From RAG to ARG: Agent Reinforced Generation for Agentic Intelligence

Jing Yang, Yonglin Tian, Fei Lin, and Fei-Yue Wang

***Abstract*—Advancements in large language models (LLMs) have markedly improved the adaptability of artificial intelligence (AI) agents in dynamic and open environments. However, with the growing number and diversity of agents, ensuring secure, reliable, and autonomous collaboration among them has become an urgent and critical challenge. To this end, this letter proposes agent reinforced generation (ARG) to establish a multi-agent system with audit trail functionality, privacy compliance, and autonomous coordination. ARG integrates the model context protocol (MCP) and agent-to-agent (A2A) protocol to define the rules and logic governing agent-to-agent communications as well as agent-to-tool/data engagements. Decentralized autonomous organizations and operations (DAOs) are employed to enable agents to coordinate and execute tasks in a transparent and tamper-resistant manner. Additionally, the operational process of ARG is elaborated from task issuance to completion to validate the auditability and immutability of task coordination and execution. Finally, we highlight five key features of ARG, including parallelism and throughput, scalability across domains and load, fault tolerance and graceful failure, resource efficiency through delegation, as well as data security and privacy protection, positioning it as a promising paradigm for the realization of agentic intelligence.**

I. Introduction

Large language models (LLMs), such as GPT-4, Claude, and DeepSeek, have revolutionized generative tasks with their emergent reasoning and in-context learning capabilities [1, 2]. However, their reliance on static pretrained knowledge leads to three fundamental limitations: (1) temporal grounding: LLMs have an inherent inability to access post-training information, which restricts their effectiveness in dealing with current or rapidly changing knowledge; (2) factual hallucination: LLMs may confidently generate false statements, which undermines their reliability in critical applications; and (3) passive interaction: LLMs lack the ability to autonomously use tools and adapt to changing environments, which limits their effectiveness in dynamic and interactive tasks [3]. These constraints hinder their feasible deployment in real-world, ever-changing scenarios.

Retrieval-augmented generation (RAG) shows promise in addressing the limitations of LLMs by integrating external knowledge retrieval into the generation pipeline [4]. While effective for fact-intensive tasks and open-domain question&answer (Q&A), standard RAG suffers from rigid retrieval policies that employ fixed query rewriting strategies, unidirectional data flow without post-generation feedback, and inability to orchestrate multi-tool workflows [5]. This results in suboptimal performance when handling complex queries that require iterative retrieval-reasoning loops. Although search-augmented generation and extension (SAGE) [6] enhances RAG by extending search coverage and applying multi-tier caching for efficient content retrieval, it still falls short in overcoming the limitations above.

Recent advances in artificial intelligence (AI) agents demonstrate that equipping LLMs with goal-directed autonomy and tool-use capabilities enables dynamic task solving [7]. In contrast to passive RAG systems, agents can proactively adjust strategies based on environmental feedback, such as improving application programming interface (API) calls and validating intermediate results [8]. As tasks become increasingly complex, this naturally leads to the evolution from single-agent systems to multi-agent systems. While they offer greater computational power, these systems also introduce new challenges in ensuring secure and reliable interactions both among agents and between agents and tools. To overcome these challenges, it is essential to establish a novel collaboration paradigm characterized by audit trail, privacy compliance, and autonomous coordination, a critical prerequisite for scaling multi-agent applications.

To this end, this letter proposes agent-reinforced generation (ARG) as a novel paradigm for agentic intelligence. ARG integrates model context protocol (MCP) [9], agent-to-agent (A2A) protocol [10], and decentralized autonomous organizations and operations (DAOs) [11] to enable the construction of secure, efficient, and reliable multi-agent systems. MCP aims at seamless integration between LLMs and external functions, tools, and data sources. A2A defines structured communication rules that enable agents to coordinate, negotiate, and collaborate effectively. DAO refers to a distributed organizational structure built on blockchain technology, where rules, decision-making processes, and operations are encoded in smart contracts. It allows agents to

Manuscript received: 6 January 2025; revised: 31 January 2025; accepted: 10 February 2025. *(Corresponding authors: Fei Lin and Fei-Yue Wang.)*

Citation: J. Yang, Y. Tian, F. Lin, and F.-Y. Wang, From RAG to ARG: Agent reinforced generation for agentic intelligence, *Int. J. Intell. Control Syst.*, 2025, 30(1), 76–82.

