2024 3rd International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC) | 979-8-3315-3028-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIOTC63215.2024.10748304

A Survey of LLM-based Agents: Theories, Technologies, Applications and Suggestions

Xiaofei Dong
Institute of Artificial Intelligence
China Academy of Information and Communications Technology
Beijing, China
dongxiaofei@caict.ac.cn

Xueqiang Zhang
Institute of Artificial Intelligence
China Academy of Information and Communications Technology
Beijing, China
zhangxueqiang@caict.ac.cn

Weixin Bu*
Institute of Artificial Intelligence
China Academy of Information and Communications Technology
Beijing, China
* Corresponding author's email: buweixin@caict.ac.cn

Dan Zhang
Institute of Artificial Intelligence
China Academy of Information and Communications Technology
Beijing, China
zhangdan@caict.ac.cn

Feng Cao
Institute of Artificial Intelligence
China Academy of Information and Communications Technology
Beijing, China
caofeng@caict.ac.cn

***Abstract*—AI Agent has presented potential towards Artificial General Intelligence (AGI), which is expected to autonomously perceive the environments, make decisions and take actions. However, most of existing AI agents tend to train in confined environments with limited knowledge, yielding sub-optimal performance. Benefiting from the remarkable progress of large language models (LLMs), diverse LLM-based agents emerge. These agents employ LLM as the central brain to perceive, plan, and memorize, etc, which exhibit human-level intelligence across multifarious applications and obtain satisfactory performance. In this paper, we propose a survey of LLM-based agents from the perspective of theories, technologies, applications and suggestions, respectively. Specifically, we first deliver a recapitulative review of the theory foundation, which includes Large Language Models, Chain of Thought and AI Alignment, Retrieval-Augmented Generation, Embodied AI, etc; With this, we then present the key technologies, comprising four critical components: Perception, Planning, Memory and Action; Subsequently, we briefly explore some domain-related and evaluation applications; Finally, we provide pertinent suggestions based on the observations of significant challenges for LLM-based agents.**

Keywords-LLM-based Agent; Theory Foundation; Key Technologies; Applications; Suggestions.

I. Introduction

"*An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.*"

--Stuart Russell and Peter Norvig (2003)

As is well-known in Artificial Intelligence (AI) community, an AI agent refers to any AI entity which enables perceiving the environments, making decisions, and taking actions [1]. To design an autonomous and intelligent AI agent, previous studies [2] consider that AI agents could act based on the simple heuristic-driven policy functions, meanwhile, learn in constrained environments. However, in this context, there may exist some inherent limitations [3]: 1) the lack of autonomous learning capabilities in unconstrained environments, 2) difficulties in multi-agent decision-making, etc. To enhance AI agents' autonomous learning, meanwhile, enable more reasonable decision-making in multi-agent system, reinforcement learning-based agents attempt to maximize the cumulative rewards through trial and error, i.e., reward the expected actions and punish unexpected actions in an interactive environment. During this, the policy adjustment mechanism enables agents to generalize to unfamiliar environments. Furthermore, multi-agent reinforcement learning adopts CTDE, i.e., "Centralized Training and Decentralized Execute", which includes value-based and actor-critic schemes, to empower the multi-agent decision-making [4]. Nevertheless, reinforcement learning-based agents may encounter obstacles [3] such as substantial training time, low sampling efficiency, and unstable learning process, etc.

Due to the exceptional multimodal comprehension and generation abilities, unparalleled knowledge acquisition and reasoning capabilities, as well as the flexibility and scalability of LLMs, AI agents employ LLMs as the core brain, attempting to realize human-level perception, cognition and behaviors [5]. Compared with previous AI agents, LLM-based agents can demonstrate 1) *Autonomy* through potent multimodal perception, which expands the interaction space and perceptual fields; 2) *Spontaneity* through complex planning, which promotes more reasonable decision-making; 3) *Reactivity* through feasible embodiment or tool utilization, which enables more dynamic actions; 4) *Interactivity* through effective pluralistic memory, which enhances more interpretable multi-agent interactions. These have positioned LLM-based agents as pivotal catalysts for AGI [27], rendering them garnering significant research interests.

