] finds out that a simple replay strategy can effectively mitigate catastrophic forgetting during continual fine-tuning. Similarly, Chen et al. [271] focus on fine-tuning LLM to overcome the difficulties brought by outdated tool documentations. In contrast to these approaches, LATM [251] emphasizes constructing a lifelong tool environment itself to better fulfill user needs. It employs GPT-4 to develop tools for handling increasing user requests and GPT-3.5 to effectively utilize these tools.

10.2.2 Web Environment

In web environments, LLM-based agents must engage with webpages based on user intent. Unlike humans, LLM perceive webpages primarily through the HTML DOM tree [278] or the accessibility tree [21]. These formats are not only lengthy but also fail to intuitively display webpage content, posing significant challenges to the *input grounding actions* of LLM. To address this, works such as AgentOccam [64] and Synapse [65] focus on simplifying webpages before including them in prompts, improving the accuracy of input grounding actions. Additionally, studies like WebPilot [257], RLEM [258], ICAL [260], AWM [10] and Synapse [65] incorporate previous trajectories or episodic experiences into prompts, enabling LLM to enhance their input grounding actions and continuously improve their understanding of webpage content during browsing.

Compared to the tool environment, it is more challenging for agent to generate high quality *output grounding actions* in web environment. The agent may be confused by the web interaction actions that are irrelevant to current webpages or hard to use. SteP [256] and LASER [259] propose removing irrelevant actions description from the prompt. Likewise, AgentOccam [64] proposes only including actions that can be grasped by LLM in the prompt.

10.2.3 Game Environment

Game environment is the most complex environment among these three kinds of environments. Given that LLM embodied agents usually operate in virtual environments, we classify their working environment as a game environment. Based on the APIs provided by different game environments [279], [280], the specific requirements for LLM's input grounding actions and output grounding actions vary across environments. For *input grounding actions*, DEPS [75], JARVIS-1 [73], and VillagerAgent [74] use specialized prompts to help LLM gain a deeper and clearer understanding of the environment. Alternatively, another line of research [261], [262] treats the environment as an image and employs Virtual Language Models to directly perceive the complex environment. These approaches help reduce the information loss that can occur when describing complex environments using natural language. For *output grounding actions*, most works enable LLM to interact with the environment by generating executable programs. These executable programs can be used to directly control the agent's behavior [61], [261] or indirectly control it by being mapped to keyboard or mouse operations [62], [262].

Due to the complexity of the game environment and its rapidly changing nature, many works enhance the long-term consistency of agent behavior and the overall capabilities of the agent from the perspective of lifelong learning. Voyager [61] is the first LLM-powered embodied lifelong learning agent in Minecraft. It can generate skills to solve current task and store them in skill library to solve subsequent tasks. What's more, JARVIS-1 [73], STEVE [261] and Cradle [262] use additional memory modules to store previous experience, prompting better plans generation in subsequent interactions.

10.3 Summary

In this subsection, we categorize environments into three categories and discuss how LLM agents can adapt to each of them. It is important to note that while some methods are specific to certain environments [12], most of these methods can be transferred to different environments. For example, the idea of making executable programs to continuously improve agent's ability [251] can also be applied in web environment. In such a case, the agent can generate subsequent plans or actions after executing an executable program, rather than after each simple mouse/keyboard action (e.g., clicking a button), thereby reducing the length of the action history and observation history, which improves the long-term consistency of the agent's behavior. Different approaches can also be combined to enhance the lifelong LLM agent's performance.

11 Action Design: Retrieval Actions

LLM agent needs *external information* to generate high quality grounding actions and reasoning actions. For grounding

actions, LLM's generation can be parsed into actions only if it matches the patterns specified by the environment. Simply fine-tuning LLM to force it generate outputs that conform to environmental constraints is impractical, as this approach not only incurs significant resource costs but also fails to adapt to the continuously changing action space. Moreover, directly including all possible action descriptions in the prompt is unfeasible due to the unacceptable context length and the *lost in the middle* problem [281]. For reasoning actions, comprehensive external knowledge from semantic memory and accurate historical trajectory from episodic memory are key factors in making correct decisions. However, text at the scale of the internet cannot be fully included in the context length of an LLM. The lengths of action history and observation history will also gradually increase with the agent's activities, eventually exceeding the context length of LLM.

These challenges highlight the critical importance of retrieval actions for lifelong LLM agents. With the retrieval actions, LLM agent can handle the continuously growing actions history and observation history, which helps maintain the long-term consistency of the LLM's behavior. It can also gain real-time knowledge by retrieving from continuously refreshed knowledge sources, leading to better performance in environment changing continuously [282]. From the lifelong learning perspective, the agent can also retrieval the knowledge gained in previous task to improve its performance on current task. Moreover, Retrieval action can also improve the interpretability of the agent and its performance on knowledge-intensive task [283]. We classify the work that allow the agent to retrieval from memory into two parts based on the retrieval source, which are *semantic memory* and *episodic memory*. The retrieval and utilization process of semantic memory can be formulated as bellow. LLM agent can perform the retrieval action only at the beginning of a trial or before generating each action. Here, we assume that the LLM agent performs the retrieval action only at the beginning of a trial, which is a common practice in existing research [62], [273].

$$m_{\mathcal{T}^{(i)}} = \text{retrieve}(\mathcal{T}^{(i)}, \mathcal{M}^{(i)}) \tag{8}$$

$$ra_t^{(i)}, a_t^{(i)} = \mathcal{G}(\mathcal{E}^{(i)}, m_{\mathcal{T}^{(i)}}, g^{(i)}, t) \tag{9}$$

where:

- retrieve denotes the retrieval process.
- $\mathcal{M}$ is the pre-defined, static database.
- $m_{\mathcal{T}^{(i)}}$ is the retrieval memory item corresponding to task $\mathcal{T}^{(i)}$, which can be background knowledge or demonstrations.

The retrieval and utilization process of episodic memory is similar to that of semantic memory, while the updating process for episodic memory can be outlined as follows.

$$\mu^{(i)} = \text{process}(\mathcal{E}^{(i)}, \xi_{T_i}^{(i)}) \tag{10}$$

$$\mathcal{M}^{(i+1)} = \text{update}(\mathcal{M}^{(i)}, \mu^{(i)}) \tag{11}$$

where:

- $\mathcal{M}^{(i)}$ is the dynamic database up to task $\mathcal{T}_i$.
- process denotes the abstraction and transformation of the agent's trajectory, which can be performed by either an LLM [65] or a human [260].
- $\mu^{(i)}$ represents the processed memory with diverse forms. It could be a corrected or simplified trajectory or a textual summary of the task.

In Table 6, we summarize the classification results.

11.1 Retrieval from Semantic Memory

Pretrained LLM is often insufficient as an agent's brain due to two key limitations: the lack of background knowledge and the lack of demonstrations. These two limitations can be addressed through retrieving from semantic memory, which is an external memory mechanism storing world knowledge. See Section 8 for further explanation.

11.1.1 Lack of Background Knowledge

The *lack of background knowledge* is typically manifested in LLM's inability to select the correct action from all possible actions or to generate actions that can be understood by the environment. To address this issue, GITM [62] retrieves relevant text from Minecraft Wiki to provide LLM with the Minecraft world knowledge, enabling LLM to execute actions in a correct order. Both SwiftSage [267] and ToolLLM [245] use SentenceBERT [285] to retrieve possible actions from a database, helping LLM select the appropriate actions by narrowing the action space. When the parameters of the action are finite, SentenceBERT can further be used to transform action parameters generated by LLM, which cannot be understood by the environment, into valid parameters [263], [266].

11.1.2 Lack of Demonstrations

The *lack of demonstrations* can diminish the quality of the agent's grounding actions and planning actions. Demonstrations have been proved to play an important role in LLM performance [11]. However, including irrelevant or outdated demonstrations in the prompt may greatly compromise the LLM agent's performance [271]. To verify relevant issues, Re-Prompting [273] and STE [249] use SentenceBERT to select most similar demonstrations from a demonstrations set.

11.2 Retrieval from Episodic Memory

While retrieving from semantic memory can improve agent's ability by providing additional background knowledge and demonstrations, it can neither address the lack of ability to leverage past experiences, nor the lack of long-term consistency in LLM. However, these two limitations can be addressed through retrieving from episodic memory. Different from semantic memory, episodic memory mainly stores past experience. See Section 7 for further explanation.

11.2.1 Lack of Ability to Leverage Past Experiences

Overcoming the *lack of ability to leverage past experiences* is a key characteristic of a lifelong LLM agent. Leveraging past experience, LLM agent can gradually improve itself during interactions with the environment. Current research utilizing past experiences can be broadly classified into two

TABLE 6
Summary of research that focus on enhancing the retrieval actions of LLM agent. This table categorizes research based on the source of retrieval and their primary focus. For retrieval from semantic memory, research primarily aims to provide additional background knowledge or demonstrations as supplementary environmental information. For retrieval from episodic memory, the focus is on improving the agent's ability to leverage past experiences and enhancing its long-term consistency.

Retrieval Source	Focus	Research
Semantic Memory	Background Knowledge	GITM [62], SwiftSage [267], ToolLLM [245], Huang et al. [263], LLM-MCTS [266], AMOR [284]
	Demonstrations	Re-Prompting [273], STE [249]
Episodic Memory	Ability to leverage past experiences	ICAL [260], GITM [62], Voyager [61], LATM [251], JARVIS-1 [73], Cradle [262], ART [247], Xu et al. [248], Synapse [65]
	Long-Term Consistency	MemoryBank [55], Cradle [262]

categories, both aimed at enhancing the agent's capabilities from a lifelong learning perspective. The first category involves storing the agent's trajectory in a database after it completes a task successfully [62], [260]. When a new task arises, the agent retrieves similar task trajectories from the database and incorporates them into the prompt. This category of research improve the quality of agent's reasoning actions. The second category focuses on representing the agent's task-solving steps as executable programs [61], [73], [251], [262]. These programs are primarily composed of the acceptable actions defined by the environment and can be seen as high-level actions generated by the agent. When facing a new task, the agent can either directly reuse these programs or combine them to address the new challenge. This category research expands the action space of the environment and improves the grounding ability of agents.

11.2.2 Lack of Long-Term Consistency

The *lack of long-term consistency* primarily arises from the fact that the context length of LLMs is finite, preventing them from incorporating the entire observation and action history into the prompt. Improving the long-term consistency can make the agent more similar to humans. MemoryBank [55] not only retrieve the summary of past conversations to keep current consistent with the chatting history, improving the performance of LLM in lifelong interaction scenarios.

11.3 Summary

In this subsection, we discuss the crucial role of retrieval actions for lifelong LLM agents. Most existing LLM agent research only focuses on retrieval from either semantic memory or episodic memory. However, it is worth noting that retrieving from both semantic and episodic memory can further improve the agent's performance. This approach mirrors how humans use long-term and short-term memory. Additionally, lifelong LLM agent can also leverage well-established techniques from the RAG field, such as iterative retrieval [286]–[288] or constructing higher-quality retrieval sources continuously [289], to improve the quality of the retrieved content.

12 Action Design: Reasoning Actions

The final category of actions is reasoning actions. An LLM agent can gain deeper insights from the content loaded into its working memory. After pretraining, the LLM is capable of performing basic reasoning using techniques such as *chain-of-thought* [218]. However, its reasoning ability is insufficient to handle more complex reasoning tasks within the agent's environment. There are two primary reasons for this limitation. First, the environment in which the LLM operates is relatively complex, making it difficult for the LLM to perform high-quality reasoning based solely on the knowledge acquired during pretraining. Second, the reasoning capabilities of the LLM itself are inherently limited. For instance, a pretrained LLM is unlikely to recognize errors in prior reasoning steps without techniques like reflection, resulting in incorrect conclusions.

To address these challenges, numerous research on LLM agents have focused on improving the quality of reasoning actions through carefully designed prompts or novel frameworks. Some of this research addresses the issue from the perspective of lifelong learning, allowing the LLM to refine its current reasoning behavior by building upon prior reasoning outcomes or trajectories. These approaches enable the agent to improve its reasoning ability gradually across different trials within a single episode, or over longer periods across multiple episodes. Based on this, we classify the agent's reasoning actions into *intra-episodic reasoning actions* and *inter-episodic reasoning actions*. We summary the classification result in Table 7.

12.1 Intra-Episodic Reasoning Actions

Intra-episodic reasoning actions refer to reasoning actions that leverage the experiences within the same episode. Based on whether these research stimulate the LLM's intrinsic reasoning ability within the same trial or progressively enhance its reasoning ability across different trials, we further categorize the articles into two groups.

12.1.1 Single Trial

On the one hand, nearly all research encourages the LLM agent to perform reasoning in the ReAct [194] style *within a single trial*. ReAct allows the LLM to continuously refine its reasoning process based on real-time feedback from the environment, ultimately leading to more accurate plans or decisions. On the other hand, many research break down reasoning into several key steps, assigning different LLMs to handle each step. For instance, α-UMi [255] fine-tunes

TABLE 7
Summary of research that focuses on enhancing the reasoning actions of LLM agent. This table categorizes research based on time span. For research that improve agent reasoning ability within a single episodic, we further divide them into two categories: those that stimulate the LLM's intrinsic reasoning ability within the same trial and those that progressively enhance reasoning ability across multiple trials.