Jing Yang and Yonglin Tian are with State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yangjing2020@ia.ac.cn; yonglin.tian@ia.ac.cn).

Fei Lin is with Faculty of Innovation Engineering, Macau University of Science and Technology, Macao 999078, China (e-mail: feilin@ieee.org).

Fei-Yue Wang is with Macau Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China, and also with The DeSci Center of Parallel Intelligence, Obuda University, Budapest H-1034, Hungary (e-mail: feiyue@ieee.org).

Digital Object Identifier 10.62678/IJICS202503.10177

coordinate and perform tasks in a decentralized, transparent, and tamper-proof manner, guaranteeing accountability, traceability, and autonomous governance. By combining these three technologies, ARG empowers multi-agent systems to operate with shared context, dynamic cooperation, and trustworthy execution. This integration ensures reliable agent communication and rule-based operations, enabling agentic intelligence to arise from collective multi-agent interactions.

II. LLM AND AI AGENT

The emergent capabilities demonstrated by LLMs are driving profound transformations across a wide array of domains. These capabilities arise not only from the support of large-scale data training but also from the highly scalable and general-purpose Transformer architecture [12]. Researchers have extensively explored the adaptability of three fundamental Transformer-based architectures: encoder-only, decoder-only, and encoder-decoder. Encoder-only models, such as BERT [13] and RoBERTa [14], are well-suited for tasks like text classification and sentiment analysis; however, their lack of autoregressive generation capabilities poses limitations in open-ended generative settings. Encoder-decoder architectures, such as T5 [15] and BART [16], integrate understanding and generation through a dual-tower structure, making them suitable for tasks like translation and summarization. Nevertheless, they face structural constraints in terms of inference efficiency and contextual coherence. In contrast, decoder-only models follow the autoregressive language modeling paradigm, naturally aligning with open-ended generation and multitasking learning. This architecture has become the backbone of general-purpose LLMs, giving rise to a range of high-performance models, including GPT-3 [1], LLaMA [17], Vicuna [18], and DeepSeek [2], which exhibit remarkable abilities in in-context learning, instruction following, and semantic reasoning.

Meanwhile, the modality-agnostic nature of the Transformer architecture has facilitated structural transfer across domains. Beginning with the Vision Transformer (ViT) [19], a unified encoding mechanism was introduced into visual models. Radford et al. [20] proposed a contrastive learning framework based on image-text alignment, named CLIP, laying the foundation for zero-shot multimodal generalization. Building on this trajectory, researchers further aligned language models with visual perception, leveraging large-scale multimodal training to develop vision-language models (VLMs) capable of cross-modal understanding and generation. Representative models include GPT-4V [21], LLaVA [22], and the BLIP/BLIP-2 [23] series. Beyond representation learning, the explicit orchestration of reasoning has emerged as a frontier in LLM design.

With the continuous advancement of LLMs, AI agents are gradually evolving from early rule-based systems into complex, intelligent agents empowered by LLMs for cognition, planning, and reasoning [24]. As illustrated in Fig. 1, the entire process is initiated by user queries or external environmental signals, which are combined with predefined prompt templates to construct the input for the model, thereby triggering the task execution of the agent. The agent leverages its embedded LLM module to perform task planning and multi-turn reasoning while invoking external tools to carry out specific operations and perceive environmental feedback. Simultaneously, the system utilizes vector retrieval mechanisms to access historical memory or dynamically writes key information from the current interaction into the memory module, enabling context preservation and continual learning. Finally, the agent generates a response based on its reasoning and execution results and returns it to the user [25].