In this paper, we present a comprehensive survey of LLM-based agents. Different from some existing surveys [3][5], we attach importance to mining the underlying theory foundation that highly relevant to the key technologies. Backed up by this, we organize the survey with a progressive hierarchical structure: "Theories, Technologies, Applications, Suggestions", and then present the observations and summaries.

- First, we survey significant theory foundation such as LLM, CoT and AI Alignment, RAG, Tool Learning and Embodied AI in Section 2, which corresponds to the respective key technologies;
- Second, building upon the theory foundation, we provide a summary of several existing studies in Section 3 from the perspective of four key components: Perception, Planning, Memory and Action, respectively;
- Third, in Section 4, underpinned by theories and enabled by technologies, we conduct a review of diverse applications of LLM-based agents, including game, web, coding, evaluation, etc. Meanwhile, we present several suggestions based on significant challenges for the agents;
- Finally, we conclude the survey and provide some prospects in Section 5.

II. Theory Foundation

In this section, we present a brief overview of theory foundation for LLM-based agents, including LLM, CoT and AI alignment, RAG, etc. These largely underpin the corresponding implementations of key technologies.

A. Large Language Models

LLM-based agents are expected to perceive and interact with multimodal environments, such as text, visual, and audio, etc. LLMs can function as the perception powerhouse to better understand and generate multimodal objects. Specifically, LLMs comprise some significant components: Modality Encoder, Modality Connector, LLM Backbone, Output Projector, and Modality Generator [6]. We below introduce the overview skeleton from the modal perspective.

Modality Encoder. Given diverse modalities I_m, the Modality Encoder (Menc) aims to effectively extract the features F_m:

$$\mathrm{F}_m = \mathrm{Menc}(\mathrm{I}_\mathrm{m}). \quad (1)$$

Modality Connector. Given the text feature F_t and other modalities F_m, the Modality Connector aims to align or fuse them, and minimize the text generation loss $\mathcal{L}_{tg}$:

$$\mathrm{argmin}\mathcal{L}_{tg}\big(\mathrm{LLM}(\mathrm{F}_p, \mathrm{F}_t), \mathrm{F}_t\big), \quad (2)$$

where the aligned features as prompts are denoted as F_p.

LLM Backbone. It processes representations from diverse modalities, then produce text t and tokens S_m from other modalities:

$$t, \mathrm{S}_m = \mathrm{LLM}\big(\mathrm{F}_p, \mathrm{F}_t\big), \quad (3)$$

where S_m can be considered as the instructions for the generator.

Output Projector. It converts the signal tokens S_m into features F_g that Modality Generator can easily understand. Specifically, it aims to minimize the MSE loss between F_g and the conditional text representations C_t of Modality Generator, the goal of which is given as:

$$\mathrm{argmin}\mathcal{L}_{\mathrm{mse}}\big(\mathrm{F}_g, \mathrm{C}_t\big). \quad (4)$$

Modality Generator. It generally adopts Latent Diffusion Models (LDMs) [7] to generate diverse modalities. Given F_g, the noise ϵ, and noisy latent feature z_t, the U-Net in LDMs is denoted as U_θ, the LDM loss is thus presented as:

$$\mathcal{L}_{ldm} \coloneqq \mathbb{E}_{\epsilon\in\mathcal{N}(0,1),t}||\epsilon - \mathrm{U}_\theta\big(z_t, t, \mathrm{F}_g\big)||_2^2. \quad (5)$$

Besides, LLMs continuously make efforts to tackle hallucination, support infinitely long input, strut complex decison-making, etc, which effectively facilitate multiple modules of LLM-based agents.