Focus		Research
Intra-Episodic	Single Trial	ReAct [194], α-UMi [255], API-Bank [268], LASER [259], SteP [256]
	Across Trials	Reflexion [58], Tree of Thoughts [264], Tree-planner [270], RAP [265], LLM-MCTS [266], WebPilot [257], ToolEVO [271]
Inter-Episodic		ICAL [260], GITM [62], Raman et al. [273], Voyager [61], Cradle [262], MemoryBank [55], STEVE [261]

two LLMs, with one responsible for planning and the other for summarization. Similarly, API-Bank [268] employs five LLMs to handle each state of the reasoning process, ultimately generating high-quality tool-learning training data. These research improve the reasoning action quality of an agent or an agent system.

Additionally, many research enhance the reasoning ability of LLM agents in complex environments by introducing environment-specific strategies. For example, LASER [259] models the process as state-space exploration and reduces the difficulty of reasoning by only allowing the LLM to transition between adjacent states. Similarly, SteP [256] enhances the LLM's reasoning ability in complex environments by dynamically composing handcrafted policies, with each policy serving as a method for extracting information from a specified complex environment.

12.1.2 Across Trials

Many research further improve the agent reasoning ability *across different trials* by mimicking human reasoning process. These research enable agent to make use of experience not only from current trial, but also from previous trials. Agents are also instructed to perform reasoning in ReAct style. Reflexion [58] is one of the most representative research. It enables LLM to engage in self-reflection by reviewing past failed trials. This reflective process helps the agent refine its reasoning, ultimately leading to more accurate results in future trials. The action generation process when Reflexion is applied can be formulated as bellow. LLM agent will reflect on its past trajectories at the start of each new trial, repeating this process as needed until it successfully achieves the given goals.

$$\text{reflection} = \text{reflect}(\widetilde{\xi}) \tag{12}$$

$$ra_t, a_t = \mathcal{G}(\mathcal{E}, g, \text{reflection}, t) \tag{13}$$

where:

- $\widetilde{\xi}$ is the previous full-length trajectories of the past trials.
- reflect denotes the process of extracting valuable insights from past trials.
- reflection is the verbal experience feedback for past trials.

Another line of research make use of the incomplete trails. These research typically use tree structures to manage each incomplete trial. During the reasoning process, each path from the root node to a leaf node can be viewed as an incomplete trial. With the help of the tree structure, LLM agent can perform higher-level operations such as lookahead and backtracking, continuously improving the quality of reasoning actions across multiple trials [264], [270]. Also thanks to the tree structure, many classical tree search algorithms, such as Breadth-first search, Depth-first search [264], and Best-first search [290], can be seamlessly integrated into the reasoning process. Furthermore, many research use Monte Carlo Tree Search (MCTS) to enhance the reasoning ability of agents operating in complex environments [257], [265], [266], [271]. MCTS enables the agent to more effectively explore and exploit the complex environment, ultimately leading to the formulation of a correct plan through reasoning actions.

12.2 Inter-Episodic Reasoning Actions

Inter-episodic reasoning actions refer to reasoning actions that use the experiences from different episodic. These experiences accumulate gradually as the LLM agent interacts with the lifelong environment. Experiences can take many forms, such as successful reasoning trajectories [62], [260], [273], executable code [61], or textual summary [55], [262]. These experiences are typically stored in an additional database. When encountering a new task, the LLM agent first retrieves the most relevant experiences and add them to the working memory, then uses these experiences to inform its reasoning process. Specially, AMOR [284] leverages feedback from previous tasks to fine-tune the model, improving its reasoning actions quality under specific environments.

Additionally, some research focus on how to enable LLM agent to explore complex environment employs curriculum learning to arrange different tasks. Additionally, some research focus enabling LLM agent to explore complex environment employ curriculum learning to better make use of the past experiences. These research [61], [261] arrange the task from easy to difficult, providing a steady stream of new tasks or challenges. Through curriculum learning, LLM agent can gradually master reasoning action techniques throughout the lifelong exploration process.

12.3 Summary

In this subsection, we discuss the primary method of improve the reasoning ability of LLM agent in lifelong environment. Current research mostly employ experience from previous trails or episodic to improve the reasoning ability of agent continuously. However, existing research, particularly

that which uses tree structures [264], [270], [286], [290], often overlooks the dynamic nature of complex environments and assumes that any state in the online reasoning process is retraceable. We would like to point out that this is unrealistic in practice. For instance, in a web environment, payments can be irreversible. Future research should explore how to facilitate the lifelong accumulation of reasoning techniques in more realistic settings, such as those characterized by irreversible actions.

13 Evaluation of Lifelong LLM Agents

Evaluation metrics, datasets, and benchmarks are critical components of research on lifelong LLM agents. Compared to traditional LLM agents, lifelong LLM agents can retain useful historical experiences through interactions with the environment. These experiences enable LLM agents to achieve better performance on both past and future tasks. As a result, many evaluation metrics are designed to demonstrate the superiority of lifelong learning methods by measuring the LLM's performance on tasks. However, to effectively compare different lifelong learning methods, having robust evaluation metrics alone is insufficient. Reliable datasets and highly realistic and reproducible benchmarks also indispensable. In this section, we will briefly introduce the widely used evaluation metrics in one subsection, followed by a discussion of the datasets and benchmarks in another.

13.1 Evaluation Metrics

The evaluation metrics of lifelong LLM agents primarily assess their lifelong learning ability from three perspectives: (1) the overall accuracy of all tasks, (2) the stability of history tasks (3) the plasticity of future tasks. The performance of LLM agent can be measured by different metrics from different aspects, such as accuracy, pass rate, and win rate [245]. Here, we assume that the agent's performance can be converted into a scalar obtained by aggregating various metrics. Following 3.1, we denote the performance of the LLM agent on task i as $J_{i,t}$ after completing t tasks [1].

There are two evaluation metrics that measure the lifelong learning ability of LLM agent from the first perspective, which are *average performance* (AP) and *average incremental performance* (AIP). The difference between AP and AIP is that, AIP captures the historical variation when experiencing each task.

$$\mathrm{AP}_t = \frac{1}{t}\sum_{i=1}^{t} J_{t,i}, \tag{14}$$

$$\mathrm{AIP} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{AP}_t \tag{15}$$

Other two metrics, *forgetting measure* (FGT) and *backward transfer* (BWT) measure lifelong learning ability of LLM agent from the perspective of stability. FGT evaluates the average accuracy drop of each old task, representing that whether useful experience are maintained successfully. BWT evaluates the average accuracy improvement of each old tasks, representing whether the experience gained after the LLM agent encountered a given task benefits that task.

$$\mathrm{FGT}_t = \frac{1}{t-1}\sum_{i=1}^{t-1}[\max_{j\in\{i,i+1,\cdots,t\}}(\{J_{j,i}\}_j) - J_{t,i}], \tag{16}$$

$$\mathrm{BWT}_t = \frac{1}{t-1}\sum_{i=1}^{t-1}(J_{t,i} - J_{i,i}). \tag{17}$$

There is also a metric called *forward transfer* (FWT) that calculate lifelong learning ability of LLM agent from the third perspective. It represents whether the gained experience is useful to future tasks.

$$\mathrm{FWT}_t = \frac{1}{t-1}\sum_{i=2}^{t}(J_{i,i} - \tilde{J}_i), \tag{18}$$

where $\tilde{J}_i$ is the performance of a LLM agent that has no experience.

13.2 Datasets and Benchmarks

In this subsection, we introduce the benchmarks and datasets commonly used to evaluate the lifelong learning capabilities of LLM agents. We categorize these benchmarks and datasets into two groups: the first focuses on assessing the ability of LLM agents to perform continual learning in *isolated, simple scenarios*, while the second evaluates their performance in more *integrated and complex scenarios*.

The first group of datasets and benchmarks is typically used to evaluate the lifelong learning ability of LLMs in simple scenarios. During the pretraining process, LLMs not only acquire knowledge but also develop specific skills, such as instruction following, machine translation, question answering, and code generation, through targeted corpora. Since the benchmarks and datasets in simple scenarios typically focus only on changes in a specific skill during the lifelong learning process, LLM agents only need to achieve lifelong learning in that skill to perform well on these benchmarks and datasets.

Many traditional continual learning benchmarks and datasets belong to the first group. For example, CITB [51], GLUE [291], SuperGLUE [292], NaturalInstruction [293] and SuperNI [294] mainly concern the instruction following ability of LLM agent. WMT [295], TED Talks [296] and CLLE [297] concern the machine translation aspect. There also exists benchmarks and datasets that concern the change of internal knowledge of LLM during the lifelong learning process, such as zsRE [298], FEVER [299], CounterFact [300], Concept-1K [301] and Biography [302].

The second groups of datasets and benchmarks used to evaluation the lifelong learning ability of LLMs in complex scenarios. These benchmarks and datasets closely reflect real-world scenarios, assessing the performance of LLM agents based on their ability to fulfill practical user requests. To succeed, LLM agents must achieve lifelong learning across multiple skills, effectively utilize knowledge from previous tasks, and seamlessly integrate capabilities such as reasoning, planning, multi-turn dialogue, knowledge retrieval, and tool usage.

Many benchmarks and datasets that evaluate the performance of LLM agent under special environment belong to the second group. For example, ToolBench [245], [248], StableToolBench [275], APIBench [276], ToolAlpaca [277],

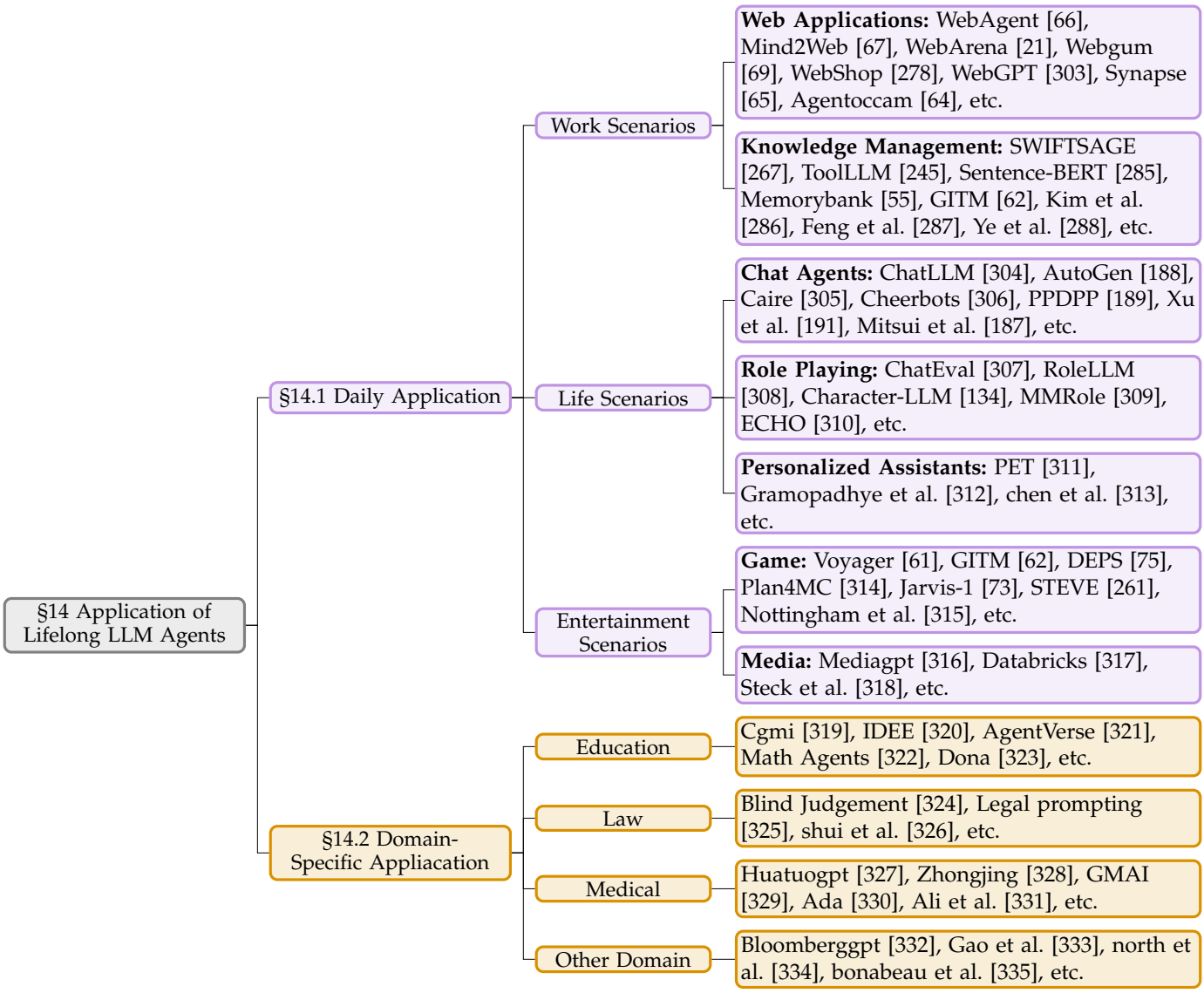


Fig. 16. Overview of the major application areas for lifelong learning LLM-based agents. These applications are broadly categorized into everyday use cases—encompassing work, life, and entertainment scenarios—and domain-specific implementations, such as education, law, and healthcare. By continuously adapting to user needs and drawing on accumulated knowledge, lifelong agents improve user experiences, decision-making processes, and operational efficiency across diverse contexts.

API-Bank [268] evaluate LLM agent's performance in tool environment. WebArena [21], WebShop [278], WorkArena [23], VisualWebArena [19], VideoWebArena [18], AgentBench [22], Visualagentbench [20] evaluate its performance under more challenge environment, such as web environment and game environment.