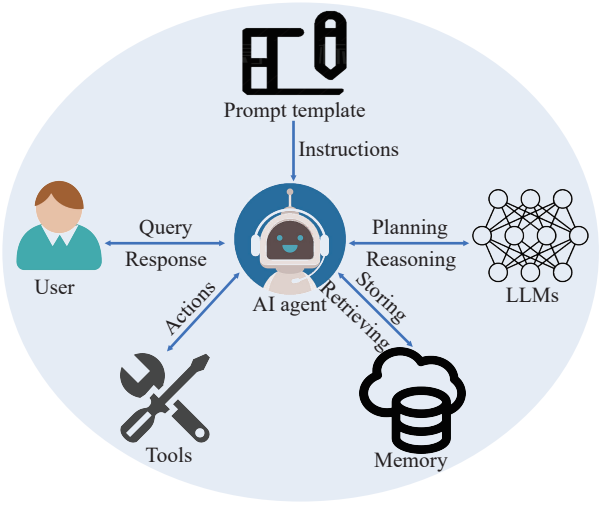

Figure 1 LLM-based AI agents.

As AI agent architectures evolve from basic perception to complex cognition, enhancing their capabilities in multi-step reasoning and structured planning has become a key focus of current research. Rawat et al. [26] proposed a multi-step planning mechanism to improve the action capabilities of the agent. Sumers et al. [27] introduced the CoALA cognitive framework, enabling language models to acquire stronger decision organization and memory functions. Building on this, the CoPlanner framework incorporates collaborative planning strategies to support task decomposition and cooperative reasoning among multiple agents [28]. Meanwhile, the Chameleon system integrates search engines, code execution, and multimodal input tools to create a highly composable and executable general-purpose agent framework [29]. These studies not only enrich the functional modules and reasoning mechanisms of intelligent agents but also lay a solid foundation for constructing systems characterized by agentic intelligence, providing critical support for the advancement of ARG systems.

III. PROMPT ENGINEERING AND RAG

Prompt engineering is a technique for optimizing LLM outputs. Its core idea is to guide the model to generate desired outputs by designing specific input prompts without modifying the internal parameters of the model. Early methods relied on manually crafted templates that improved output quality by providing explicit instructions or contextual examples [30].

With the advancement of technology, researchers have proposed new prompting strategies such as few-shot

prompting [1] to overcome the limitations of manual template design. By including multiple task examples in the prompt, models are better guided to understand task patterns and generate target outputs more accurately. Further, the prompting paradigm has evolved to chain of thought (CoT) prompting [31], enabling models to produce structured intermediate reasoning steps, thereby achieving breakthroughs in logic, mathematics, and strategic tasks. In parallel, hybrid prompting strategies such as prompt chaining [32] and reasoning and acting (ReAct) [33] have emerged, facilitating procedural control over multi-step tasks.

At the same time, lightweight learning approaches such as AutoPrompt [34] and prefix tuning [35] have shifted prompt engineering toward learnable and modular designs. However, in practical applications, especially those involving external domain knowledge or long-tail factual queries, the intrinsic understanding of LLMs often falls short of delivering high-accuracy results. Against this backdrop, prompt engineering has increasingly merged with external knowledge injection, giving rise to a new RAG paradigm. RAG inherits the guiding principle of prompts at the architectural level but augments it with dynamic knowledge retrieval by integrating retrievers and knowledge bases. This backend enhancement provides "factual support" for prompts. As shown in Fig. 2, the basic structure consists of three components: retrievers, generators, and knowledge bases. The retriever encodes the user query and retrieves the most relevant text passages from an external knowledge base; these passages, along with the original input, are then passed to the LLM-based generator to produce the final response based on the enriched context. The knowledge base is responsible for storing and organizing information sources [4]. Obviously, RAG addresses the limitations of static knowledge in LLMs by enabling access to broader and more up-to-date information resources, significantly improving factual consistency and timeliness.