B. Chain of Thought and AI Alignment

LLM-based agents ought to conduct high-level reasoning and planning for decision-making. Concretely, the agents tend to carry out task decomposition, single-path and multi-path reasoning, planning reflection, etc. Given the task goal g, environment e, the prompt set $\mathcal{P}$ and overall parameters Θ of LLMs, task decomposition can be formalized as [8]:

$$g_{0,1,\dots,n} = \mathrm{decp}(g, e, \mathcal{P}, \Theta), \quad (6)$$

where "decp" denotes the decomposition operation, and $g_{0,1,\dots,n}$ denotes the sub-goals. Upon this, LLM-based agents ought to support single-path and multi-path reasoning, which can be empowered by CoT and its variants [9].

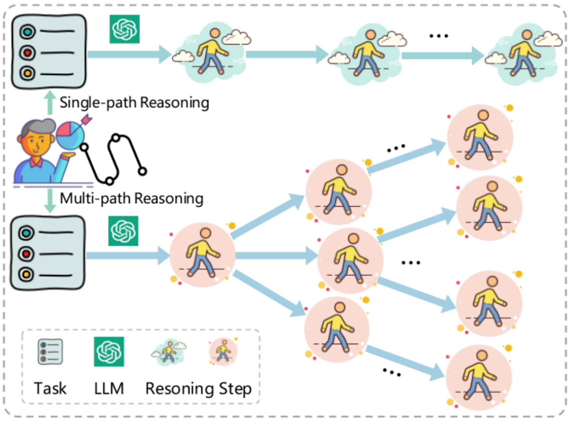

Figure 1. A simple schematic of single-path and multi-path reasoning. LLMs function as the brain of the agents, enabling more logical reasoning.

For single-path reasoning, each step follows a single path to the next. CoT advocates incorporating detailed reasoning steps into the query when tackling intricate tasks. These procedural steps function as illustrative examples, encouraging LLMs to devise and execute solutions in a methodical, step-wise fashion. For multi-path reasoning, each step can lead to multiple follow-up steps. the reasoning process leading to the generation of the ultimate plans is structured in a hierarchical, tree-like organization. Tree of Thought (ToT) and Graph of Thought (GoT) [10], etc, follow this way, expanding the reasoning space and planning schemes. In this context, the multi-path selection problem emerges, i.e., given different reasoning paths $p_{0,1,\ldots,n}$, the optimal path selection process can be formalized as [8]:

$$\hat{p} = \mathrm{select}\left(g, e, p_{0,1,\ldots,n}, \Theta, \mathcal{F}\right), \tag{7}$$

where $\mathcal{F}$ denotes the set of path search function. We provide a simple schematic for single-path reasoning and multi-path reasoning in Figure 1.

On the basis, it's imperative to reflect on the planning process from diverse perspectives, one of which is the alignment with human intentions and values. AI alignment [11], especially LLM alignment, focuses on the RICE principle: Robustness, Interpretability, Controllability and Ethicality. Furthermore, it necessitates that the planning process must fulfill both outer and inner alignment. Outer alignment requires the planning process to translate human intentions or expected goals into the training objectives of LLM-based agents, which generally adopts RLHF, comprising supervised fine-tuning, reward modeling and policy optimization. Analogously, inner alignment requires the planning process to ensure the inner optimization objectives be consistent with the training objectives of LLM-based agents. In detail, inner alignment emphasizes alignment assurance of planning in the light of safety evaluations, interpretability, and human values verification. With the outer-inner alignment, LLM-based agents are enabled to engage in trustworthy reasoning and planning.

C. *Retrieval-Augmented Generation*

LLM-based agents are supposed to memorize and retrieve high-quality external knowledge, and then generate the most relevant response. To this end, RAG incorporates external knowledge to enrich user queries or generate answers. It first invokes the retrieval module, which retrieves the most relevant chunk-based content, then measures the similarity between queries and the chunks in vector databases. Then, the retrieved content are combined with the user queries, to bolster generating more informative and matching answers. Finally, RAG combines the intermediate content and the generated outputs, to augment the answers. With the scheme, RAG has demonstrated impressive strength for knowledge retrieval and memory [3].