14 Application of Lifelong LLM Agents

In the digital age, LLM-based agents play an increasingly significant role in both everyday life and professional settings. Driven by advances in the lifelong learning paradigm, these agents continuously adapt and optimize their functionalities to meet users' evolving needs. The applications of these lifelong agents can be broadly categorized into two areas: *daily application* and *domain-specific application*. A summary of the related research appears in Figure 16.

14.1 Daily Application

In the context of human daily life, LLM agents significantly enhance people's work, life, and entertainment experiences through continuous learning and adaptation. These intelligent agents not only comprehend user needs but also dynamically adjust their functionalities to better serve users' everyday activities. Specifically, daily applications can be categorized into several key scenarios as follows:

In work scenarios. Agents play a variety of key roles that significantly enhance work and learning efficiency. For example, on web pages [21], [64]–[67], [69], [278], [303], agents continuously optimize search algorithms and content recommendations through lifelong learning, helping users find relevant information and resources more efficiently. In knowledge management, LLM-based agents effectively organize and retrieve information [55], [62], [245], [267], [285]–[289], assisting users in quickly accessing the knowledge

they need, promoting information sharing, and supporting decision-making.

In life scenarios. Lifelong agents have the potential to enhance the convenience and comfort of daily life. In terms of communication, LLM-based agents, integrating lifelong learning approaches such as role-playing [134], [307]–[310] and long-context text understanding [193], [336]–[340], can engage in continuous interactions with users, gradually comprehending their personalities and preferences. This enables the provision of a more natural and emotionally resonant conversational experience [187]–[189], [191], [304]–[306]. As personalized assistants, these agents can also assist users in completing daily household tasks based on their environments [311]–[313], such as automatically adjusting air conditioning, lighting, and cleaning, thereby improving the overall user experience.

In entertainment scenarios. Agents also play a significant role. For instance, in gaming, Minecraft, as an open-world simulation game, has emerged as a primary choice for testing agents within the gaming environment [61], [62], [73], [75], [261], [314], [315]. Specifically, Voyager [61], which is the first lifelong agent in Minecraft, is capable of autonomous exploration of unknown worlds without human intervention, utilizing a feedback mechanism. JARVIS-1 [73] enhances its understanding of the environment through self-reflection and self-explanation, incorporating previous plans into its prompts. Additionally, the entertainment media industry is undergoing an intelligent transformation. By continuously gathering the latest information from users, it recommends relevant high-quality movies and music to them [316]–[318].

14.2 Domain-Specific Application

In domain-specific applications, lifelong agents demonstrate remarkable adaptability and expertise, providing customized solutions across various industries. These intelligent agents continuously accumulate industry knowledge and user feedback through lifelong learning, thereby enhancing their effectiveness in specific fields.

In the education domain, LLM-based agents facilitate deep understanding of knowledge by simulating classroom environments and teacher-student interactions [319], [321]–[323], [341], as well as offering personalized learning support and resources [322], [342]. For teachers, the introduction of agents can assist in grading assignments, collecting class materials, and acting as an assistant to address students' questions [343], [344]. For students, one of the main advantages of using large language models like ChatGPT in education is their ability to help students complete assignments more efficiently and provide personalized learning experiences [320], [345]–[347]. Additionally, a lifelong agent can also serve as a teacher to instruct other models [311], [348], [349].

In the law domain, these agents can analyze legal documents and cases to provide users with legal advice and compliance recommendations [326], as well as assist in legal decision-making and drafting [324], [325], [350]–[352].

In the medical domain, LLM-based agents also have broad applications [329], [353]–[356]. Some methods assist doctors in diagnostic and treatment decision-making [327], [328], [331]. By using tools [330], [357], agents enable the system to interact with patients, enhancing the quality and efficiency of healthcare services.

Furthermore, in other industries, lifelong agents can adapt to new real-world tasks through continuous learning, thereby reducing labor costs [332]–[335].

15 Practical Insights and Future Directions

Building lifelong learning LLM agents involves a tight integration of perception, memory, and action modules, each playing a distinct yet interdependent role in enabling continuous adaptation to evolving tasks and environments [13], [14]. Although the progress in each module is substantial, several practical insights and future research directions remain.

15.1 Perception Module: Enhancing Robustness and Multimodality

The perception module has evolved from handling primarily textual inputs to addressing complex, multimodal inputs, including web content, images, and game environments [21]. Current approaches often rely on handcrafted prompt-engineering [61], compression strategies [123], or pretrained vision-language alignment models [15] to incorporate new modalities. However, a key challenge is ensuring that the perception module continues to function reliably when confronted with novel data distributions or modalities absent during initial training.

Adaptive Perception Architectures. Developing methods for automatic modality selection and integration that scale to newly introduced data types without extensive human intervention is essential for enhancing adaptability.

Domain-Agnostic Perception. Approaches that utilize universal, modality-agnostic encoders can handle arbitrary input formats, continuously learning cross-modal alignment strategies to maintain versatility across different domains.

Contextualized Perception Memory. Implementing mechanisms to store and retrieve perception-related knowledge enables the agent to recall representations of previously encountered modalities or domains, facilitating more efficient adaptation.

15.2 Memory Module: Balancing Stability, Plasticity, and Scalability

The memory module—comprising working, episodic, semantic, and parametric memory—forms the backbone of an agent's ability to learn over time. While current strategies mitigate catastrophic forgetting and integrate new knowledge, managing the ever-growing volume of information remains a significant challenge. Existing solutions often rely on replay buffers [43], knowledge distillation [113], [199], [358], or architectural modifications [44], [46] to store and recall old knowledge.

Memory Architecture Specialization. Designing hierarchical memory structures that differentiate between short-term context (working memory), long-term events (episodic memory), structured knowledge (semantic memory), and

learned weights (parametric memory) can enhance memory efficiency and retrieval accuracy.

Optimized Retrieval Mechanisms. Incorporating sophisticated retrieval strategies beyond naive nearest-neighbor approaches allows for the dynamic selection of the most relevant past experiences, scaling effectively with task complexity and the agent's lifetime.

Dynamic Memory Management. Investigating techniques that proactively prune or summarize outdated or less relevant knowledge is crucial for balancing memory growth with the retention of critical information.

Neuro-inspired Consolidation. Drawing inspiration from human cognition for memory consolidation processes ensures that new memories integrate or coexist with existing ones without overwriting them, promoting long-term knowledge retention.

Transfer Learning for Rapid Adaptation. Incorporating transfer learning techniques allows agents to quickly learn new tasks by leveraging previously acquired knowledge, enhancing learning speed and efficiency in dynamic environments [359].

Scalable Memory Architectures. Developing memory systems that are infinitely expandable while ensuring fast storage and retrieval is essential for handling the growing knowledge base of lifelong learning agents. Techniques such as scalable indexing and distributed memory systems can achieve this balance.

15.3 Action Module: Complex Reasoning and Efficient Adaptation

The action module enables LLM agents to interact with their environment, perform grounding actions, retrieve knowledge, and reason effectively. While existing methods have successfully integrated tool usage [59], [245], [246], web navigation [21], [303], and game-based tasks [61], the complexity of real-world environments calls for more advanced action design principles.

Hierarchical Action Spaces. Structuring action policies into hierarchies allows for flexible and scalable control, enabling agents to break down complex tasks into subtasks and reuse previously acquired skills.

Continuous Self-improvement. Leveraging reasoning trajectories, retrieval actions, and reflection-based methods can refine decision-making across trials and episodes, allowing the agent to improve its actions over its entire operational lifetime.

Task-Agnostic Action Generalization. Developing frameworks that promote the transferability of action policies across different tasks and environments reduces the need for environment-specific tuning or retraining.

Human-in-the-Loop Fine-tuning. Involving end-users or domain experts to provide corrective feedback and demonstrations guides the agent toward more robust and human-aligned action policies.

Reinforcement Learning for Self-Learning and Evolution. Integrating reinforcement learning approaches enables agents to self-learn and evolve through interactions with the environment, promoting autonomous adaptation and improvement over time [360].

15.4 Integrative Approaches and Long-Horizon Planning

The interplay between perception, memory, and action modules is crucial [14]. As lifelong LLM agents must continuously integrate new modalities, knowledge, and action repertoires, future research should emphasize integrative approaches. Developing unified frameworks where each module can inform and refine the others is key to unlocking the full potential of lifelong adaptation.

Perception-Memory-Action Feedback Loops. Structuring an agent's architecture to enable direct feedback across modules ensures that perception outputs are informed by memory state, and action decisions are guided by both perception and memory cues.

Curriculum and Continual Curriculum Learning. Introducing curricula that incrementally increase task complexity helps the agent's evolving perception, memory, and action modules develop robust long-horizon planning capabilities.

Tool and Knowledge Graph Integration. Seamlessly incorporating external tools and knowledge bases improves both memory recall and action precision, enabling agents to leverage external resources for long-term skill acquisition.

Collaborative Learning Frameworks. Encouraging agents to learn collaboratively can enhance knowledge sharing and accelerate the learning process, leading to more efficient long-horizon planning and decision-making.

The field of lifelong LLM agents is poised for transformative growth [13]. By focusing on robust perception designs that adapt to new modalities, memory architectures that efficiently handle ever-growing knowledge, and action modules that continually refine decision-making, future research can foster agents that not only excel in their initial domains but also adapt gracefully to emerging tasks. Long-term, the goal is to create LLM agents that resemble human-like learners, capable of genuine lifelong learning—forever perceiving, reasoning, and acting in increasingly complex, dynamic worlds.

16 Conclusion

In this survey, we examine the evolving landscape of lifelong learning for LLM-based agents. We begin by highlighting the motivation and foundational definitions of lifelong LLM agents, underscoring their capacity not only to adapt to changing tasks and environments but also to continuously improve from past experiences. Through a systematic exploration of perception, memory, and action modules—each playing a critical role in enabling lifelong learning—we showcase how state-of-the-art methods and frameworks address the challenges of integrating and retaining knowledge over extended periods.

We further discuss evaluation metrics, benchmarks, and application scenarios that underscore the practical significance of lifelong LLM agents in both everyday and specialized domains. Finally, we offer insights and directions for future research, aiming to inspire the development of agents that more closely mirror the adaptability, resilience, and learning capabilities seen in human intelligence. As LLM-based agents continue to evolve, the pursuit of robust lifelong learning methods remains a vital frontier, promising

increasingly capable, adaptable, and context-aware solutions to complex, real-world challenges.

REFERENCES

[1] J. Zheng, S. Qiu, C. Shi, and Q. Ma, "Towards lifelong learning of large language models: A survey," *ArXiv preprint*, vol. abs/2406.06391, 2024. [Online]. Available: https://arxiv.org/abs/2406.06391

[2] Z. Chen and B. Liu, *Lifelong machine learning*. Morgan & Claypool Publishers, 2018.

[3] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[4] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[5] S. Legg, "Machine super intelligence," 2008.

[6] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[7] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, "Loss of plasticity in deep continual learning," *Nature*, vol. 632, no. 8026, pp. 768–774, 2024.

[8] Z. Abbas, R. Zhao, J. Modayil, A. White, and M. C. Machado, "Loss of plasticity in continual deep reinforcement learning," in *Conference on Lifelong Learning Agents*. PMLR, 2023, pp. 620–636.

[9] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[10] Z. Z. Wang, J. Mao, D. Fried, and G. Neubig, "Agent workflow memory," *ArXiv preprint*, vol. abs/2409.07429, 2024. [Online]. Available: https://arxiv.org/abs/2409.07429

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[12] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *ArXiv preprint*, vol. abs/2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[13] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[14] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *ArXiv preprint*, vol. abs/2309.07864, 2023. [Online]. Available: https://arxiv.org/abs/2309.07864

[15] X. Liu, B. Qin, D. Liang, G. Dong, H. Lai, H. Zhang, H. Zhao, I. L. Iong, J. Sun, J. Wang *et al.*, "Autoglm: Autonomous foundation agents for guis," *ArXiv preprint*, vol. abs/2411.00820, 2024. [Online]. Available: https://arxiv.org/abs/2411.00820

[16] A. Zhao, D. Huang, Q. Xu, M. Lin, Y. Liu, and G. Huang, "Expel: LLM agents are experiential learners," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 19 632–19 642. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29936

[17] W. Pian, S. Deng, S. Mo, Y. Guo, and Y. Tian, "Modality-inconsistent continual learning of multimodal large language models," *ArXiv preprint*, vol. abs/2412.13050, 2024. [Online]. Available: https://arxiv.org/abs/2412.13050

[18] L. Jang, Y. Li, C. Ding, J. Lin, P. P. Liang, D. Zhao, R. Bonatti, and K. Koishida, "Videowebarena: Evaluating long context multimodal agents with video understanding web tasks," *ArXiv preprint*, vol. abs/2410.19100, 2024. [Online]. Available: https://arxiv.org/abs/2410.19100

[19] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried, "Visualwebarena: Evaluating multimodal agents on realistic visual web tasks," *ArXiv preprint*, vol. abs/2401.13649, 2024. [Online]. Available: https://arxiv.org/abs/2401.13649

[20] X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao *et al.*, "Visualagentbench: Towards large multimodal models as visual foundation agents," *ArXiv preprint*, vol. abs/2408.06327, 2024. [Online]. Available: https://arxiv.org/abs/2408.06327

[21] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig, "Webarena: A realistic web environment for building autonomous agents," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=oKn9c6ytLx