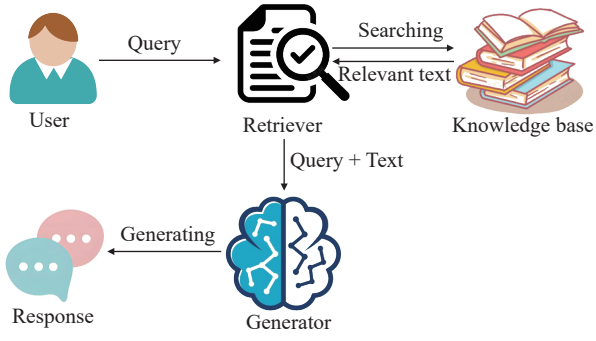

Figure 2 Retrieval augmented generation.

As the technology matures, RAG architectures have diversified in their design. In the retriever component, Mala et al. explored hybrid retrieval strategies that combine sparse retrieval methods with dense retrieval to balance high recall and semantic relevance [36]. In the generator component, beyond the traditional encoder-decoder architectures, Izacard et al. proposed using decoder-only structures with prompt reconstruction to improve generation efficiency [37]. Moreover, RAG systems are evolving toward more advanced capabilities, such as multi-turn retrieval, cross-document aggregation (Multi-hop RAG) [38], and knowledge graph enhancement (Graph-RAG) [39], which enable the support of complex reasoning chains and long-context dependencies. SAGE [40] is a representative advanced variant that enhances RAG by enabling dynamic extraction of relevant information from external data sources. It introduces an intelligent routing unit to adaptively switch between search and retrieval processes, and incorporates a hierarchical management system of internal and external caches to efficiently handle the exchange between external data and the limited input contents of LLMs. Despite its advantages in mitigating static knowledge limitations, RAG still faces challenges in task control, state tracking, and proactive reasoning [41]. These limitations have motivated emerging research into integrating agent-based mechanisms, driving the evolution of the architecture toward the ARG paradigm.

IV. Agent Reinforced Generation

ARG is a novel LLM-based multi-agent system architecture that integrates MCP, A2A, and DAO, aiming to build a trustworthy, efficient, and cross-domain collaborative agent ecosystem in open environments. As illustrated in Fig. 3, centered on agents, ARG involves inter-agent communication and collaboration, agent access to underlying tools and data, decentralized governance among all agents, and integration with external data sources, covering the full spectrum of perception, decision-making, collaboration, and execution.

A. Inter-Agent Collaboration

In ARG, there are two types of LLM-based agents: client agents and remote agents, whose roles are dynamically interchangeable. Client agents act as both task initiators and central coordinators, responsible for interpreting user intent, planning tasks, and managing overall execution. When a task exceeds the local capabilities of the client agent or involves heterogeneous resources, it invokes the A2A protocol to establish communication with multiple functionally diverse remote agents. Various tasks are then delegated to these remote agents according to their respective capabilities to ensure efficient and effective task completion. Through the

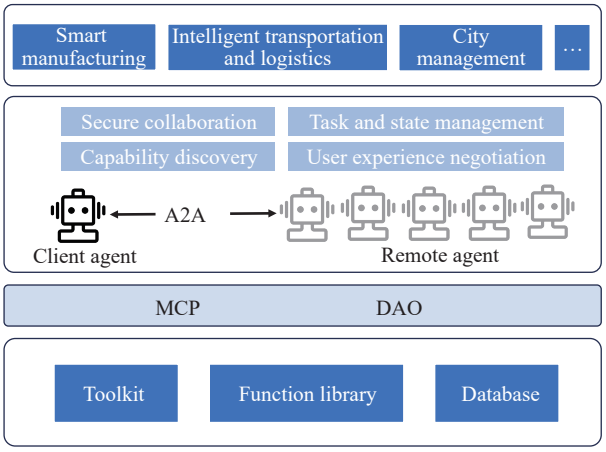

Figure 3 Basic architecture of ARG.

A2A protocol, agents can perform four core functions: capability discovery, task and state management, secure collaboration, and user experience negotiation. Specifically, capability discovery enables agents to identify and select optimal collaborators through capability broadcasting, task and state management coordinates complete task lifecycle execution, secure collaboration achieves protected information exchange without exposing internal logic, and user experience negotiation dynamically adapts interaction formats to interface capabilities. This ensures that inter-agent interactions are interpretable, consistent, and resilient, while delivering a more adaptive and satisfying user experience.