However, most of existing RAGs fail to handle query-focused summarization tasks, i.e., answer global questions over an entire corpus. GraphRAG adopts a graph-based community-aware scheme in the overall consideration. It first indexes chunks into graph elements: nodes (entities) and edges (relationships), and builds a chunk-based graph. Then, it adopts community detection to partition the entire graph into community groups. Finally, it requires each community group to generate a summary w.r.t. the given query, and the final answer is obtained by summarizing all query-focused relevant community responses. With the advancement of RAGs, LLM-based agents can possess high-level memory capabilities [5].

D. *Tool Learning and Embodied Intelligence*

LLM-based agents should effectively utilize tools, such as API calling, code interpreters, and even the embodied ones. These expand the action space and capabilities of the agents. Additionally, it's critical for the agents to comprehend both individual tool functionalities and the dependencies between the tools, enabling more intelligent actions.

Furthermore, in the pursuit of AGI, the embodied intelligence empowers the agents to proactively integrate with the environments based on the physical entities. Specifically, the embodied agents should carry out some fundamental actions, such as observations, manipulations, and navigations. First, the embodied agents are expected to acquire multimodal environmental information by real-time observation. Then, the embodied agents should conduct diverse manipulation tasks such as motion simulation, multi-step reasoning and planning manipulation, etc. Moreover, the embodied agents ought to make dynamic navigation after precise observations and long-term manipulations, and then autonomously explore the environments. With the high-quality tool learning and embodied intelligence, LLM-based agents can take human-like actions, and dynamically interacts with diverse environments.

III. KEY TECHNOLOGIES

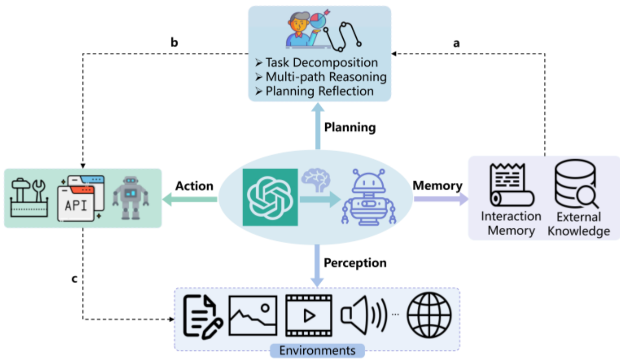

Figure 2. Overview skeleton of key technologies comprising four modules. LLMs function as the brain of the agents, empowering these modules. a) The memory module can facilitate planning with entensive knowledge and experience; b) The planning module develops reasonable plans, and then hand them over to the action module; c) The action module can utilize tools ,apis and even embodied actions to interact with environments and receive feedbacks.

In this section, we below present a concise summary of key technologies for LLM-based agents, which is composed of four significant modules: Perception, Planning, Memory and Action. The overview skeleton of them is provided in Figure 2.

A. *Perception*

The perception module is responsible for interacting with multiple environments and receiving feedbacks. LLM-based agents draw support from LLMs to perceive the textual, visual, auditory and other objects from the environments.

Auto-GPT and XAgent[3], etc, can interact with humans by textual content, meanwhile, understand the implicit intentions, and generate the relevant answers. FLAN [5] further enables understanding text instructions in a zero-shot manner, which enhances the generalization capability in textual perception tasks. Analogously, for visual perception, ViT and its variants adopt the transformer-based architecture for visual understanding. Meanwhile, it's vital to convert the output visual representations into the ones that LLMs can comprehend. To this end, BLIP-2 [12] adopts Q-Former as the modality connector, which utilizes a learnable set of query vectors to extract linguistically salient visual representations. MiniGPT-4 [13] further proposes a more efficient projection module to enable the alignment between the visual encoder and the frozen LLMs. Since videos can be considered as continuous image frames, LLM-based agents can perceive them with existing visual perception schemes. Upon this, Flamingo and its variants further adopt an effective mask mechanism to capture the temporal attributes of videos. AST utilizes a ViT-like scheme to understand audio spectrogram images. Upon this, AudioGPT leverages various audio LLMs to conduct modality transformation, task analysis, model assignment, etc, better understanding and generating audios.