[22] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang *et al.*, "Agentbench: Evaluating llms as agents," *ArXiv preprint*, vol. abs/2308.03688, 2023. [Online]. Available: https://arxiv.org/abs/2308.03688

[23] A. Drouin, M. Gasse, M. Caccia, I. H. Laradji, M. Del Verme, T. Marty, L. Boisvert, M. Thakkar, Q. Cappart, D. Vazquez *et al.*, "Workarena: How capable are web agents at solving common knowledge work tasks?" *ArXiv preprint*, vol. abs/2403.07718, 2024. [Online]. Available: https://arxiv.org/abs/2403.07718

[24] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari, "Continual learning for large language models: A survey," *ArXiv preprint*, vol. abs/2402.01364, 2024. [Online]. Available: https://arxiv.org/abs/2402.01364

[25] H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, Z. Wang, S. Ebrahimi, and H. Wang, "Continual learning of large language models: A comprehensive survey," *ArXiv preprint*, vol. abs/2404.16789, 2024. [Online]. Available: https://arxiv.org/abs/2404.16789

[26] Z. Ke and B. Liu, "Continual learning of natural language processing tasks: A survey," *ArXiv preprint*, vol. abs/2211.12701, 2022. [Online]. Available: https://arxiv.org/abs/2211.12701

[27] D.-W. Zhou, H.-L. Sun, J. Ning, H.-J. Ye, and D.-C. Zhan, "Continual learning with pre-trained models: A survey," *ArXiv preprint*, vol. abs/2401.16386, 2024. [Online]. Available: https://arxiv.org/abs/2401.16386

[28] M. Biesialska, K. Biesialska, and M. R. Costa-jussà, "Continual lifelong learning in natural language processing: A survey," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 6523–6541. [Online]. Available: https://aclanthology.org/2020.coling-main.574

[29] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[30] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun *et al.*, "Personal llm agents: Insights and survey about the capability, efficiency and security," *ArXiv preprint*, vol. abs/2401.05459, 2024. [Online]. Available: https://arxiv.org/abs/2401.05459

[31] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects," *ArXiv preprint*, vol. abs/2401.03428, 2024. [Online]. Available: https://arxiv.org/abs/2401.03428

[32] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang, "A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges," *Vicinagearth*, vol. 1, no. 1, p. 9, 2024.

[33] D. Zhang, L. Chen, S. Zhang, H. Xu, Z. Zhao, and K. Yu, "Large language models are semi-parametric reinforcement learning agents," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing*

Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/f6b22ac37beb5da61efd4882082c9ecd-Abstract-Conference.html

[34] G. A. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, no. 3, pp. 77–88, 1988.

[35] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[36] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[37] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[38] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[39] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive science*, vol. 11, no. 1, pp. 23–63, 1987.

[40] S. Thrun, "Lifelong learning algorithms," in *Learning to learn*. Springer, 1998, pp. 181–209.

[41] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[42] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[43] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 348–358. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html

[44] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 7765–7773. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Mallya_PackNet_Adding_Multiple_CVPR_2018_paper.html

[45] S. V. Mehta, D. Patil, S. Chandar, and E. Strubell, "An empirical investigation of the role of pre-training in lifelong learning," *Journal of Machine Learning Research*, vol. 24, no. 214, pp. 1–50, 2023.

[46] A. Razdaibiedina, Y. Mao, R. Hou, M. Khabsa, M. Lewis, and A. Almahairi, "Progressive prompts: Continual learning for language models," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=UJTgQBc91_

[47] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 2790–2799. [Online]. Available: http://proceedings.mlr.press/v97/houlsby19a.html

[48] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[49] X. Jin, D. Zhang, H. Zhu, W. Xiao, S.-W. Li, X. Wei, A. Arnold, and X. Ren, "Lifelong pretraining: Continually adapting language models to emerging corpora," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, A. Fan, S. Ilic, T. Wolf, and M. Gallé, Eds. virtual+Dublin: Association for Computational Linguistics, 2022, pp. 1–16. [Online]. Available: https://aclanthology.org/2022.bigscience-1.1

[50] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An empirical study of catastrophic forgetting in large language models during continual fine-tuning," *ArXiv preprint*, vol. abs/2308.08747, 2023. [Online]. Available: https://arxiv.org/abs/2308.08747

[51] Z. Zhang, M. Fang, L. Chen, and M.-R. Namazi-Rad, "CITB: A benchmark for continual instruction tuning," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 9443–9455. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.633

[52] T. Shen, R. Jin, Y. Huang, C. Liu, W. Dong, Z. Guo, X. Wu, Y. Liu, and D. Xiong, "Large language model alignment: A survey," *ArXiv preprint*, vol. abs/2309.15025, 2023. [Online]. Available: https://arxiv.org/abs/2309.15025

[53] J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, and M. Seo, "Towards continual knowledge learning of language models," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=vfsRB5MImo9

[54] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li, "Knowledge editing for large language models: A survey," *ACM Computing Surveys*, 2023.

[55] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: Enhancing large language models with long-term memory," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 19 724–19 731. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29946

[56] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, and J.-R. Wen, "A survey on the memory mechanism of large language model based agents," *ArXiv preprint*, vol. abs/2404.13501, 2024. [Online]. Available: https://arxiv.org/abs/2404.13501

[57] J. Lu, S. An, M. Lin, G. Pergola, Y. He, D. Yin, X. Sun, and Y. Wu, "Memochat: Tuning llms to use memos for consistent long-range open-domain conversation," *ArXiv preprint*, vol. abs/2308.08239, 2023. [Online]. Available: https://arxiv.org/abs/2308.08239

[58] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html

[59] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html

[60] C. Hu, J. Fu, C. Du, S. Luo, J. Zhao, and H. Zhao, "Chatdb: Augmenting llms with databases as their symbolic memory," *ArXiv preprint*, vol. abs/2306.03901, 2023. [Online]. Available: https://arxiv.org/abs/2306.03901

[61] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied

agent with large language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.16291

[62] X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, Y. Qiao, Z. Zhang, and J. Dai, "Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory," 2023. [Online]. Available: https://arxiv.org/abs/2305.17144

[63] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models. arxiv 2023," *ArXiv preprint*, vol. abs/2302.13971, 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[64] K. Yang, Y. Liu, S. Chaudhary, R. Fakoor, P. Chaudhari, G. Karypis, and H. Rangwala, "Agentoccam: A simple yet strong baseline for llm-based web agents," *ArXiv preprint*, vol. abs/2410.13825, 2024. [Online]. Available: https://arxiv.org/abs/2410.13825

[65] L. Zheng, R. Wang, X. Wang, and B. An, "Synapse: Trajectory-as-exemplar prompting with memory for computer control," in *The Twelfth International Conference on Learning Representations*, 2023.

[66] I. Gur, H. Furuta, A. Huang, M. Safdari, Y. Matsuo, D. Eck, and A. Faust, "A real-world webagent with planning, long context understanding, and program synthesis," *ArXiv preprint*, vol. abs/2307.12856, 2023. [Online]. Available: https://arxiv.org/abs/2307.12856

[67] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html

[68] P. Shaw, M. Joshi, J. Cohan, J. Berant, P. Pasupat, H. Hu, U. Khandelwal, K. Lee, and K. Toutanova, "From pixels to UI actions: Learning to follow instructions via graphical user interfaces," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/6c52a8a4fadc9129c6e1d1745f2dfd0f-Abstract-Conference.html

[69] H. Furuta, K.-H. Lee, O. Nachum, Y. Matsuo, A. Faust, S. S. Gu, and I. Gur, "Multimodal web navigation with instruction-finetuned foundation models," *ArXiv preprint*, vol. abs/2305.11854, 2023. [Online]. Available: https://arxiv.org/abs/2305.11854

[70] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding *et al.*, "Cogagent: A visual language model for gui agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 281–14 290.

[71] J. Pan, Y. Zhang, N. Tomlin, Y. Zhou, S. Levine, and A. Suhr, "Autonomous evaluation and refinement of digital agents," *ArXiv preprint*, vol. abs/2404.06474, 2024. [Online]. Available: https://arxiv.org/abs/2404.06474

[72] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su, "Gpt-4v (ision) is a generalist web agent, if grounded," *ArXiv preprint*, vol. abs/2401.01614, 2024. [Online]. Available: https://arxiv.org/abs/2401.01614

[73] Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang *et al.*, "Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models," *ArXiv preprint*, vol. abs/2311.05997, 2023. [Online]. Available: https://arxiv.org/abs/2311.05997

[74] Y. Dong, X. Zhu, Z. Pan, L. Zhu, and Y. Yang, "VillagerAgent: A graph-based multi-agent framework for coordinating complex task dependencies in Minecraft," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 16 290–16 314. [Online]. Available: https://aclanthology.org/2024.findings-acl.964

[75] Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," *ArXiv preprint*, vol. abs/2302.01560, 2023. [Online]. Available: https://arxiv.org/abs/2302.01560

[76] J. Urbanek, A. Fan, S. Karamcheti, S. Jain, S. Humeau, E. Dinan, T. Rocktäschel, D. Kiela, A. Szlam, and J. Weston, "Learning to speak and act in a fantasy text adventure game," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 673–683. [Online]. Available: https://aclanthology.org/D19-1062

[77] Y. Li, J. He, X. Zhou, Y. Zhang, and J. Baldridge, "Mapping natural language instructions to mobile UI action sequences," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, 2020, pp. 8198–8210. [Online]. Available: https://aclanthology.org/2020.acl-main.729

[78] Z. Zhang and A. Zhang, "You only look at screens: Multimodal chain-of-action agents," *ArXiv preprint*, vol. abs/2309.11436, 2023. [Online]. Available: https://arxiv.org/abs/2309.11436

[79] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 4651–4664. [Online]. Available: http://proceedings.mlr.press/v139/jaegle21a.html

[80] H. Akbari, L. Yuan, R. Qian, W. Chuang, S. Chang, Y. Cui, and B. Gong, "VATT: transformers for multimodal self-supervised learning from raw video, audio and text," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 24 206–24 221. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/cb3213ada48302953cb0f166464ab356-Abstract.html

[81] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 16 081–16 091. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.01563

[82] X. Wang, B. Zhuang, and Q. Wu, "Modaverse: Efficiently transforming modalities with llms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 606–26 616.

[83] H. Tan and M. Bansal, "Vokenization: Improving language understanding with contextualized, visual-grounded supervision," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, 2020, pp. 2066–2080. [Online]. Available: https://aclanthology.org/2020.emnlp-main.162

[84] Z. Tang, J. Cho, H. Tan, and M. Bansal, "Vidlankd: Improving language understanding via video-distilled knowledge transfer," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 24 468–24 481. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/ccdf3864e2fa9089f9eca4fc7a48ea0a-Abstract.html

[85] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2827–2836. [Online]. Available: https://doi.org/10.1109/CVPR.2016.309

[86] Z. Zheng, M. Ma, K. Wang, Z. Qin, X. Yue, and Y. You, "Preventing zero-shot transfer degradation in continual learning of vision-language models," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 19 068–19 079. [Online]. Available: https://doi.org/10.1109/ICCV51070.2023.01752

[87] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-

visual navigation in 3d environments," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 17–36.

[88] X. Chen, J. Zhang, X. Wang, N. Zhang, T. Wu, Y. Wang, Y. Wang, and H. Chen, "Continual multimodal knowledge graph construction," *ArXiv preprint*, vol. abs/2305.08698, 2023. [Online]. Available: https://arxiv.org/abs/2305.08698

[89] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 5491–5500. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00559

[90] Y. Zhang, L. Shao, and C. G. M. Snoek, "Repetitive activity counting by sight and sound," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 14 070–14 079. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_Repetitive_Activity_Counting_by_Sight_and_Sound_CVPR_2021_paper.html

[91] M. Ma, J. Ren, L. Zhao, S. Tulyakov, C. Wu, and X. Peng, "SMIL: multimodal learning with severely missing modality," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 2302–2310. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16330

[92] J. Yu, H. Xiong, L. Zhang, H. Diao, Y. Zhuge, L. Hong, D. Wang, H. Lu, Y. He, and L. Chen, "Llms can evolve continually on modality for x-modal reasoning," *ArXiv preprint*, vol. abs/2410.20178, 2024. [Online]. Available: https://arxiv.org/abs/2410.20178

[93] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He, "Boosting continual learning of vision-language models via mixture-of-experts adapters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 219–23 230.