B. Knowledge and Tool Invocation

During task execution, LLM-based agents must access external knowledge and tools to support computation and decision-making. ARG employs MCP to enable context sharing between agents and the external environment, thereby ensuring the reliability and accuracy of LLM-generated content. MCP allows agents to extract dynamic contextual information such as historical task trajectories, user preferences, and tool feedback. This information is injected into the current input context of the LLM to enhance the coherence and depth of reasoning, helping agents better comprehend task contexts and user requirements. Additionally, MCP supports integration with external databases and vector knowledge bases, enabling agents to retrieve real-time data, industry regulations, and domain-specific documents. This capability not only improves the accuracy of responses but also significantly enhances the trustworthiness and professionalism of the agent. When agents need to invoke tools or execute functions, MCP bridges the model outputs with tool execution and feeds the intermediate states returned by the tools back into the context for subsequent reasoning. Through these mechanisms, MCP endows agents with the capabilities of "memory retention", "knowledge integration", and "tool invocation", advancing LLMs from pure language generators to intelligent agents capable of task execution.

C. Decentralized Operation

The DAO, built on blockchain and smart contracts, is employed in ARG to ensure trustworthy task execution, regulatory compliance, and auditable behavior within multi-agent environments. Through smart contracts, the DAO defines and enforces permissions, reward distributions, and resource usage for task collaboration in a distributed network, ensuring that each agent's behavior adheres to preset constraints. Meanwhile, all interactions between agents are recorded on blockchain to form audit logs, preventing malicious tampering and data loss, thereby guaranteeing transparency and traceability of operations. When client agents collaborate with untrusted remote agents, the DAO leverages on-chain reputation systems, policy verification, and arbitration mechanisms to effectively reduce strategic risks, achieving "trustless trust". Furthermore, the DAO complements MCP and A2A protocols: MCP provides the operational context while DAO ensures the verifiability and decentralized execution of actions; A2A establishes communication channels while DAO guarantees compliance and accountability of communication results.

D. Tool and Data Resource

The underlying infrastructure of ARG is composed of three types of resources, which are continuously integrated via MCP to provide agents with environmental awareness, task feedback, and capability augmentation, as follows:

(1) Toolkit: Toolkit includes external APIs, computational functions, and plugin tools that provide executable operations for agents across various tasks.

(2) Function library: Function library encapsulates domain-specific functional components (e.g., path planners and image analyzers), enabling enhanced capabilities and rapid transfer learning for agents in multimodal tasks.

(3) Database: Database stores structured data, user records, intermediate states, and on-chain transactions, supporting continuous learning and long-term reasoning for agents.

Clearly, ARG extends and upgrades the traditional RAG paradigm by not only incorporating external knowledge retrieval but also integrating tool invocation, function execution, and decentralized data interaction into a unified multi-agent context. While RAG focuses primarily on enhancing language models through retrieved textual information, ARG transforms LLMs into active agents capable of perceiving dynamic environments, coordinating with peers, executing domain-specific functions, and maintaining memory through structured data. This shift enables ARG to support more complex, interactive, and autonomous decision-making processes across diverse scenarios, such as smart manufacturing [42], intelligent transportation and logistics [43], and city management [44].

E. Operational Process

The detailed operational process of ARG systems, spanning from task issuance to completion, is presented to demonstrate the system's transparency, integrity, and accountability, and consists of the following steps.

Step 1: Task reception and context sensing. When a user or an external system sends a request to an agent, the agent, which is regarded as a client agent, perceives the necessary contextual information through MCP that includes the task background, relevant history, system status, and tool resources.

Step 2: Context retrieval and assembly. Based on the perceived information, the client agent dynamically constructs a context structure for LLMs by retrieving and integrating tools, functions, and data resources.

Step 3: Context injection and task assessment/completion. The client agent performs a preliminary evaluation of the task to determine whether it can be completed independently. If so, it leverages the contextual information and tools integrated in the previous step to execute the task. Otherwise, it generates task metadata, including structured objectives, expected outcomes, and required capabilities, as preparatory material for a proposal.