LLM-based agents are expected to perceive multimodal environments, then understand and generate diverse modalities. It's promising to explore richer environmental perceptions, and empower subsequent technical modules.

B. *Planning*

The planning module strives to empower the LLM-based agents to decompose tasks, make complex reasoning and conduct plan reflection. It ensures the agents behave in a more reasonable and reliable manner.

Typically, the agents perform task decomposition in two different branches: decomposition-first and interleaved [8]. The former first decomposes the tasks into sub-goals and then plan for them, while the latter directly conducts interleaved task decomposition and sub-goal planning, where each step reveals limited sub-tasks. For the decomposition-first branch, HuggingGPT [14] explicitly directs LLMs to decompose the given task into the sub-ones utilizing defined interdependencies. PSP [15] further enables zero-shot planning, which refines the prompts in a two-step manner than Zero-Shot-CoT [16], and enhances the reasoning capability in mathematic, common sense, etc. For the interleaved branch, Zero-Shot-CoT enables zero-shot planning with only one trigger prompt: "*think step by step*". Different from CoT-based methods, ReAct [17] alternates between the decoupled reasoning and planning, other than integrating reasoning into the planning process. PAL utilizes the LLMs to parse task descriptions and generate codes for transitional reasoning, while delegating the execution to a Python interpreter, enabling better arithmetic and symbolic reasoning. Following the task decomposition and reasoning, LLM-based agents ought to invoke the planning reflection to handle hallucination and "thought loops" issues. Specifically, Reflexion [18] contemplates the task feedbacks with an evaluator and the self-reflection scheme, retaining the reflective narrative in episodic memory, enhancing subsequent decision-making. CRITIC further adopts external tools to validate the actions, and then carry out self-reflection and self-correction with external knowledge. LEMA learns from mistakes, which identifies erroneous reasoning sequences, elucidate the underlying causes, and refine them.

It's crucial for the LLM-based agents to decompose tasks, devise appropriate plans with reasoning and plan reflection. With these, the agents can dynamically adjust the plans, ensuring the behaviors align with human intentions, values and the environments.

C. *Memory*

The memory module is dedicated to memorizing the observations, thoughts, actions, feedbacks and external knowledge of LLM-based agents, which can facilitate consistent and effective actions. Specifically, the memory module comprises two main sub-modules: sources and operations [3].

For memory sources, it can be categoried into inside-interaction, cross-interaction and external knowledge. The inside-interaction memory means the historical information inside a interaction, which is considered as the most relevant. Generative Agents leverage LLMs to mimic human daily routines, and the memory stores direct historical behaviors of different agents. Voyager [19] attempts to explore Minecraft game agents, whose memory comprises essential executable codes and routines for task completion. Analogously, the cross-interaction memory means the long-term historical information accumulated across various interactions, which contains more generalized information. Reflexion [18] extracts verbalized information across diverse interactions, and implements them in subsequent steps. Retroformer further enhances the reflection module, enabling more effective and adaptive cross-interaction memory. Besides, LLM-based agents should incorporate with external knowledge to facilitate actions. ReAct [17] invokes APIs to obtain external knowledge during reasoning and interactions. CodeAgent employs various programming tools and effective search strategy to obtain external knowledge for repo-level code generation. Various agents adopt RAG to retrieve high-quality external knowledge, which can be organized in different formats, such as textual, parametric and vectorized, etc.