[94] J. Zhao, R. Li, and Q. Jin, "Missing modality imagination network for emotion recognition with uncertain missing modalities," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, 2021, pp. 2608–2618. [Online]. Available: https://aclanthology.org/2021.acl-long.203

[95] Y. Wang, Z. Cui, and Y. Li, "Distribution-consistent modal recovering for incomplete multimodal learning," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 21 968–21 977. [Online]. Available: https://doi.org/10.1109/ICCV51070.2023.02013

[96] Y. Zhang, H. Doughty, and C. Snoek, "Learning unseen modality interaction," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/abb4847bbd60f38b1b7649d26c7a0067-Abstract-Conference.html

[97] H. Wang, Y. Chen, C. Ma, J. Avery, L. Hull, and G. Carneiro, "Multi-modal learning with missing modality via shared-specific feature modelling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 878–15 887. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.01524

[98] R. Lin and H. Hu, "MissModal: Increasing robustness to missing modality in multimodal sentiment analysis," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1686–1702, 2023. [Online]. Available: https://aclanthology.org/2023.tacl-1.94

[99] M. Ma, J. Ren, L. Zhao, D. Testuggine, and X. Peng, "Are multimodal transformers robust to missing modality?" in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 18 156–18 165. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.01764

[100] J. He, H. Guo, M. Tang, and J. Wang, "Continual instruction tuning for large multimodal models," *ArXiv preprint*, vol. abs/2311.16206, 2023. [Online]. Available: https://arxiv.org/abs/2311.16206

[101] D. Zhu, Z. Sun, Z. Li, T. Shen, K. Yan, S. Ding, K. Kuang, and C. Wu, "Model tailor: Mitigating catastrophic forgetting in multi-modal large language models," *ArXiv preprint*, vol. abs/2402.12048, 2024. [Online]. Available: https://arxiv.org/abs/2402.12048

[102] Z. Ni, L. Wei, S. Tang, Y. Zhuang, and Q. Tian, "Continual vision-language representation learning with off-diagonal information," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 26 129–26 149. [Online]. Available: https://proceedings.mlr.press/v202/ni23c.html

[103] H. Zhu, Y. Wei, X. Liang, C. Zhang, and Y. Zhao, "CTP: towards vision-language continual pretraining via compatible momentum contrast and topology preservation," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 22 200–22 210. [Online]. Available: https://doi.org/10.1109/ICCV51070.2023.02034

[104] X. Zhang, F. Zhang, and C. Xu, "VQACL: A novel visual question answering continual learning setting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 19 102–19 112. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.01831

[105] F. Sarfraz, B. Zonooz, and E. Arani, "Beyond unimodal learning: The importance of integrating multiple modalities for lifelong learning," *ArXiv preprint*, vol. abs/2405.02766, 2024. [Online]. Available: https://arxiv.org/abs/2405.02766

[106] Y. Cai, J. Thomason, and M. Rostami, "Task-attentive transformer architecture for continual learning of vision-and-language tasks using knowledge distillation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 6986–7000. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.466

[107] R. Yang, S. Wang, H. Zhang, S. Xu, Y. Guo, X. Ye, B. Hou, and L. Jiao, "Knowledge decomposition and replay: A novel cross-modal image-text retrieval continual learning method," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 6510–6519.

[108] S. Yan, L. Hong, H. Xu, J. Han, T. Tuytelaars, Z. Li, and X. He, "Generative negative text replay for continual vision-language pretraining," in *European Conference on Computer Vision*. Springer, 2022, pp. 22–38.

[109] S. W. Lei, D. Gao, J. Z. Wu, Y. Wang, W. Liu, M. Zhang, and M. Z. Shou, "Symbolic replay: Scene graph as prompt for continual learning on VQA task," in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, B. Williams, Y. Chen, and J. Neville, Eds. AAAI Press, 2023, pp. 1250–1259. [Online]. Available: https://doi.org/10.1609/aaai.v37i1.25208

[110] C. He, S. Cheng, Z. Qiu, L. Xu, F. Meng, Q. Wu, and H. Li, "Continual egocentric activity recognition with foreseeable-generalized visual-imu representations," *IEEE Sensors Journal*, 2024.

[111] S. Cheng, C. He, K. Chen, L. Xu, H. Li, F. Meng, and Q. Wu, "Vision-sensor attention based continual multimodal egocentric activity recognition," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 6300–6304.

[112] W. Jin, D.-H. Lee, C. Zhu, J. Pujara, and X. Ren, "Leveraging visual knowledge in language tasks: An empirical study on intermediate pre-training for cross-modal knowledge transfer," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 2750–2762. [Online]. Available: https://aclanthology.org/2022.acl-long.196

[113] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation:

A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[114] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, pp. 275–293, 2014.

[115] G. Song and X. Tan, "Real-world cross-modal retrieval via sequential learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 1708–1721, 2020.

[116] J. Han, K. Gong, Y. Zhang, J. Wang, K. Zhang, D. Lin, Y. Qiao, P. Gao, and X. Yue, "Onellm: One framework to align all modalities with language," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 584–26 595.

[117] S. Moon, A. Madotto, Z. Lin, T. Nagarajan, M. Smith, S. Jain, C.-F. Yeh, P. Murugesan, P. Heidari, Y. Liu *et al.*, "Anymal: An efficient and scalable any-modality augmented language model," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024, pp. 1314–1332.

[118] J. Zheng, Q. Ma, Z. Liu, B. Wu, and H. Feng, "Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer," *ArXiv preprint*, vol. abs/2401.09181, 2024. [Online]. Available: https://arxiv.org/abs/2401.09181

[119] R. D. Chiaro, B. Twardowski, A. D. Bagdanov, and J. van de Weijer, "RATT: recurrent attention to transient tasks for continual image captioning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/c2964caac096f26db222cb325aa267cb-Abstract.html

[120] A. Chevalier, A. Wettig, A. Ajith, and D. Chen, "Adapting language models to compress contexts," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 3829–3846. [Online]. Available: https://aclanthology.org/2023.emnlp-main.232

[121] J. Mu, X. Li, and N. D. Goodman, "Learning to compress prompts with gist tokens," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html

[122] Y. Li, B. Dong, F. Guerin, and C. Lin, "Compressing context to enhance inference efficiency of large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 6342–6353. [Online]. Available: https://aclanthology.org/2023.emnlp-main.391

[123] H. Jiang, Q. Wu, C.-Y. Lin, Y. Yang, and L. Qiu, "LLMLingua: Compressing prompts for accelerated inference of large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 13 358–13 376. [Online]. Available: https://aclanthology.org/2023.emnlp-main.825

[124] H. Jiang, Q. Wu, X. Luo, D. Li, C.-Y. Lin, Y. Yang, and L. Qiu, "Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression," *ArXiv preprint*, vol. abs/2310.06839, 2023. [Online]. Available: https://arxiv.org/abs/2310.06839

[125] M. Ding, C. Zhou, H. Yang, and J. Tang, "Cogltx: Applying BERT to long texts," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/96671501524948bc3937b4b30d0e57b9-Abstract.html

[126] P. Manakul and M. Gales, "Long-span summarization via local attention and content selection," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, 2021, pp. 6026–6041. [Online]. Available: https://aclanthology.org/2021.acl-long.470

[127] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 484–494. [Online]. Available: https://aclanthology.org/P16-1046

[128] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1073–1083. [Online]. Available: https://aclanthology.org/P17-1099

[129] P. Gao, J. Lu, H. Li, R. Mottaghi, and A. Kembhavi, "Container: Context aggregation network," *ArXiv preprint*, vol. abs/2106.01401, 2021. [Online]. Available: https://arxiv.org/abs/2106.01401

[130] M. Ivgi, U. Shaham, and J. Berant, "Efficient long-text understanding with short-text models," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 284–299, 2023. [Online]. Available: https://aclanthology.org/2023.tacl-1.17

[131] N. Ratner, Y. Levine, Y. Belinkov, O. Ram, I. Magar, O. Abend, E. Karpas, A. Shashua, K. Leyton-Brown, and Y. Shoham, "Parallel context windows for large language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 6383–6402. [Online]. Available: https://aclanthology.org/2023.acl-long.352

[132] N. Chen, Y. Wang, Y. Deng, and J. Li, "The oscars of ai theater: A survey on role-playing with language models," *ArXiv preprint*, vol. abs/2407.11484, 2024. [Online]. Available: https://arxiv.org/abs/2407.11484

[133] J. Chen, X. Wang, R. Xu, S. Yuan, Y. Zhang, W. Shi, J. Xie, S. Li, R. Yang, T. Zhu *et al.*, "From persona to personalization: A survey on role-playing language agents," *ArXiv preprint*, vol. abs/2404.18231, 2024. [Online]. Available: https://arxiv.org/abs/2404.18231

[134] Y. Shao, L. Li, J. Dai, and X. Qiu, "Character-LLM: A trainable agent for role-playing," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 13 153–13 187. [Online]. Available: https://aclanthology.org/2023.emnlp-main.814

[135] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou *et al.*, "Metagpt: Meta programming for multi-agent collaborative framework," *ArXiv preprint*, vol. abs/2308.00352, 2023. [Online]. Available: https://arxiv.org/abs/2308.00352

[136] S. Han, L. Chen, L.-M. Lin, Z. Xu, and K. Yu, "Ibsen: Director-actor agent collaboration for controllable and interactive drama script generation," *ArXiv preprint*, vol. abs/2407.01093, 2024. [Online]. Available: https://arxiv.org/abs/2407.01093

[137] H. Li, Y. Chong, S. Stepputtis, J. Campbell, D. Hughes, C. Lewis, and K. Sycara, "Theory of mind for multi-agent collaboration via large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 180–192. [Online]. Available: https://aclanthology.org/2023.emnlp-main.13

[138] A. Fourney, G. Bansal, H. Mozannar, C. Tan, E. Salinas, F. Niedtner, G. Proebsting, G. Bassman, J. Gerrits, J. Alber *et al.*, "Magentic-one: A generalist multi-agent system for solving complex tasks," *ArXiv preprint*, vol. abs/2411.04468, 2024. [Online]. Available: https://arxiv.org/abs/2411.04468

[139] S. Mousavi, R. L. Gutiérrez, D. Rengarajan, V. Gundecha, A. R. Babu, A. Naug, A. Guillen, and S. Sarkar, "N-critics: Self-refinement of large language models with ensemble of critics," *ArXiv preprint*, vol. abs/2310.18679, 2023. [Online]. Available: https://arxiv.org/abs/2310.18679

[140] L. Li, Z. Chen, G. Chen, Y. Zhang, Y. Su, E. Xing, and K. Zhang, "Confidence matters: Revisiting intrinsic self-correction capabilities of large language models," *ArXiv preprint*, vol. abs/2402.12563, 2024. [Online]. Available: https://arxiv.org/abs/2402.12563

[141] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, and W. Chen, "Critic: Large language models can self-correct with tool-interactive critiquing," *ArXiv preprint*, vol. abs/2305.11738, 2023. [Online]. Available: https://arxiv.org/abs/2305.11738

[142] Q. Guo, R. Wang, J. Guo, B. Li, K. Song, X. Tan, G. Liu, J. Bian, and Y. Yang, "Connecting large language models with evolutionary algorithms yields powerful prompt optimizers," *ArXiv preprint*, vol. abs/2309.08532, 2023. [Online]. Available: https://arxiv.org/abs/2309.08532

[143] X. Wang, C. Li, Z. Wang, F. Bai, H. Luo, J. Zhang, N. Jojic, E. P. Xing, and Z. Hu, "Promptagent: Strategic planning with language models enables expert-level prompt optimization," *ArXiv preprint*, vol. abs/2310.16427, 2023. [Online]. Available: https://arxiv.org/abs/2310.16427

[144] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.

[145] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 11 816–11 825. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/e562cd9c0768d5464b64cf61da7fc6bb-Abstract.html

[146] L. Caccia, E. Belilovsky, M. Caccia, and J. Pineau, "Online learned continual compression with adaptive quantization modules," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1240–1250. [Online]. Available: http://proceedings.mlr.press/v119/caccia20a.html

[147] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5533–5542. [Online]. Available: https://doi.org/10.1109/CVPR.2017.587

[148] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.

[149] C. Wu, L. Herranz, X. Liu, Y. Wang, J. van de Weijer, and B. Raducanu, "Memory replay gans: Learning to generate new categories without forgetting," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 5966–5976. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/a57e8915461b83adefb011530b711704-Abstract.html

[150] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 2990–2999. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/0efbe98067c6c73dba1250d2beaa81f9-Abstract.html

[151] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=BkQqq0gRb

[152] C. He, R. Wang, S. Shan, and X. Chen, "Exemplar-supported generative reproduction for class incremental learning," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 98. [Online]. Available: http://bmvc2018.org/contents/papers/0325.pdf

[153] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. v. de Weijer, "Generative feature replay for class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 226–227.

[154] M. Toldo and M. Ozay, "Bring evanescent representations to life in lifelong class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 732–16 741.

[155] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, pp. 293–321, 1992.

[156] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.05952

[157] M. Ramicic and A. Bonarini, "Entropy-based prioritized sampling in deep q-learning," in *2017 2nd international conference on image, vision and computing (ICIVC)*. IEEE, 2017, pp. 1068–1072.

[158] A. A. Li, Z. Lu, and C. Miao, "Revisiting prioritized experience replay: A value perspective," *ArXiv preprint*, vol. abs/2102.03261, 2021. [Online]. Available: https://arxiv.org/abs/2102.03261

[159] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2216–2226, 2018.

