Intelligent Digital Agents in the Era of Large Language Models

Benjamin Faught*[1], Helen Lu*[12], Thomas Marshall*[1], Harshvardhan Sikka*+[134], Pranav Guruprasad[1], Bhavul Gauri[15]

[1]Manifold Research
[2]Computer Science Department, Tufts University
[3]Metarch.ai
[4]School of Interactive Computing, Georgia Institute of Technology
[5]Meta

***Abstract*—In recent years, the emergence of large language models (LLMs) has revolutionized the field of artificial intelligence, showcasing remarkable proficiency in natural language understanding and generation. This advancement has spurred a growing research area focused on the development of LLM-based autonomous agents, aiming to achieve human-like decision-making capabilities. This Position Paper provides an overview of current research areas and breakthroughs in LLM-based AI agents. We highlight key advancements and discuss limitations within each area. Furthermore, we identify opportunities for future research to accelerate progress towards realizing more capable versions of AI agents.**

I. INTRODUCTION

Autonomous agents have long represented a cornerstone in the pursuit of artificial intelligence, embodying the capacity for independent decision-making and action execution. Over recent years, the emergence of large language models (LLMs) has ushered in a new era of AI capabilities. These LLMs, powered by vast datasets and advanced deep learning architectures, have showcased remarkable proficiency in natural language understanding and generation, approaching human-like levels of intelligence. Capitalizing on the capabilities of LLMs, researchers have ventured into a burgeoning research area: the development of LLM-based autonomous agents. By employing LLMs as central controllers, these agents aim to achieve human-like decision-making capabilities, navigating complex environments and interacting with users in natural language. This convergence of LLMs and autonomous agents marks a significant milestone in AI research, opening up new avenues for the creation of intelligent systems capable of understanding and engaging with humans in meaningful ways.Various definitions of AI Agents exist. For example, according to [16] an artificial intelligence (AI) agent is a discerning entity capable of sensory perception, decision-making, and undertaking actions to fulfill predetermined objectives. Distinct from an autonomous agent, which operates within the perceive-decide-act cycle sans human guidance, an intelligent agent possesses the supplementary capacity to enhance its functionality through the assimilation of new information [15].

* equal contribution, alphabetical order.
\+ Corresponding Author: harshsikka@gatech.edu

This paper is meant to serve as a Position Paper as opposed to a technical nor scholarly paper in the traditional sense. Our contribution is three fold.

1) First, we provide an overview of the current major research areas of LLM based AI agents and highlights key research breakthroughs from the recent literature.
2) Second, we outline limitations in each of these research areas.
3) Finally, we outline opportunities for new research that the broader community might pursue to accelerate towards realizing ever more capable versions of AI Agents.

II. RESEARCH AREAS & LIMITATIONS

A. Memory

Memory is a critical aspect of Language and Learning Models (LLMs), influencing their ability to process and understand input data effectively. In this section, we delve into the various dimensions of memory-related methods and categorizations within the current landscape of LLM research, drawing insights from recent studies and advancements.

1) Understanding Memory in LLMs: A significant challenge facing many LLM-based systems is their limited capability to process exceptionally lengthy inputs, often referred to as Memory in the context of LLM systems [7]. This limitation impedes broader generalization and hinders the performance of these models.

Memory-related methods in LLMs have been categorized into several distinct types, offering unique approaches to address memory limitations and enhance model performance.

1) **Short and Long-Term Memory**: One common categorization distinguishes between short and long-term memory mechanisms. Short-term memory methods, such as Context Learning and Prompt Engineering, primarily focus on immediate context processing. In contrast, long-term memory involves accessing external stores or databases during query time [21]. Recent research has explored integrating long-term memory directly into LLMs, as demonstrated by the LONGMEM framework, which enables LLMs to memorize extensive historical contexts effectively [18].

2) **Internal vs External Memory Based Methods**: Another useful categorization divides memory mechanisms into internal and external memory-based methods. Internal memory methods aim to optimize computational efficiency by leveraging specialized positional encoding techniques to handle longer input sequences [4]. In contrast, external memory-based methods utilize physical memory caches to store historical information, enabling seamless augmentation of LLMs without memory constraints [18].

2) Advancements in Memory-Augmented Networks (MANNs): Memory-augmented networks (MANNs), such as Neural Turing Machines (NTMs), represent a promising avenue for enhancing memory capabilities in LLMs [8]. These networks utilize external memory caches to store and manipulate information, facilitating tasks requiring long-term context interactions. Recent studies have focused on leveraging MANNs for long-term conversations, introducing datasets and mechanisms inspired by cognitive psychology principles [27].