For memory operations, the common procedure includes memory writing, memory management and memory retrieval. TiM and ChatDB [3] capture the entity relations and write them into a structured database, and enable memory operations through SQL instructions. Besides, different agents write memory into vector databases for high-quality retrieval. Due to various writing mechanisms, it's vital to effectively manage the stored memory. MemoryBank [5] summarizes the long-term interactions, then dynamically refines and updates the memory. Likewise, Voyager [19] refines the memory from interaction feedbacks. These management operations enhance the capabilities of LLM-based agents to produce high-level memory, remove duplicated memory and drop irrelevant memory. Afterwards, the memory retrieval can extract the most relevant, recent and significant memory for planning, actions, etc. We below present the goal of memory retrieval [3]:

$$\mathrm{m}^{\star} = \arg\min_{\mathrm{m}\in\mathrm{M}} \alpha\, r^{rel}(q, \mathrm{m}) + \beta r^{rec}(q, \mathrm{m}) + \gamma r^{sig}(\mathrm{m}), \quad (8)$$

where q denotes the query, $m \in M$ denotes the memory and the memory set. Besides, $r^{rel}(\cdot,\cdot)$, $r^{rec}(\cdot,\cdot)$ and $r^{sig}(\cdot)$ denote the relevant, recent and significant score functions, respectively; α, β, and γ are the corresponding balance factors. Following this, some LLM-based agents such as ChatDB, ExpeL [3] retrieve the memory with SQL commands or vector matching from different memory sources.

It's indispensable for LLM-based agents to enable effective memory mechanisms. With this, the agents could better learn from experience, self-evolve and take human-like actions.

D. Action

The action module is mainly used to take goal-driven adaptive actions and interact with the environments utilizing creative tools. We below provide a brief summary of tool learning and embodied actions for LLM-based agents.

Generally, the agents can employ various tools to enhance and expand the action capabilities, they should understand tools, use tools and integrate tools. Before utilizing tools, the agents can adopt zero-shot or few-shot prompting to acquire knowledge of the function, parameters, and usage instructions about the tools. With this, the agents can enhance the tool utilization. HuggingGPT [14] empowers the agents to accomplish complex tasks leveraging external APIs. API-Bank further enables dynamic API recommendation and generation. ToolFormer supports the transformation of a given tool into another one with explicit instructions. Differently, MemoryBank further integrates external model-level tools for encoding and matching. CRAFT proposes to develop general-purpose tools for tailored actions [3].

The embodied actions must fully understand the complex tasks and environments, then conduct human-level perceptions, comprehension, planning and interactions through high-level tools. Google proposes the embodied LLMs of RT series, including RT-1, RT-2 and RT-H. RT-1 mainly focuses on enhancing the robots' actions by combining the multimodal information and LLMs to generate control instructions. RT-2 further develops the capabilities of cross-modal learning and reasoning, which allows the robots to reason about specific actions from more abstract instructions, realizing end-to-end embodied control. RT-H combines the flexible hierarchical policies and LLMs, and then predict high-level language motions as the conditions to conduct embodied actions. Besides, OpenVLA builds an vision-language-action model based on Llama2 and multimodal models. It first captures visual features at diverse granularities, and then explore efficient fine-tuning strategies for embodied actions. Differently, RoboMamba adopts Mamba model to provide both reasoning and action capabilities, meanwhile, maintaining efficient fine-tuning and inference for embodied actions [3][5].

LLM-based agents tend to employ diverse tools to empower the actions. Further, the embodied agents generally attempt to build Vision-Language-Action models with different strategies.

IV. Applications and Suggestions

We present a brief review of applications in multiple domains for LLM-based agents in this section. Considering the theory foundation, key technologies, domain constraints and potential challenges, we propose some pertinent suggestions.

A. Applications

LLM-based agents have demonstrated great potential in diverse domains due to the powerful understanding, reasoning and action capabilities. We below introduce some agents in domains such as natural science, social science, engineering, etc. Besides, we also summary some evaluation applications. Some representative application directions are given in Table 1, and the scope is gradually expanding.

TABLE I. Some representative application directions of LLM-based agents [3][5].