[160] G. Novati and P. Koumoutsakos, "Remember and forget for experience replay," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 4851–4860. [Online]. Available: http://proceedings.mlr.press/v97/novati19a.html

[161] P. Sun, W. Zhou, and H. Li, "Attentive experience replay," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 5900–5907. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6049

[162] H. Liu, A. Trott, R. Socher, and C. Xiong, "Competitive experience replay," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Sklsm20ctX

[163] A. Modarressi, A. Imani, M. Fayyaz, and H. Schütze, "Ret-llm: Towards a general read-write memory for large language models," *ArXiv preprint*, vol. abs/2305.14322, 2023. [Online]. Available: https://arxiv.org/abs/2305.14322

[164] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 6467–6476. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html

[165] X. Kou, Y. Lin, S. Liu, P. Li, J. Zhou, and Y. Zhang, "Disentangle-based Continual Graph Representation Learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, 2020, pp. 2961–2972. [Online]. Available: https://aclanthology.org/2020.emnlp-main.237

[166] Y. Cui, Y. Wang, Z. Sun, W. Liu, Y. Jiang, K. Han, and W. Hu, "Lifelong embedding learning and transfer for growing knowledge graphs," in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, B. Williams, Y. Chen, and J. Neville, Eds. AAAI Press, 2023, pp. 4217–4224. [Online]. Available: https://doi.org/10.1609/aaai.v37i4.25539

[167] J. Su, D. Zou, Z. Zhang, and C. Wu, "Towards robust graph incremental learning on evolving graphs," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023,

pp. 32 728–32 748. [Online]. Available: https://proceedings.mlr.press/v202/su23a.html

[168] M. Chen, W. Zhang, Z. Yao, Y. Zhu, Y. Gao, J. Z. Pan, and H. Chen, "Entity-agnostic representation learning for parameter-efficient knowledge graph embedding," in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, B. Williams, Y. Chen, and J. Neville, Eds. AAAI Press, 2023, pp. 4182–4190. [Online]. Available: https://doi.org/10.1609/aaai.v37i4.25535

[169] J. Liu, W. Ke, P. Wang, J. Wang, J. Gao, Z. Shang, G. Li, Z. Xu, K. Ji, and Y. Li, "Fast and continual knowledge graph embedding via incremental lora," *ArXiv preprint*, vol. abs/2407.05705, 2024. [Online]. Available: https://arxiv.org/abs/2407.05705

[170] H. Chase, "LangChain," 2022. [Online]. Available: https://github.com/langchain-ai/langchain

[171] J. Liu, "LlamaIndex," 2022. [Online]. Available: https://github.com/jerryjliu/llama_index

[172] I. Martínez, D. Gallego Vico, and P. Orgaz, "PrivateGPT," 2023. [Online]. Available: https://github.com/imartinez/privateGPT

[173] Z. Cheng, T. Xie, P. Shi, C. Li, R. Nadkarni, Y. Hu, C. Xiong, D. Radev, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu, "Binding language models in symbolic languages," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=lH1PV42cbF

[174] R. Sun, S. Ö. Arik, A. Muzio, L. Miculicich, S. Gundabathula, P. Yin, H. Dai, H. Nakhost, R. Sinha, Z. Wang *et al.*, "Sql-palm: Improved large language model adaptation for text-to-sql (extended)," *ArXiv preprint*, vol. abs/2306.00739, 2023. [Online]. Available: https://arxiv.org/abs/2306.00739

[175] R. Pedro, D. Castro, P. Carreira, and N. Santos, "From prompt injections to sql injection attacks: How protected is your llm-integrated web application?" *ArXiv preprint*, vol. abs/2308.01990, 2023. [Online]. Available: https://arxiv.org/abs/2308.01990

[176] R. Liu, J. Zhang, Y. Song, Y. Zhang, and B. Yang, "Filling memory gaps: Enhancing continual semantic parsing via sql syntax variance-guided llms without real data replay," *ArXiv preprint*, vol. abs/2412.07246, 2024. [Online]. Available: https://arxiv.org/abs/2412.07246

[177] T. On, S. Ghosh, M. Du, and S. B. Roy, "Proportionate diversification of top-k llm results using database queries," *DEF*, vol. 2, p. 1, 2002.

[178] S. Xue, C. Jiang, W. Shi, F. Cheng, K. Chen, H. Yang, Z. Zhang, J. He, H. Zhang, G. Wei *et al.*, "Db-gpt: Empowering database interactions with private large language models," *ArXiv preprint*, vol. abs/2312.17449, 2023. [Online]. Available: https://arxiv.org/abs/2312.17449

[179] Y. Wang, J. Zeng, X. Liu, D. F. Wong, F. Meng, J. Zhou, and M. Zhang, "Delta: An online document-level translation agent based on multi-level memory," *ArXiv preprint*, vol. abs/2410.08143, 2024. [Online]. Available: https://arxiv.org/abs/2410.08143

[180] S. V. Mehta, J. Gupta, Y. Tay, M. Dehghani, V. Q. Tran, J. Rao, M. Najork, E. Strubell, and D. Metzler, "DSI++: Updating transformer memory with new documents," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 8198–8213. [Online]. Available: https://aclanthology.org/2023.emnlp-main.510

[181] V. Kishore, C. Wan, J. Lovelace, Y. Artzi, and K. Q. Weinberger, "Incdsi: Incrementally updatable document retrieval," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 17 122–17 134. [Online]. Available: https://proceedings.mlr.press/v202/kishore23a.html

[182] T.-L. Huynh, T.-T. Vu, W. Wang, Y. Wei, T. Le, D. Gasevic, Y.-F. Li, and T.-T. Do, "Promptdsi: Prompt-based rehearsal-free instance-wise incremental learning for document retrieval," *ArXiv preprint*, vol. abs/2406.12593, 2024. [Online]. Available: https://arxiv.org/abs/2406.12593

[183] J. Chen, R. Zhang, J. Guo, Y. Liu, Y. Fan, and X. Cheng, "Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 191–200.

[184] J. Guo, C. Zhou, R. Zhang, J. Chen, M. de Rijke, Y. Fan, and X. Cheng, "Corpusbrain++: A continual generative pre-training framework for knowledge-intensive language tasks," *ArXiv preprint*, vol. abs/2402.16767, 2024. [Online]. Available: https://arxiv.org/abs/2402.16767

[185] X. Zhou, G. Li, and Z. Liu, "Llm as dba," *ArXiv preprint*, vol. abs/2308.05481, 2023. [Online]. Available: https://arxiv.org/abs/2308.05481

[186] N. Liu, L. Chen, X. Tian, W. Zou, K. Chen, and M. Cui, "From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models," *ArXiv preprint*, vol. abs/2401.02777, 2024. [Online]. Available: https://arxiv.org/abs/2401.02777

[187] K. Mitsui, Y. Hono, and K. Sawada, "Towards human-like spoken dialogue generation between ai agents from written dialogue," *ArXiv preprint*, vol. abs/2310.01088, 2023. [Online]. Available: https://arxiv.org/abs/2310.01088

[188] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, "Autogen: Enabling next-gen llm applications via multi-agent conversation framework," *ArXiv preprint*, vol. abs/2308.08155, 2023. [Online]. Available: https://arxiv.org/abs/2308.08155

[189] Y. Deng, W. Zhang, W. Lam, S.-K. Ng, and T.-S. Chua, "Plug-and-play policy planner for large language model powered dialogue agents," in *The Twelfth International Conference on Learning Representations*, 2023.

[190] S. Qiu, S.-C. Zhu, and Z. Zheng, "Theory-of-mind enhanced dialogue generation in situated contexts."

[191] X. Xu, Z. Gou, W. Wu, Z.-Y. Niu, H. Wu, H. Wang, and S. Wang, "Long time no see! open-domain conversation with long-term persona memory," in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 2639–2650. [Online]. Available: https://aclanthology.org/2022.findings-acl.207

[192] Z. Huang, S. Gutierrez, H. Kamana, and S. MacNeil, "Memory sandbox: Transparent and interactive memory management for conversational agents," in *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–3.

[193] S. Bae, D. Kwak, S. Kang, M. Y. Lee, S. Kim, Y. Jeong, H. Kim, S.-W. Lee, W. Park, and N. Sung, "Keep me updated! memory management in long-term conversations," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 3769–3787. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.276

[194] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=WE_vluYUL-X

[195] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE transactions on knowledge and data engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[196] Z. Pan and P. Wang, "Hyperbolic hierarchy-aware knowledge graph embedding for link prediction," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2941–2948. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.251

[197] J. Liu, P. Wang, Z. Shang, and C. Wu, "Iterde: An iterative knowledge distillation framework for knowledge graph embeddings," in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, B. Williams,

Y. Chen, and J. Neville, Eds. AAAI Press, 2023, pp. 4488–4496. [Online]. Available: https://doi.org/10.1609/aaai.v37i4.25570

[198] Z. Shang, P. Wang, Y. Liu, J. Liu, and W. Ke, "Askrl: An aligned-spatial knowledge representation learning framework for open-world knowledge graph," in *International Semantic Web Conference*. Springer, 2023, pp. 101–120.

[199] J. Liu, W. Ke, P. Wang, Z. Shang, J. Gao, G. Li, K. Ji, and Y. Liu, "Towards continual knowledge graph embedding via incremental distillation," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 8759–8768. [Online]. Available: https://doi.org/10.1609/aaai.v38i8.28722

[200] H. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang, "Sentence embedding alignment for lifelong relation extraction," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 796–806. [Online]. Available: https://aclanthology.org/N19-1086

[201] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[202] X. Li, J. Jin, Y. Zhou, Y. Zhang, P. Zhang, Y. Zhu, and Z. Dou, "From matching to generation: A survey on generative information retrieval," *ArXiv preprint*, vol. abs/2404.14851, 2024. [Online]. Available: https://arxiv.org/abs/2404.14851

[203] Y. Tay, V. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. P. Gupta, T. Schuster, W. W. Cohen, and D. Metzler, "Transformer memory as a differentiable search index," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/892840a6123b5ec99ebaab8be1530fba-Abstract-Conference.html

[204] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian *et al.*, "Toolllm: Facilitating large language models to master 16000+ real-world apis," *ArXiv preprint*, vol. abs/2307.16789, 2023. [Online]. Available: https://arxiv.org/abs/2307.16789

[205] Z. Yuan, H. Yuan, C. Li, G. Dong, K. Lu, C. Tan, C. Zhou, and J. Zhou, "Scaling relationship on learning mathematical reasoning with large language models," *ArXiv preprint*, vol. abs/2308.01825, 2023. [Online]. Available: https://arxiv.org/abs/2308.01825

[206] Y. Xu, X. Liu, X. Liu, Z. Hou, Y. Li, X. Zhang, Z. Wang, A. Zeng, Z. Du, W. Zhao *et al.*, "Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline," *ArXiv preprint*, vol. abs/2404.02893, 2024. [Online]. Available: https://arxiv.org/abs/2404.02893

[207] A. Zeng, M. Liu, R. Lu, B. Wang, X. Liu, Y. Dong, and J. Tang, "Agenttuning: Enabling generalized agent abilities for llms," *ArXiv preprint*, vol. abs/2310.12823, 2023. [Online]. Available: https://arxiv.org/abs/2310.12823

[208] D. Yin, F. Brahman, A. Ravichander, K. Chandu, K.-W. Chang, Y. Choi, and B. Y. Lin, "Agent lumos: Unified and modular training for open-source language agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 12 380–12 403.

[209] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *ArXiv preprint*, vol. abs/2107.03374, 2021. [Online]. Available: https://arxiv.org/abs/2107.03374

[210] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, 2019, pp. 4791–4800. [Online]. Available: https://aclanthology.org/P19-1472

[211] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1601–1611. [Online]. Available: https://aclanthology.org/P17-1147

[212] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "BoolQ: Exploring the surprising difficulty of natural yes/no questions," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 2924–2936. [Online]. Available: https://aclanthology.org/N19-1300

[213] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *ArXiv preprint*, vol. abs/2110.14168, 2021. [Online]. Available: https://arxiv.org/abs/2110.14168

[214] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=d7KBjmI3GmQ

[215] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. Le, E. Chi, D. Zhou, and J. Wei, "Challenging BIG-bench tasks and whether chain-of-thought can solve them," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 13 003–13 051. [Online]. Available: https://aclanthology.org/2023.findings-acl.824

[216] V. Subramaniam, A. Torralba, and S. Li, "Debategpt: Fine-tuning large language models with multi-agent debate supervision," 2024.

[217] B. Chen, C. Shu, E. Shareghi, N. Collier, K. Narasimhan, and S. Yao, "Fireact: Toward language agent fine-tuning," *ArXiv preprint*, vol. abs/2310.05915, 2023. [Online]. Available: https://arxiv.org/abs/2310.05915

[218] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

[219] Z. Tao, T.-E. Lin, X. Chen, H. Li, Y. Wu, Y. Li, Z. Jin, F. Huang, D. Tao, and J. Zhou, "A survey on self-evolution of large language models," *ArXiv preprint*, vol. abs/2404.14387, 2024. [Online]. Available: https://arxiv.org/abs/2404.14387

[220] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 13 484–13 508. [Online]. Available: https://aclanthology.org/2023.acl-long.754

[221] Z. Chen, Y. Deng, H. Yuan, K. Ji, and Q. Gu, "Self-play fine-tuning converts weak language models to strong language models," *ArXiv preprint*, vol. abs/2401.01335, 2024. [Online]. Available: https://arxiv.org/abs/2401.01335

[222] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[223] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Advances in Neural Information*

Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html

[224] M. G. Azar, Z. D. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello, "A general theoretical paradigm to understand learning from human preferences," in *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, ser. Proceedings of Machine Learning Research, S. Dasgupta, S. Mandt, and Y. Li, Eds., vol. 238. PMLR, 2024, pp. 4447–4455. [Online]. Available: https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html

[225] Q. Zhao, H. Fu, C. Sun, and G. Konidaris, "Epo: Hierarchical llm agents with environment preference optimization," *ArXiv preprint*, vol. abs/2408.16090, 2024. [Online]. Available: https://arxiv.org/abs/2408.16090

[226] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, "Kto: Model alignment as prospect theoretic optimization," *ArXiv preprint*, vol. abs/2402.01306, 2024. [Online]. Available: https://arxiv.org/abs/2402.01306

[227] W. Shi, M. Yuan, J. Wu, Q. Wang, and F. Feng, "Direct multi-turn preference optimization for language agents," *ArXiv preprint*, vol. abs/2406.14868, 2024. [Online]. Available: https://arxiv.org/abs/2406.14868

[228] J. Hong, N. Lee, and J. Thorne, "Orpo: Monolithic preference optimization without reference model," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 11 170–11 189.