3) Working Memory Enhancements: Enabling "Working Memory" in LLMs has been explored to enhance their architecture and improve contextual reasoning during complex tasks and collaborative scenarios [5]. Efforts have been made to address the low controllability and robustness of out-of-the-box models with regards to working memory, with proposed solutions such as knowledge-aware fine-tuning methods [6].

4) Challenges and Future Directions: Despite significant advancements, several challenges remain in developing reliable and adaptable long-term memory mechanisms for LLMs. Existing methods often require repeated reasoning over stored histories and suffer from computational inefficiencies in recalling relevant information [23]. Future research directions may involve exploring more efficient memory retrieval mechanisms, integrating cognitive psychology principles, and optimizing memory-augmented networks to further enhance LLMs' memory capabilities.

Memory plays a crucial role in the effectiveness and generalization abilities of LLM-based AI agents. By understanding and advancing memory-related methods, researchers can unlock new potentials for these agents in handling complex language tasks and real-world applications.

B. Decision Making and Reasoning

LLMs are known to hallucinate and struggle with tasks that require extensive reasoning. To deal with this issue, researchers have explored prompting techniques that guide the LLM through the reasoning steps and neuro-symbolic methods that combine LLMs with classical AI planners for better performance on long-horizon planning tasks.

1) Prompting Strategies: For example, Wei et al. prompts the LLM by showing exemplar chains of thought i.e. series of intermediate reasoning steps that solve complex problems [20]. The LLM emulates this process and generates a chain of thought for every problem that's prompted with chain-of-thought prompting, significantly increasing its accuracy to solve problems that require reasoning. Yao et al. generalize over chain of thought and introduces Tree of Thought. Tree of Thought is a framework that allows the LLM to explore different chains of thought using traditional breadth-first search or depth-first search [24]. Tree of thought significantly improves the LLM's problem solving ability on planning or search tasks. Yao et al. introduces an approach named ReAct in which the reasoning steps and acting steps are interleaved [25]. Prior to ReAct, reasoning and action plan generation have been studied mostly as separate topics. ReAct prompts the model to generate a sequence of reasoning steps, followed by an action. This approach is showcased across a variety of tasks, including multi-hop question-answering, fact-checking, and interactive decision-making. The method aims to produce more interpretable, diagnosable, and controllable task-solving trajectories than previous methods.

2) Neuro-symbolic Strategies: To utilize LLM's ability to understand natural language descriptions of tasks and classical AI planners' ability to solve complex planning tasks, some researchers have looked into using LLM as a translator to translate natural language description of tasks into a syntax that classical AI planners can use. Liu et al. connects LLMs with AI planners by using the LLM to turn the natural language description of a robot planning problem into a planning domain definition language (PDDL) that classical AI planners are able to parse [1]. The output from the planner is translated back into natural language by the LLM. This approach significantly augments the LLM's ability, leveraging both the natural language understanding capability of the LLM and the search and plan capability of AI planners.

3) Challenges and Future Directions: LLM generated plans still suffer from hallucinations in long horizon tasks. Therefore, connecting LLMs with planners capable of structured planning seems to be a promising direction worth more exploration. On the other hand, we posit that relying on natural language for solving complex planning problems has its limitations due to its limited sequence-to-sequence nature. Promising approaches to solving long horizon planning tasks pair LLMs with symbolic AI planners, essentially prompting the LLM to translate natural language description of the problem into non-human language [22]. Therefore, a future direction could be to explore other representations of thinking and reasoning. Raw internal model states is one potential representation.

C. Tool Use

Modifying or augmenting LLMs to allow for the use of tools is one of the major ways that researchers have addressed some of the limitations of LLMs. For example, retrieval of external data, lookup of updated information, arithmetic, etc. are all tasks that are important for AI agents, but that LLMs struggle with (e.g. arithmetic) or are entirely incapable of natively (e.g. lookup of updated information).

Tool use is a very broad concept which takes many different forms. We propose breaking tool-use down into a few categories:

1) Lookup of static information
   Example: LOOKUP('capital of France')

2) Lookup of Dynamic information
   Example: LOOKUP('weather in Boston')

3) Stateless function invocation
   Example: MULTIPLY(256, 172)

4) Stateful function invocation
   Example: SET('current_state', 'success')

5) World interaction
   Example: UBEREATS('order', 'burger')

Breaking down "tool-use" into these categories is important as many current works that fall under the header of "tool-use" will focus on some subset (or even just one) of these categories. But all are important for the successful integration of tools into LLM-based agents. Furthermore, each of these categories deserves to be considered conceptually distinct, even if their invocation syntax is often congruent, because their patterns of interaction are fundamentally different in ways that are specific to each category.