Step 4: Task registration and proposal for DAO. If the

task requires collaborative processing, the client agent registers the task metadata on-chain and submits a proposal through a smart contract to request cooperative support for remote agents in the DAO.

Step 5: Member voting and task allocation. Upon receiving the proposal, remote member agents volunteer for collaboration and engage in decentralized voting, after which the smart contract finalizes role assignments and resource allocation based on the voting outcome.

Step 6: A2A discovery and task dispatching. The client agent establishes a collaborative connection with the selected remote agents via the A2A protocol and transmits the subtask descriptions and contextual information in an encrypted manner.

Step 7: Collaborative execution and contextualization. Upon receiving the subtask, the remote agents follow the same process as the client agent by performing steps 1, 2, and 3 to complete the task. Throughout the execution, they synchronize the task status, intermediate results, and any exceptions with the client agent in real time via the A2A protocol.

Step 8: Result delivery and reasoning integration. Remote agents deliver the final results to the client agent via an encrypted channel, and the client agent analyzes and consolidates these results to formulate a response.

Step 9: Log recording and incentive settlement. The context and activity logs of the entire execution process are recorded on the blockchain, and smart contracts distribute rewards to each participating agent and update their on-chain reputations. This marks the end of the task execution.

V. Characteristic Analysis

The design of the ARG emphasizes five core features: parallelism and throughput, scalability across domains and load, fault tolerance and graceful failure, resource efficiency through delegation, as well as data security and privacy protection, which collectively support the effectiveness and flexibility of the agent in handling complex tasks.

(1) Parallelism and throughput: ARG supports task distribution across multiple agents via the A2A protocol and enables parallel execution through MCP. Achieving significant speedups requires that the tasks themselves be inherently parallelizable and that intelligent coordination mechanisms are in place to manage inter-task dependencies and integrate results, but coordination itself introduces additional overhead.

(2) Scalability across domains and load: ARG enables easier scalability by adding more agents or MCP servers. However, real-world scalability depends not only on the protocols themselves but also on factors such as discovery service speed, network latency, coordination complexity, and the performance of the underlying AI models. Additionally, the interface between A2A and MCP can itself become a performance bottleneck.

(3) Fault tolerance and graceful failure: From a decentralized perspective, ARG offers potential for backup options. Building a robust decentralized fault-tolerance mechanism requires agents to reliably detect failures, evaluate alternatives, and dynamically replan. This approach makes agents far more complex than simple error-handling units, endowing them with greater autonomy and coordination capabilities to ensure high availability and robustness in a system without centralized control.

(4) Resource efficiency through delegation: ARG facilitates intelligent task allocation and coordination by assessing task complexity and agent capabilities, while utilizing encapsulated functions or off-the-shelf tools to simplify tasks, thereby enabling efficient resource utilization and effective task execution. This requires the prior collection and efficient retrieval of tools and functions.

(5) Data security and privacy protection: ARG leverages a blockchain-based DAO with smart contracts to ensure transparent, fair, and immutable management of task permissions, resource use, and rewards. All agent interactions are recorded on-chain for traceability and auditability, preventing tampering and data loss. It employs identity-based and permission-based access controls to protect sensitive data and supports privacy technologies like differential privacy and homomorphic encryption, complying with privacy regulations through blockchain's transparent auditing. However, this introduces increased computational and storage overhead, which may impact the real-time responsiveness and scalability of the system.

VI. Conclusion

This letter provides an in-depth analysis of the limitations of LLMs, RAG, and AI agents. To overcome these limitations, an ARG paradigm is proposed to achieve secure, efficient, scalable, and autonomous collaborative task execution among agents in complex open environments. In ARG, MCP, A2A protocol, and DAO are applied to enable seamless integration between LLMs and external tools, structured and secure communication among agents, as well as decentralized coordination and governance. We believe that ARG lays a foundational pathway for the progression from artificial intelligence to agentic intelligence, and holds strong potential for enabling truly self-learning, self-organizing, self-evolving, and context-aware intelligent systems with autonomous intelligence.