[229] X. Liang, C. Chen, J. Wang, Y. Wu, Z. Fu, Z. Shi, F. Wu, and J. Ye, "Robust preference optimization with provable noise tolerance for llms," *ArXiv preprint*, vol. abs/2404.04102, 2024. [Online]. Available: https://arxiv.org/abs/2404.04102

[230] Y. Guo, G. Cui, L. Yuan, N. Ding, Z. Sun, B. Sun, H. Chen, R. Xie, J. Zhou, Y. Lin *et al.*, "Controllable preference optimization: Toward controllable multi-objective alignment," *ArXiv preprint*, vol. abs/2402.19085, 2024. [Online]. Available: https://arxiv.org/abs/2402.19085

[231] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li, "Knowledge editing for large language models: A survey," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–37, 2024.

[232] P. Wang, Z. Li, N. Zhang, Z. Xu, Y. Yao, Y. Jiang, P. Xie, F. Huang, and H. Chen, "Wise: Rethinking the knowledge memory for lifelong model editing of large language models," *ArXiv preprint*, vol. abs/2405.14768, 2024. [Online]. Available: https://arxiv.org/abs/2405.14768

[233] T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi, "Aging with grace: Lifelong model editing with key-value adaptors."

[234] K. Lee, W. Han, S.-w. Hwang, H. Lee, J. Park, and S.-W. Lee, "Plug-and-play adaptation for continuously-updated QA," in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 438–447. [Online]. Available: https://aclanthology.org/2022.findings-acl.37

[235] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, J. Ji, G. Cao, D. Jiang, and M. Zhou, "K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, 2021, pp. 1405–1418. [Online]. Available: https://aclanthology.org/2021.findings-acl.121

[236] J. Li, Q. Wang, Z. Wang, Y. Zhang, and Z. Mao, "Enhance lifelong model editing with continuous data-adapter association," *ArXiv preprint*, vol. abs/2408.11869, 2024. [Online]. Available: https://arxiv.org/abs/2408.11869

[237] C. Hu, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Wilke: Wise-layer knowledge editor for lifelong knowledge editing," *ArXiv preprint*, vol. abs/2402.10987, 2024. [Online]. Available: https://arxiv.org/abs/2402.10987

[238] J.-Y. Ma, H. Wang, H.-X. Xu, Z.-H. Ling, and J.-C. Gu, "Perturbation-restrained sequential model editing," *ArXiv preprint*, vol. abs/2405.16821, 2024. [Online]. Available: https://arxiv.org/abs/2405.16821

[239] A. M. Albano, J. Muench, C. Schwartz, A. Mees, and P. Rapp, "Singular-value decomposition and the grassberger-procaccia algorithm," *Physical review A*, vol. 38, no. 6, p. 3017, 1988.

[240] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[241] Y. Lin, H. Lin, W. Xiong, S. Diao, J. Liu, J. Zhang, R. Pan, H. Wang, W. Hu, H. Zhang *et al.*, "Mitigating the alignment tax of rlhf," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 580–606.

[242] H. Zhang, L. Gui, Y. Lei, Y. Zhai, Y. Zhang, Y. He, H. Wang, Y. Yu, K.-F. Wong, B. Liang *et al.*, "Copr: Continual human preference learning via optimal policy regularization," *ArXiv preprint*, vol. abs/2402.14228, 2024. [Online]. Available: https://arxiv.org/abs/2402.14228

[243] H. Zhang, Y. Lei, L. Gui, M. Yang, Y. He, H. Wang, and R. Xu, "Cppo: Continual learning for reinforcement learning with human feedback," in *The Twelfth International Conference on Learning Representations*, 2024.

[244] Y. Lu, H. Yu, and D. Khashabi, "GEAR: Augmenting language models with generalizable and efficient tool resolution," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Y. Graham and M. Purver, Eds. St. Julian's, Malta: Association for Computational Linguistics, 2024, pp. 112–138. [Online]. Available: https://aclanthology.org/2024.eacl-long.7

[245] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, dahai li, Z. Liu, and M. Sun, "ToolLLM: Facilitating large language models to master 16000+ real-world APIs," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=dHng2O0Jjr

[246] S. Yuan, K. Song, J. Chen, X. Tan, Y. Shen, K. Ren, D. Li, and D. Yang, "EASYTOOL: Enhancing LLM-based agents with concise tool instruction," in *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. [Online]. Available: https://openreview.net/forum?id=3TuG3S68bb

[247] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro, "Art: Automatic multi-step reasoning and tool-use for large language models," 2023. [Online]. Available: https://arxiv.org/abs/2303.09014

[248] Q. Xu, F. Hong, B. Li, C. Hu, Z. Chen, and J. Zhang, "On the tool manipulation capability of open-sourced large language models," 2024. [Online]. Available: https://openreview.net/forum?id=iShM3YolRY

[249] B. Wang, H. Fang, J. Eisner, B. Van Durme, and Y. Su, "LLMs in the imaginarium: Tool learning through simulated trial and error," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 10 583–10 604. [Online]. Available: https://aclanthology.org/2024.acl-long.570

[250] S. Gao, Z. Shi, M. Zhu, B. Fang, X. Xin, P. Ren, Z. Chen, J. Ma, and Z. Ren, "Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 18 030–18 038. [Online]. Available: https://doi.org/10.1609/aaai.v38i16.29759

[251] T. Cai, X. Wang, T. Ma, X. Chen, and D. Zhou, "Large language models as tool makers," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=qV83K9d5WB

[252] S. Hao, T. Liu, Z. Wang, and Z. Hu, "Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html

[253] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html

[254] X. Liu, Z. Peng, X. Yi, X. Xie, L. Xiang, Y. Liu, and D. Xu, "Toolnet: Connecting large language models with massive tools via tool graph," 2024. [Online]. Available: https://arxiv.org/abs/2403.00839

[255] W. Shen, C. Li, H. Chen, M. Yan, X. Quan, H. Chen, J. Zhang, and F. Huang, "Small llms are weak tool learners: A multi-llm agent," *arXiv e-prints*, pp. arXiv–2401, 2024.

[256] P. Sodhi, S. Branavan, Y. Artzi, and R. McDonald, "Step: Stacked LLM policies for web actions," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=5fg0VtRxgi

[257] Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, and V. Tresp, "Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration," 2024. [Online]. Available: https://arxiv.org/abs/2408.15978

[258] D. Zhang, L. Chen, S. Zhang, H. Xu, Z. Zhao, and K. Yu, "Large language models are semi-parametric reinforcement learning agents," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/f6b22ac37beb5da61efd4882082c9ecd-Abstract-Conference.html

[259] K. Ma, H. Zhang, H. Wang, X. Pan, and D. Yu, "LASER: LLM agent with state-space exploration for web navigation," in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. [Online]. Available: https://openreview.net/forum?id=sYFFyAILy7

[260] G. Sarch, L. Jang, M. J. Tarr, W. W. Cohen, K. Marino, and K. Fragkiadaki, "Vlm agents generate their own memories: Distilling experience into embodied programs," 2024. [Online]. Available: https://arxiv.org/abs/2406.14596

[261] Z. Zhao, W. Chai, X. Wang, B. Li, S. Hao, S. Cao, T. Ye, and G. Wang, "See and think: Embodied agent in virtual environment," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 187–204.

[262] W. Tan, W. Zhang, X. Xu, H. Xia, Z. Ding, B. Li, B. Zhou, J. Yue, J. Jiang, Y. Li, R. An, M. Qin, C. Zong, L. Zheng, Y. Wu, X. Chai, Y. Bi, T. Xie, P. Gu, X. Li, C. Zhang, L. Tian, C. Wang, X. Wang, B. F. Karlsson, B. An, S. Yan, and Z. Lu, "Cradle: Empowering foundation agents towards general computer control," 2024. [Online]. Available: https://arxiv.org/abs/2403.03186

[263] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 9118–9147. [Online]. Available: https://proceedings.mlr.press/v162/huang22a.html

[264] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html

[265] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, and Z. Hu, "Reasoning with language model is planning with world model," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 8154–8173. [Online]. Available: https://aclanthology.org/2023.emnlp-main.507

[266] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/65a39213d7d0e1eb5d192aa77e77eeb7-Abstract-Conference.html

[267] B. Y. Lin, Y. Fu, K. Yang, F. Brahman, S. Huang, C. Bhagavatula, P. Ammanabrolu, Y. Choi, and X. Ren, "Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/4b0eea69deea512c9e2c469187643dc2-Abstract-Conference.html

[268] M. Li, Y. Zhao, B. Yu, F. Song, H. Li, H. Yu, Z. Li, F. Huang, and Y. Li, "API-bank: A comprehensive benchmark for tool-augmented LLMs," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 3102–3116. [Online]. Available: https://aclanthology.org/2023.emnlp-main.187

[269] A. Prasad, A. Koller, M. Hartmann, P. Clark, A. Sabharwal, M. Bansal, and T. Khot, "ADaPT: As-needed decomposition and planning with language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, K. Duh, H. Gomez, and S. Bethard, Eds. Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 4226–4252. [Online]. Available: https://aclanthology.org/2024.findings-naacl.264

[270] M. Hu, Y. Mu, X. Yu, M. Ding, S. Wu, W. Shao, Q. Chen, B. Wang, Y. Qiao, and P. Luo, "Tree-planner: Efficient close-loop task planning with large language models," *ArXiv preprint*, vol. abs/2310.08582, 2023. [Online]. Available: https://arxiv.org/abs/2310.08582

[271] G. Chen, Z. Zhang, X. Cong, F. Guo, Y. Wu, Y. Lin, W. Feng, and Y. Wang, "Learning evolving tools for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2410.06617

[272] A. Gui, J. Li, Y. Dai, N. Du, and H. Xiao, "Look before you leap: Towards decision-aware and generalizable tool-usage for large language models," *ArXiv preprint*, vol. abs/2402.16696, 2024. [Online]. Available: https://arxiv.org/abs/2402.16696

[273] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, "Planning with large language models via corrective re-prompting," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. [Online]. Available: https://openreview.net/forum?id=cMDMRBe1TKs

[274] T. R. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, "Cognitive architectures for language agents," 2023. [Online]. Available: https://arxiv.org/abs/2309.02427

[275] Z. Guo, S. Cheng, H. Wang, S. Liang, Y. Qin, P. Li, Z. Liu, M. Sun, and Y. Liu, "StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 11 143–11 156. [Online]. Available: https://aclanthology.org/2024.findings-acl.664

[276] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez, "Gorilla: Large language model connected with massive APIs," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=tBRNC6YemY

[277] Q. Tang, Z. Deng, H. Lin, X. Han, Q. Liang, B. Cao, and L. Sun, "Toolalpaca: Generalized tool learning for language models with 3000 simulated cases," 2023. [Online]. Available: https://arxiv.org/abs/2306.05301

[278] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November*

28 - December 9, 2022, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html

[279] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 8494–8502. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Puig_VirtualHome_Simulating_Household_CVPR_2018_paper.html

[280] PrismarineJS, "Mineflayer: Create minecraft bots with a powerful, stable, and high level javascript api," https://github.com/PrismarineJS/mineflayer, 2024, accessed: 2024-11-24. [Online]. Available: https://github.com/PrismarineJS/mineflayer

[281] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.9

[282] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2312.10997

[283] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-tuning or retrieval? comparing knowledge injection in LLMs," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 237–250. [Online]. Available: https://aclanthology.org/2024.emnlp-main.15

[284] J. Guan, W. Wu, Z. Wen, P. Xu, H. Wang, and M. Huang, "Amor: A recipe for building adaptable modular knowledge agents through process feedback," *ArXiv preprint*, vol. abs/2402.01469, 2024. [Online]. Available: https://arxiv.org/abs/2402.01469

[285] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[286] G. Kim, S. Kim, B. Jeon, J. Park, and J. Kang, "Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 996–1009. [Online]. Available: https://aclanthology.org/2023.emnlp-main.63

[287] Z. Feng, X. Feng, D. Zhao, M. Yang, and B. Qin, "Retrieval-generation synergy augmented large language models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 661–11 665.

[288] X. Ye, R. Sun, S. Ö. Arik, and T. Pfister, "Effective large language model adaptation for improved grounding," *ArXiv preprint*, vol. abs/2311.09533, 2023. [Online]. Available: https://arxiv.org/abs/2311.09533

[289] X. Cheng, D. Luo, X. Chen, L. Liu, D. Zhao, and R. Yan, "Lift yourself up: Retrieval-augmented text generation with self-memory," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/887262aeb3eafb01ef0fd0e3a87a8831-Abstract-Conference.html

[290] J. Y. Koh, S. McAleer, D. Fried, and R. Salakhutdinov, "Tree search for language model agents," *ArXiv preprint*, vol. abs/2407.01476, 2024. [Online]. Available: https://arxiv.org/abs/2407.01476

[291] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=rJ4km2R5t7

[292] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 3261–3275. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html

[293] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 3470–3487. [Online]. Available: https://aclanthology.org/2022.acl-long.244

[294] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, E. Pathak, G. Karamanolakis, H. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, K. K. Pal, M. Patel, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. Doshi, S. K. Sampat, S. Mishra, S. Reddy A, S. Patro, T. Dixit, and X. Shen, "Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 5085–5109. [Online]. Available: https://aclanthology.org/2022.emnlp-main.340

[295] P. Koehn, "Statistical and neural machine translation," https://www.statmt.org/, accessed: 2024-12-05.