For example, an LLM fine-tuned for lookup of static information may not understand that subsequent queries for the weather in Boston may give different results. An LLM fine-tuned for the dynamic case may not understand that subsequent queries for the capital of France are extremely unlikely to yield any new information. Neither may understand that stateful function invocation may not return any result at all. Similarly, a tool-use agent trained to make calls to stateless functions may fail to understand the concept of stateful interaction.

1) Recent Work on Tool-Use: TALM: Tool Augmented Language Models stands as an important starting point for understanding tool-use strategies in modern language models and ultimately offers an important conceptual step forward in the development of LLM-based agents [9]. The technique involves augmenting language models with the ability to invoke non-differentiable tool APIs along with a method for updating models for tool-use via iterative self-play. Tool-use has become more sophisticated in many ways since TALM, but it proved that tool-use was possible and effective. And it ultimately sets the backdrop for discussion about the further development of tool use in AI agents. Schick et al. introduces a very similar concept in Toolformer [12]. The broader goal here is generally the same–imbue a language model with the capacity for tool-use, with specific focuses on generality and self-supervision, with the methodological distinction between these two techniques largely boiling down to implementation strategy.

Shen et al. builds upon these ideas in HuggingGPT, showing that LLMs can be used as high-level controllers to solve multi-modal tasks by leveraging machine learning APIs [13]. HuggingGPT also further augments the tool use idea by integrating a system for planning, taking inspiration from chain of thought prompting [20]. HuggingGPT is as much an experiment in multimodality as it is a demonstration of tool use. In fact, it is likely best understood as multimodality *via* tool-use. In other words, HuggingGPT moves away from the generality that TALM and Toolformer make a priority and instead places its emphasis on leveraging relatively few, but high-impact, machine learning based tools.

Patil et al. develops Gorilla to further extend the theme of API calling, specifically focusing on compatibility with a large number of machine learning APIs such as HuggingFace, TorchHub, and TensorHub [10]. If HuggingGPT showed that it was possible to train a model to leverage machine learning APIs, Gorilla showed that it was possible at scale by increasing the number of APIs that the model is able to call.

A common theme among all of these attempts at building tool-use capabale LLMs and LLM-based agents it self-supervision, and in particular, the Self-Instruct process of semi-supervised instruction tuning proposed by Wang. et al. [19]. Self-Instruct stands at the center of many important tool-use examples. It provides a significant head start by allowing for self-supervised instruction data to drive fine-tuning, but it has significant limitations which these tool-use demonstrations inherit, including the reliance on language generation as the mechanism for generation of self-supervised tool-use examples.

Some of the most interesting successes in the world of tool-use have been in its application to domain specific problems. Bran et al. demonstrates this in the chemistry domain by developing ChemCrow [2]. In this case, and unlike above, the focus here is not breadth, but rather hand-crafting specific tools designed by experts which are then coordinated using ReAct-style prompting [25]. Interestingly, ChemCrow demonstrates superior performance to GPT-4 on novel or less well-known problems, while lagging behind on problems involving rote memorization [2].

The successes of tool-use agents in reasoning over domain-specific applications shows that the benefits of tool-use are not limited to trivial extensions of LLMs; it is not simply advantageous in extending or updating the knowledge available to an LLM. Instead, tool-use appears capable of legitimately enhancing the reasoning capacity of intelligent agents. All of these capabilities taken together point strongly to tool-use being one of the most important directions forward in the field of AI agent research.

2) Challenges and Limitations: While current SOTA tool-use demonstrations have good coverage of the first four proposed tool-use categories, it is much harder to find robust examples of the last two. This is likely because stateful function invocation and world interaction both require an LLM to maintain an internal representation of an external system, making these two categories harder, and arguably even, a *different problem entirely*. Nevertheless, tackling these two cases is critical for the maturation of tool-use agents.

We also identify the reliance on language as a potential weak point of current tool-use methods. Training LLMs to produce the correct utterance in order to invoke a tool can be effective in simple cases, but the inherently sequential nature of this strategy may be insufficient for more challenging tool-use cases. On top of this, tools may deal with multimodal data processing that language models will naturally have challenges in dealing with.

D. Learning

One characteristic that distinguishes intelligent agents from autonomous agents is their capacity for continuous learning. Intelligent agents possess the capability to not merely engage with their environment but also to observe the effects of their engagement and adapt accordingly. A clear challenge between modern LLM based agents and this pattern of learning is that large models are effectively frozen: their weights do not adapt as agents interact with their environment. This is a notable challenge that has yet to be circumvented in most LLM Agent research.