Acknowledgment

This work was supported by the Science and Technology Development Fund, Macao SAR (Nos. 0093/2023/RIA2 and 0145/2023/RIA3).

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., Language models are few-shot learners, in *Proc. 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2020, 159.

[2] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu et al., DeepSeek LLM: Scaling open-source language models with longtermism, arXiv preprint arXiv: 2401.02954, 2024.

[3] J. Yang, X. Wang, Y. Zhao, Y. Liu, and F.-Y. Wang, RAG-based crowdsourcing task decomposition via masked contrastive learning with prompts, *IEEE Trans. Comput. Social Syst*., to be published.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in *Proc. 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2020, 793.

[5] G. Izacard and E. Grave, Distilling knowledge from reader to retriever for question answering, in *Proc. 9th International Conference on Learning Representations*, Virtual Event, 2021.

[6] F.-Y. Wang, From RAG/RAT to SAGE: Parallel driving for smart mobility, *IEEE Trans. Intell. Veh.*, 2024, 9(5), 4821–4825.

[7] 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, *Sci. China Inf. Sci.*, 2025, 68(2), 121101.

[8] J. S. Park, J. O'Brien, C. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, Generative agents: Interactive simulacra of human behavior, in *Proc. 36th Annual ACM Symposium on User Interface Software and Technology*, San Francisco, CA, USA, 2023, 2.

[9] N. Krishnan, Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications, arXiv preprint arXiv: 2504.21030, 2025.

[10] P. P. Ray, A review on agent-to-agent protocol: Concept, state-of-the-art, challenges and future directions, *TechRxiv*, to be published.

[11] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F. Wang, Decentralized autonomous organizations: Concept, model, and applications, *IEEE Trans. Comput. Social Syst.*, 2019, 6(5), 870–878.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, 6000–6010.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, 2019, 4171–4186.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv: 1907.11692, 2019.

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

[16] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Virtual Event, 2020, 7871–7880.

[17] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan et al., The Llama 3 herd of models, arXiv preprint arXiv: 2407.21783, 2024.

[18] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, and Y. Zhuang, Vicuna: An open-source Chatbot impressing GPT-4 with 90%* Chatgpt quality [Online], https://vicuna.lmsys.org, 14 April 2023.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., An image is worth 16×16 words: Transformers for image recognition at scale, in *Proc. 9th International Conference on Learning Representations*, Virtual Event, 2021.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., Learning transferable visual models from natural language supervision, in *Proc. 38th International Conference on Machine Learning*, Virtual Event, 2021, 8748–8763.

[21] OpenAI, GPT-4V(ision) system card [Online], https://openai.com/index/gpt-4v-system-card/, 16 November 2024.

[22] H. Liu, C. Li, Q. Wu, and Y. J. Lee, Visual instruction tuning, in *Proc. 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2023, 1516.

[23] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in *Proc. 40th International Conference on Machine Learning*, Honolulu, HI, USA, 2023, 19730–19742.

[24] M. A. Ferrag, N. Tihanyi, and M. Debbah, From LLM reasoning to autonomous AI agents: A comprehensive review, arXiv preprint arXiv: 2504.19678, 2025.

[25] N. Krishnan, AI agents: Evolution, architecture, and real-world applications, arXiv preprint arXiv: 2503.12687, 2025.

[26] M. Rawat, A. Gupta, R. Goomer, A. Di Bari, N. Gupta, and R. Pieraccini, Pre-Act: Multi-step planning and reasoning improves acting in LLM agents, arXiv preprint arXiv: 2505.09970, 2025.

[27] T. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, Cognitive architectures for language agents, *Trans. Mach. Learn. Res.*, 2024, 2024.

[28] D. Wang, Z. Ye, F. Fang, and L. Li, Cooperative strategic planning enhances reasoning capabilities in large language models, arXiv preprint arXiv: 2410.20007, 2024.

[29] P. Lu, B. Peng, H. Cheng, M. Galley, K. W. Chang, Y. N. Wu, S. C. Zhu, and J. Gao, Chameleon: Plug-and-play compositional reasoning with large language models, in *Proc. 37th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2023, 1882.