[296] K. Duh, "The multitarget ted talks task," http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/, 2018.

[297] H. Zhang, S. Zhang, Y. Xiang, B. Liang, J. Su, Z. Miao, H. Wang, and R. Xu, "CLLE: A benchmark for continual language learning evaluation in multilingual machine translation," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 428–443. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.30

[298] N. De Cao, W. Aziz, and I. Titov, "Editing factual knowledge in language models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 6491–6506. [Online]. Available: https://aclanthology.org/2021.emnlp-main.522

[299] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 809–819. [Online]. Available: https://aclanthology.org/N18-1074

[300] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html

[301] J. Zheng, S. Qiu, and Q. Ma, "Concept-1k: A novel benchmark for instance incremental learning," *ArXiv preprint*, vol. abs/2402.08526, 2024. [Online]. Available: https://arxiv.org/abs/2402.08526

[302] Z. Allen-Zhu and Y. Li, "Physics of language models: Part 3.2, knowledge manipulation," *ArXiv preprint*, vol. abs/2309.14402, 2023. [Online]. Available: https://arxiv.org/abs/2309.14402

[303] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim,

C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *ArXiv preprint*, vol. abs/2112.09332, 2021. [Online]. Available: https://arxiv.org/abs/2112.09332

[304] R. Hao, L. Hu, W. Qi, Q. Wu, Y. Zhang, and L. Nie, "Chatllm network: More brains, more intelligence," *ArXiv preprint*, vol. abs/2304.12998, 2023. [Online]. Available: https://arxiv.org/abs/2304.12998

[305] Z. Lin, P. Xu, G. I. Winata, F. B. Siddique, Z. Liu, J. Shin, and P. Fung, "Caire: An empathetic neural chatbot," *ArXiv preprint*, vol. abs/1907.12108, 2019. [Online]. Available: https://arxiv.org/abs/1907.12108

[306] J.-H. Jhan, C.-P. Liu, S.-K. Jeng, and H.-Y. Lee, "Cheerbots: Chatbots toward empathy and emotionusing reinforcement learning," *ArXiv preprint*, vol. abs/2110.03949, 2021. [Online]. Available: https://arxiv.org/abs/2110.03949

[307] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu, "Chateval: Towards better llm-based evaluators through multi-agent debate," *ArXiv preprint*, vol. abs/2308.07201, 2023. [Online]. Available: https://arxiv.org/abs/2308.07201

[308] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, J. Yang *et al.*, "Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models," *ArXiv preprint*, vol. abs/2310.00746, 2023. [Online]. Available: https://arxiv.org/abs/2310.00746

[309] Y. Dai, H. Hu, L. Wang, S. Jin, X. Chen, and Z. Lu, "Mmrole: A comprehensive framework for developing and evaluating multimodal role-playing agents," *ArXiv preprint*, vol. abs/2408.04203, 2024. [Online]. Available: https://arxiv.org/abs/2408.04203

[310] M. T. Ng, H. T. Tse, J.-t. Huang, J. Li, W. Wang, and M. R. Lyu, "How well can llms echo us? evaluating ai chatbots' role-play ability with echo," *ArXiv preprint*, vol. abs/2404.13957, 2024. [Online]. Available: https://arxiv.org/abs/2404.13957

[311] Y. Wu, S. Y. Min, Y. Bisk, R. Salakhutdinov, A. Azaria, Y. Li, T. Mitchell, and S. Prabhumoye, "Plan, eliminate, and track–language models are good teachers for embodied agents," *ArXiv preprint*, vol. abs/2305.02412, 2023. [Online]. Available: https://arxiv.org/abs/2305.02412

[312] M. Gramopadhye and D. Szafir, "Generating executable action plans with environmentally-aware language models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3568–3575.

[313] P.-L. Chen and C.-S. Chang, "Interact: Exploring the potentials of chatgpt as a cooperative agent," *ArXiv preprint*, vol. abs/2308.01552, 2023. [Online]. Available: https://arxiv.org/abs/2308.01552

[314] P. BAAI, "Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks," *ArXiv preprint*, vol. abs/2303.16563, 2023. [Online]. Available: https://arxiv.org/abs/2303.16563

[315] K. Nottingham, P. Ammanabrolu, A. Suhr, Y. Choi, H. Hajishirzi, S. Singh, and R. Fox, "Do embodied agents dream of pixelated sheep: Embodied decision making using language guided world modelling," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 26 311–26 325. [Online]. Available: https://proceedings.mlr.press/v202/nottingham23a.html

[316] Z. Wang, "Mediagpt: A large language model target chinese media," *ArXiv preprint*, vol. abs/2307.10930, 2023. [Online]. Available: https://arxiv.org/abs/2307.10930

[317] "Databricks - media entertainment solutions," https://www.databricks.com/solutions/industries/media-and-entertainment.

[318] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. Basilico, "Deep learning for recommender systems: A netflix case study," *AI Magazine*, vol. 42, no. 3, pp. 7–18, 2021.

[319] S. Jinxin, Z. Jiabao, W. Yilei, W. Xingjiao, L. Jiawen, and H. Liang, "Cgmi: Configurable general multi-agent interaction framework," *ArXiv preprint*, vol. abs/2308.12503, 2023. [Online]. Available: https://arxiv.org/abs/2308.12503

[320] J. Su and W. Yang, "Unlocking the power of chatgpt: A framework for applying generative ai in education," *ECNU Review of Education*, vol. 6, no. 3, pp. 355–366, 2023.

[321] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Qian, C.-M. Chan, Y. Qin, Y. Lu, R. Xie *et al.*, "Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents," *ArXiv preprint*, vol. abs/2308.10848, 2023. [Online]. Available: https://arxiv.org/abs/2308.10848

[322] M. Swan, T. Kido, E. Roland, and R. P. d. Santos, "Math agents: Computational infrastructure, mathematical embedding, and genomics," *ArXiv preprint*, vol. abs/2307.02502, 2023. [Online]. Available: https://arxiv.org/abs/2307.02502

[323] V. Kalvakurthi, A. S. Varde, and J. Jenq, "Hey dona! can you help me with student course registration?" *ArXiv preprint*, vol. abs/2303.13548, 2023. [Online]. Available: https://arxiv.org/abs/2303.13548

[324] S. Hamilton, "Blind judgement: Agent-based supreme court modelling with gpt," *ArXiv preprint*, vol. abs/2301.05327, 2023. [Online]. Available: https://arxiv.org/abs/2301.05327

[325] F. Yu, L. Quartey, and F. Schilder, "Legal prompting: Teaching a language model to think like a lawyer," *ArXiv preprint*, vol. abs/2212.01326, 2022. [Online]. Available: https://arxiv.org/abs/2212.01326

[326] R. Shui, Y. Cao, X. Wang, and T.-S. Chua, "A comprehensive evaluation of large language models on legal judgment prediction," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 7337–7348. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.490

[327] H. Zhang, J. Chen, F. Jiang, F. Yu, Z. Chen, G. Chen, J. Li, X. Wu, Z. Zhiyi, Q. Xiao, X. Wan, B. Wang, and H. Li, "HuatuoGPT, towards taming language model to be a doctor," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, 2023, pp. 10 859–10 885. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.725

[328] S. Yang, H. Zhao, S. Zhu, G. Zhou, H. Xu, Y. Jia, and H. Zan, "Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 19 368–19 376. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29907

[329] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical artificial intelligence," *Nature*, vol. 616, no. 7956, pp. 259–265, 2023.

[330] S. M. Jungmann, T. Klan, S. Kuhn, and F. Jungmann, "Accuracy of a chatbot (ada) in the diagnosis of mental disorders: comparative case study with lay and expert users," *JMIR formative research*, vol. 3, no. 4, p. e13863, 2019.

[331] M. R. Ali, S. Z. Razavi, R. Langevin, A. Al Mamun, B. Kane, R. Rawassizadeh, L. K. Schubert, and E. Hoque, "A virtual conversational agent for teens with autism spectrum disorder: Experimental results and design lessons," in *Proceedings of the 20th ACM international conference on intelligent virtual agents*, 2020, pp. 1–8.

[332] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *ArXiv preprint*, vol. abs/2303.17564, 2023. [Online]. Available: https://arxiv.org/abs/2303.17564

[333] W. Gao, X. Gao, and Y. Tang, "Multi-turn dialogue agent as sales' assistant in telemarketing," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–9.

[334] M. J. North and C. M. Macal, *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*. Oxford University Press, 2007.

[335] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the national academy of sciences*, vol. 99, no. suppl_3, pp. 7280–7287, 2002.

[336] H. Chen, R. Pasunuru, J. Weston, and A. Celikyilmaz, "Walking down the memory maze: Beyond context limit through interactive reading," *ArXiv preprint*, vol. abs/2310.05029, 2023. [Online]. Available: https://arxiv.org/abs/2310.05029

[337] J. Zhao, C. Zu, H. Xu, Y. Lu, W. He, Y. Ding, T. Gui, Q. Zhang, and X. Huang, "Longagent: Scaling language models to 128k context through multi-agent collaboration,"

ArXiv preprint, vol. abs/2402.11550, 2024. [Online]. Available: https://arxiv.org/abs/2402.11550

[338] G. Chen, X. Li, Z. Meng, S. Liang, and L. Bing, "Clex: Continuous length extrapolation for large language models," *ArXiv preprint*, vol. abs/2310.16450, 2023. [Online]. Available: https://arxiv.org/abs/2310.16450

[339] Y. Sun, L. Dong, B. Patra, S. Ma, S. Huang, A. Benhaim, V. Chaudhary, X. Song, and F. Wei, "A length-extrapolatable transformer," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 14 590–14 604. [Online]. Available: https://aclanthology.org/2023.acl-long.816

[340] J. Xu, A. Szlam, and J. Weston, "Beyond goldfish memory: Long-term open-domain conversation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 5180–5197. [Online]. Available: https://aclanthology.org/2022.acl-long.356

[341] B. P. Woolf, *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.

[342] A. L. Soller and A. Lesgold, "A computational approach to analyzing online knowledge sharing interaction," in *Proceedings of Artificial Intelligence in Education*, 2003.

[343] B.-C. Kuo, F. T. Chang, and Z.-E. Bai, "Leveraging llms for adaptive testing and learning in taiwan adaptive learning platform (talp)." in *LLM@ AIED*, 2023, pp. 101–110.

[344] J. Prather, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly *et al.*, "The robots are here: Navigating the generative ai revolution in computing education," in *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, 2023, pp. 108–159.

[345] A. Haleem, M. Javaid, and R. P. Singh, "An era of chatgpt as a significant futuristic support tool: A study on features, abilities, and challenges," *BenchCouncil transactions on benchmarks, standards and evaluations*, vol. 2, no. 4, p. 100089, 2022.

[346] H. Crompton and D. Burke, "Artificial intelligence in higher education: the state of the field," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 22, 2023.

[347] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[348] S. Doveh, A. Arbelle, S. Harary, E. Schwartz, R. Herzig, R. Giryes, R. Feris, R. Panda, S. Ullman, and L. Karlinsky, "Teaching structured vision & language concepts to vision & language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 2657–2668. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.00261

[349] S. Saha, P. Hase, and M. Bansal, "Can language models teach weaker agents? teacher explanations improve students via theory of mind," *ArXiv preprint*, vol. abs/2306.09299, 2023. [Online]. Available: https://arxiv.org/abs/2306.09299

[350] T. Bench-Capon and G. Sartor, "A model of legal reasoning with cases incorporating theories and values," *Artificial Intelligence*, vol. 150, no. 1-2, pp. 97–143, 2003.

[351] L. K. Branting, *Reasoning with rules and precedents: a computational model of legal analysis*. Springer Science & Business Media, 2013.

[352] K. Y. Iu and V. M.-Y. Wong, "Chatgpt by openai: The end of litigation lawyers?" *Available at SSRN 4339839*, 2023.

[353] F. C. Kitamura, "Chatgpt is shaping the future of medical writing but still requires human judgment," p. e230171, 2023.

[354] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, M. Madriaga, R. Aggabao, G. Diaz-Candido, J. Maningo *et al.*, "Performance of chatgpt on usmle: potential for ai-assisted medical education using large language models," *PLoS digital health*, vol. 2, no. 2, p. e0000198, 2023.

[355] M. Sallam, "Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns," in *Healthcare*, vol. 11, no. 6. MDPI, 2023, p. 887.

[356] M. Cascella, J. Montomoli, V. Bellini, and E. Bignami, "Evaluating the feasibility of chatgpt in healthcare: an analysis of multiple clinical and research scenarios," *Journal of medical systems*, vol. 47, no. 1, p. 33, 2023.

[357] P. Malik, M. Pathania, V. K. Rathaur *et al.*, "Overview of artificial intelligence in medicine," *Journal of family medicine and primary care*, vol. 8, no. 7, pp. 2328–2331, 2019.

[358] Y. Peng, J. Qi, Z. Ye, and Y. Zhuo, "Hierarchical visual-textual knowledge distillation for life-long correlation learning," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 921–941, 2021.

[359] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[360] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.