1) Learning and Memory: One approach to learning is to learn persistent memories to affect future decision-making. This approach is demonstrated by Shinn et al., who propose Reflexion - a framework for language-based agents, that converts binary or scalar feedback from the environment that the agent interacts with, into verbal feedback in the form of a textual summary. This framework aims to reinforce the language agents without updating weights. The agents that are augmented with the Reflexion framework, reflect on their task's feedback signals and maintain a verbal feedback version of the signals in an episodic memory buffer, which is then used to improve decision making in subsequent trials [14]. Zhao et al. showcase an agent that is able to autonomously gather experiences and extract knowledge using natural language from a collection of training tasks [26]. At inference, the agent recalls its extracted insights and past experiences to make informed decisions. The agent gathers diverse experiences though Reflexion mentioned earlier and continuously retrying tasks until success. The agent implements a form of retrospective learning similar to Reflexion.

2) Tasks: Another approach is to learn tasks directly. While many agents use modular tools that are assumed to complete the tasks assigned to them reliably and optimally, there are many tasks that have no such guarantees. Tasks that use reinforcement learning (RL) often require a complicated and extensive series of actions to be successfully completed. More formally, they display high *sparseness*, which measures the average number of steps needed, when executing random actions sampled from a uniform distribution, to solve the given task and receive reward. Di Palo, Norman, et al. demonstrate the utility of LLMs and vision language models (VLM)–which together they term *foundation models*–towards learning RL tasks, as well as transferring learned experiences and skills towards new tasks [3]. Regarding exploring new tasks, the foundation models allow for the agent's number of steps needed to achieve a certain success rate (or its learning rate) to grow more slowly than the sparseness of the task. The models accomplish this by decomposing tasks into subtasks, so while the overall task may show exponentially increasing sparseness, the subtasks themselves do not (when completed in order) and are relatively attainable by the RL agent. Task transfer learning is brought into play by storing policies learned for given subtasks–or skills–into memory, then allowing for the foundation models to decompose a task into subtasks for which there are corresponding skills. If the agent has learned the relevant skill, then it executes it. This can be done for wholly new problems, if the problem can be decomposed into tasks and skills that the agent has already learned.

Wang, Guanzhi, et al. take a similar approach with an embodied agent that navigates the Minecraft virtual world [17]. The Voyager agent uses LLM-prompted automatic curriculum to generate attainable tasks, and a skill library for storing and retrieving complex behaviors. Additionally, the agent uses an iterative prompting mechanism to generate executable code for embodied control.

While many agents rely on a LLM-based planner to choose actions and tools to execute those actions, these two agents demonstrate methods by which the action-space can be more thoroughly explored, and learned skills can be more effectively leveraged. The potential is there for agents to learn through experimentation how best to leverage the tools at their disposal or even to create new ones, rather than relying on the planner to dictate actions. However, both agents assume the knowledge available to the LLM will be sufficient to direct tasking. As mentioned earlier, moving beyond a traditionally language only paradigm for planning may also be beneficial to planning.

3) External Data Stores: While most agents rely on LLMs for planning and knowledge retrieval, retrieval learning has played a significant role in agent success and capacity to learn before the outbreak of LLM approaches. XiaoIce is a social chatbot intended to facilitate the formation of long-term emotional connections from its users towards itself [28]. It is designed with over two hundred task-solving skills (considered to emulate "IQ") as well as components that gauge emotional content ("EQ") of its users' and its own responses, all constrained by the need to maintain a consistent personality. XiaoIce models its hierarchical decision-making processes as Markov Decision Processes (MDP), where each turn in dialogue is a state and a skill or a response represents an action to be taken at that state. Of especial note is its Core Chat component, which combines a retrieval-based generator (which enables XiaoIce to consistently plagiarize itself as its continued usage leads produces more and more data) for high-quality responses to popular topics, with a complementary GRU- RNN model-backed neural response generator for robustness and high coverage. This retrieval system distinguishes it from its LLM-backed peers in that it allows for XiaoIce to retrieve data from dynamic sources, while LLMs are frozen. While it may not be wholly fair to consider changing data as learning, the effect is still similar in that the system can stay abreast of recent developments in topics of interest.

III. Moving Beyond Language Presented Opportunities for AI Agents

In this paper, we have explored some of the major trends in LLM-based agent research and highlighted some limitations of this kind of agent. We propose that AI agents may need to move beyond language and autoregressive token generation to more fully realize the reasoning potential of these agents and to address these limitations.