Directions	Sub-Directions	Representative Work	
Natural Science	Math	Math Agents[3]	ToRA[20]
		COPRA[3]	…
	Chemistry	ChemCrow[21]	ChatMOF[5]
		Coscientist[5]	…
	Biology	BSDG[22]	BioPlanner[3]
	…	…	
Social Science	Economics	Alpha-GPT[3]	Homo-Silicus[5]
		…	
	Law	LJP-Agent[23]	ChatLaw[5]
		Blind Judgement[3]	…
	Psychology	SMHTE[5]	Replika-MWS[5]
	…	…	
Engineer	Computer and Games	GPT-Engineer[5]	AutoGen[5]
		WebAgent[3]	GITM[5]
		Voyager[19]	…
	Industrial engineer	TaPA[3]	LLM-Planner[5]
	…	…	
Evaluation	Tool utilization	ToolLLM[3]	…
	Env interaction	AgentBench[24]	SafetyBench[25]
…	Chinese alignment	AlignBench[26]	…

For math, Math Agents focus on building a mathematical computing platform, then deal with mathematical embedding and genomics problems; ToRA [20] employs high-level reasoning and external tools for complex mathematical problems. Besides, some agents are dedicated to mathematical theorem derivation and proving, etc. For chemistry, ChemCrow [21] attempts to automatically address biosynthesis, drug discovery, and materials design issues. For biology, BSDG [22] proposes an active learning method with GFlowNet to generate biological sequences. Likewise, for economics, Alpha-GPT [3] adopts the heuristic method to understand quantitative concepts and conduct Alpha mining. For laws, LJP-Agent [23] supports the case-based learning and answering questions within the legal domain. In engineering domain, GPT-Engineer and AutoGen [5], etc, allow the code generation. For game industry, GITM and Voyager [19] support various role-playing games. On top of that, various agents burst, and influence multiple domains.

There exists diverse evaluation benchmarks for LLM-based agents. We present some representative work. ToolLLM is employed for evaluating tool utilization in an instruction-tuning manner. AgentBench [24] enables the evaluation of agents in 8 distinct environments such as code, game, web, etc. SafetyBench [25] attempts to evaluate the safety of agents with diverse multiple- choice questions. AlignBench [26] proposes to evaluate the Chinese alignment with a human-in-the-loop pipeline on different tasks. With the diversified evaluations of LLM-based agents, the agent community will develop towards goodness and improvement.

B. Suggestions

We below present some suggestions considering the intricate challenges from a macro perspective for LLM-based agents.

- **LLM-based agents are expected to make further breakthroughs in intrinsic constraints**. Although LLMs have demonstrated powerful capabilities for the agents, the intrinsic constraints pose fundamental challenges. The agents still have trouble with hallucination; Besides, since the agents have to effectively interact with multiple complex environments, infinite long input and memory, high-level multimodal perception, intricate reasoning and planning for more intelligent actions are required;
- **LLM-based agents ought to scale up**. Most of existing multi-agent systems run in a small-scale way. How to dynamically adjust the scale, reasonably schedule and allocate resources, and enable high-level multi-agent collaborations, remains in-depth exploration;
- **LLM-based agents preferably follow controllable AI alignment**, including human laws, intensions, and values, etc, developing towards goodness. In order to better serve humans, the agents should run in the controllable manner. That is, the agents have to abide by the laws and rules, align with correct human intensions and values, act under appropriate permissions;
- **LLM-based agents need more comprehensive and holistic evaluations.** Diverse existing evaluation benchmarks evaluate the agents from different separate perspectives, such as tool utilization, reasoning capability, or action performance, etc. Nevertheless, it remains a long way to build a standard and unified evaluation platform, which requires joint efforts.