[30] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with ChatGPT, arXiv preprint arXiv: 2302.11382, 2023.

[31] 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 *Proc. 36th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022, 1800.

[32] T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. Cai, PromptChainer: Chaining large language model prompts through visual programming, in *Proc. CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, USA, 2022, 359.

[33] 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 *Proc. 11th International Conference on Learning Representations*, Kigali, Rwanda, 2023.

[34] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, AutoPrompt: Eliciting knowledge from language models with automatically generated prompts, in *Proc. Conference on Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, 2020, 4222–4235.

[35] X. L. Li and P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Bangkok, Thailand, 2021, 4582–4597.

[36] C. S. Mala, G. Gezici, and F. Giannotti, Hybrid retrieval for hallucination mitigation in large language models: A comparative analysis, arXiv preprint arXiv: 2504.05324, 2025.

[37] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J.

Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, Atlas: Few-shot learning with retrieval augmented language models, *J. Mach. Learn. Res.*, 2023, 24(1), 251.

[38] Y. Tang and Y. Yang, MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries, arXiv preprint arXiv: 2401.15391, 2024.

[39] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, and Z. Li, Retrieval-augmented generation with knowledge graphs for customer service question answering, in *Proc. 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Washington, DC, USA, 2024, 2905–2909.

[40] Y. Tian, Y. Wang, X. Wang, J. Yang, T. Shen, J. Wang, L. Fan, C. Guo, S. Wang, Y. Zhao et al., From retrieval-augmented generation to SAGE: The state of the art and prospects, *Acta Autom. Sin.*, 2025, 51, 1145–1169, (in Chinese).

[41] W. Feng, C. Hao, Y. Zhang, J. Song, and H. Wang, AirRAG: Activating intrinsic reasoning for retrieval augmented generation using tree-based search, arXiv preprint arXiv: 2501.10053, 2025.

[42] J. Lim, B. Vogel-Heuser, and I. Kovalenko, Large language model-enabled multi-agent manufacturing systems, in *Proc. 20th International Conference on Automation Science and Engineering*, Bari, Italy, 2024, 3940–3946.

[43] T. Liu, J. Yang, and Y. Yin, Toward LLM-agent-based modeling of transportation systems: A conceptual framework, arXiv preprint arXiv: 2412.06681, 2024.

[44] A. Kalyuzhnaya, S. Mityagin, E. Lutsenko, A. Getmanov, Y. Aksenkin, K. Fatkhiev, K. Fedorin, N. O. Nikitin, N. Chichkova, V. Vorona et al., LLM agents for smart city management: Enhancing decision support through multi-agent AI systems, *Smart Cities*, 2025, 8, 19.

Jing Yang received the BS degree in automation from Beijing University of Chemical Technology, China, in 2020. She is currently pursuing the PhD degree at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include crowdsourcing, parallel manufacturing, social manufacturing, cyber-physical-social systems, and artificial intelligence.

Yonglin Tian received the PhD degree in control science and engineering from University of Science and Technology of China, China, in 2022. He is currently an assistant researcher at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, China. His research interests include parallel intelligence, autonomous driving, and intelligent transportation systems.

Fei Lin received the BS degree in chemical engineering and technology from Shenyang University of Chemical Technology, China, in 2018, and the MS degree in intelligent technology from Macau University of Science and Technology, China, in 2023. He is currently pursuing the PhD degree at Department of Engineering Science, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao, China. His research interests include parallel intelligence, large language models, and embodied agents.

Fei-Yue Wang received the PhD degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He is currently a professor at State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, at School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, and also at Macau Institute of Systems Engineering, Macau University of Science and Technology, Macao, China. His research interests include methods and applications for parallel systems, social computing, and knowledge automation. He received the IEEE ITS Outstanding Application and Research Awards in 2009, 2011, and 2015, and the IEEE SMC Norbert Wiener Award in 2014. In 2021, he became the IFAC Pavel J. Nowacki Distinguished Lecturer. He is a fellow of the International Council on Systems Engineering, International Federation of Automatic Control, American Society of Mechanical Engineers, and American Association for the Advancement of Science.