Recent trends in LLM-based agent research illuminate the inadequacies of language as the sole modality for internal reasoning. Techniques such as chain-of-thought or tree-of-thought exemplify innovative strategies to navigate beyond language's linear constraints, aiming to mimic more complex reasoning pathways that are not inherently sequential or symbolic.

Moreover, the application of LLMs to generate outputs beyond the realm of language—such as sequences that invoke tool-use systems or sequences for planning algorithms—underscores that researchers are increasingly leveraging LLMs to produce outputs that, while originated from language models, serve as inputs for non-linguistic processing. This transition from generating purely linguistic content to creating information that aligns more closely with other computational paradigms signals a recognition of language's limitations.

Furthermore, the transition towards multimodality in LLMs and LLM-based agents signifies more than the integration of diverse sensory inputs; it signifies a fundamental transformation in AI reasoning and problem-solving. This shift is crucial not because it simply adds a variety of data inputs but because it fundamentally enhances the AI's reasoning capabilities. The true value of multimodality lies not in the individual modalities themselves but in the enhanced reasoning ability that emerges when these modalities are combined [11]. A generalist model trained across action spaces, language, video, code, etc. gains its most interesting advantage not from the depth of knowledge in each modality but from the enriched reasoning capacity enabled by their integration. Embracing multimodality thus represents a critical evolution in AI agent development in that it demonstrates a shift away from language as the internal mechanism for reasoning.

We suggest that a further departure from autoregressive token generation–as is usually the technique employed by even generalist models–may be necessary to fully realize the reasoning potential of AI agents. The process of reasoning, we assert, is not adequately expressed by low bandwidth and linearly sequential language generation. And this idea is underscored by techniques that augment LLMs in ways that attempt to break them free of these constraints. Tree-of-thought attempts to break out of linear sequence generation. And the successful generalization across modalities of multimodel agents, as demonstrated by projects like GATO, demonstrate the criticality of moving beyond language itself.

IV. Conclusion

In this paper, we have explored recent trends in LLM-based agent research, highlighting key aspects such as memory, tool-use, and decision-making and reasoning. Despite advancements, we identify significant limitations, many of which arise from the reliance on language and autoregressive token generation for internal reasoning. In light of these limitations, we suggest that to enhance the reasoning capabilities of AI agents, a shift away from traditional language-based and linear sequential processing methods is necessary.

In conclusion, we propose a future research direction that transcends the conventional frameworks, advocating for AI agents to adopt more advanced, less sequential approaches to harness their full reasoning potential. This shift is crucial for overcoming the existing barriers and unlocking new possibilities in AI and LLM-based agent research.

References

[1] LLM+p: Empowering large language models with optimal planning proficiency.

[2] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. ChemCrow: Augmenting large-language models with chemistry tools.

[3] Norman Di Palo, Arunkumar Byravan, Leonard Hasenclever, Markus Wulfmeier, Nicolas Heess, and Martin Riedmiller. Towards a unified agent with foundation models. *arXiv preprint arXiv:2307.09668*, 2023.

[4] Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. A practical survey on faster and lighter transformers. *ACM Computing Surveys*, 55(14s):1–40, 2023.

[5] Jing Guo, Nan Li, Jianchuan Qi, Hang Yang, Ruiqiao Li, Yuzhen Feng, Si Zhang, and Ming Xu. Empowering working memory for large language model agents. *arXiv preprint arXiv:2312.17259*, 2023.

[6] Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*, 2022.

[7] Qi Liu, Dani Yogatama, and Phil Blunsom. Relational memory-augmented language models. *Transactions of the Association for Computational Linguistics*, 10:555–572, 2022.

[8] Lian Meng and Minlie Huang. Dialogue intent classification with long short-term memory networks. In *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6*, pages 42–50. Springer, 2018.

[9] Aaron Parisi, Yao Zhao, and Noah Fiedel. TALM: Tool augmented language models.

[10] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs.

[11] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[12] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools.

[13] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face.

[14] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning.

[15] Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents.

[16] Kristinn Thórisson and Helgi Helgasson. Cognitive architectures and autonomy: A comparative review. 3(2):1–30.

[17] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[18] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models.

[21] Lilian Weng. LLM powered autonomous agents. Section: posts.

[22] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023.

[23] Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*, 2021.

[24] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models.

[25] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models.

[26] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners.

[27] Wanjun Zhong, Lianghong Guo, Qiqi Gao, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

[28] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.