V. CONCLUSIONS

This paper presents a systematical review of LLM-based agents, where we focus on mining the theory foundation that highly relevant to the key technologies. Furthermore, we provide a concise summary of the significant modules, including Perception, Planning, Memory and Action, inside which we introduce some significant work. Based on the theory foundation and key technologies, we review some applications in diverse domains, also, some representative evaluation benchmarks. Considering the observations of some intricate challenges to be addressed, we propose some sincere suggestions. As the saying puts it: “To some extent, embarking on the journey forward is more vital than arriving at the destination.”, although the research progress is far from realizing AGI yet, the LLM-based agents do take a solid step forward. We hope our efforts can facilitate furture research.

REFERENCES

[1] Russell, S.J. and Norvig, P., 2016. *Artificial intelligence: a modern approach*. Pearson.

[2] Nilsson, N.J., 1992. *Toward agent programs with circuit semantics* (No. STAN-CS-92-1412).

[3] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., et al, 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, *18*(6), p.186345.

[4] Canese, L., Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M. and Spanò, S., 2021. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, *11*(11), p.4948.

[5] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., et al, 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

[6] Zhang, D., Yu, Y., Li, C., Dong, J., Su, D., Chu, C. and Yu, D., 2024. MM-LLMs: Recent Advances in MultiModal Large Language Models. *Findings of the Association for Computational Linguistics ACL 2024*, pages 12401–12430.

[7] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

[8] Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., et al, 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716*.

[9] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.

[10] Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., et al, 2024, March. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 16, pp. 17682-17690).

[11] Ji, J., Qiu, T., Chen, B., Zhang, B., Lou, H., Wang, K., et al, 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

[12] Li, J., Li, D., Savarese, S. and Hoi, S., 2023, July. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning* (pp. 19730-19742). PMLR.

[13] Zhu, D., Chen, J., Shen, X., Li, X. and Elhoseiny, M., 2024, MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *The Twelfth International Conference on Learning Representations.*

[14] Shen, Y., Song, K., Tan, X., Li, D., Lu, W. and Zhuang, Y., 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, *36.*

[15] Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W. and Lim, E.P., 2023, July. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2609-2634).

[16] Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y., 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, pp.22199-22213.

[17] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R. and Cao, Y., 2023, ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

[18] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. and Yao, S., 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, *36*.

[19] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L. and Anandkumar, A., Voyager: An Open-Ended Embodied Agent with Large Language Models. In *Intrinsically-Motivated and Open-Ended Learning Workshop@ NeurIPS2023.*

[20] Gou, Z., Shao, Z., Gong, Y., Yang, Y., Huang, M., Duan, N. and Chen, W., 2024. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *The Twelfth International Conference on Learning Representations.*

[21] Bran, A.M., Cox, S., Schilter, O., Baldassari, C., White, A. and Schwaller, P., ChemCrow: Augmenting large language models with chemistry tools. In *NeurIPS 2023 AI for Science Workshop*.

[22] Jain, M., Bengio, E., Hernandez-Garcia, A., Rector-Brooks, J., Dossou, B.F., Ekbote, C.A., Fu, J., Zhang, T., Kilgour, M., Zhang, D. and Simine, L., 2022, June. Biological sequence design with gflownets. In *International Conference on Machine Learning* (pp. 9786-9801). PMLR.

[23] Shui, R., Cao, Y., Wang, X. and Chua, T.S., 2023, December. A Comprehensive Evaluation of Large Language Models on Legal Judgment Prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 7337-7348).

[24] Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., et al, 2024, AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations.*

[25] Zhang, Z., Lei, L., Wu, L., Sun, R., Huang, Y., Long, C., et al, 2024, August. SafetyBench: Evaluating the Safety of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 15537-15553).

[26] Liu, X., Lei, X., Wang, S., Huang, Y., Feng, Z., et al, 2024. AlignBench: Benchmarking Chinese Alignment of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 11621–11640.

[27] Jiang, Y., Li, X., Luo, H., Yin, S. and Kaynak, O., 2022. Quo vadis artificial intelligence?. Discover Artificial Intelligence, 2(1), p.4.