Large Language Model Enhanced Multi-Agent Systems for 6G Communications

Feibo Jiang, Yubo Peng, Li Dong, Kezhi Wang, Kun Yang, Cunhua Pan, Dusit Niyato, and Octavia A. Dobre

Abstract

The rapid development of the large language model (LLM) presents huge opportunities for 6G communications — for example, network optimization and management — by allowing users to input task requirements to LLMs with natural language. However, directly applying native LLMs in 6G encounters various challenges, such as a lack of communication data and knowledge, and limited logical reasoning, evaluation, and refinement abilities. Integrating LLMs with the capabilities of retrieval, planning, memory, evaluation, and reflection in agents can greatly enhance the potential of LLMs for 6G communications. To this end, we propose CommLLM, a multi-agent system with customized communication knowledge and tools for solving communication-related tasks using natural language. This system consists of three components: multi-agent data retrieval (MDR), which employs the condensate and inference agents to refine and summarize communication knowledge from the knowledge base, expanding the knowledge boundaries of LLMs in 6G communications; multi-agent collaborative planning (MCP), which utilizes multiple planning agents to generate feasible solutions for the communication-related task from different perspectives based on the retrieved knowledge; and multi-agent evaluation and reflection (MER), which utilizes the evaluation agent to assess the solutions, and applies the reflection agent and refinement agent to provide improvement suggestions for current solutions. Finally, we validate the effectiveness of the proposed multi-agent system by designing a semantic communication system as a case study of 6G communications.

Introduction

The future generation of wireless communication, for example, 6G, is anticipated to provide exceptional data rates, ultra-low latency, and significantly enhanced capacity to accommodate a massive number of user devices. To support the above vision, several innovative techniques such as edge intelligence and Semantic Communication (SC) have been proposed, where Artificial Intelligence (AI)/Machine Learning (ML) has been applied as a key enabling technology. However, the current intelligent communication system design is mainly based on the traditional AI/ML that can be seen as a discriminative AI technique that faces several challenges, when applied in 6G communications.

Large AI Models (LAMs), as state-of-the-art pretrained foundation models, offer an entirely new paradigm for solving the challenges of the discriminative AI[1]. LAMs represent a significant advancement in generative AI, leveraging their immense size, powerful generalization, and vast amounts of training data to achieve state-of-the-art performance in various tasks. Their ability to understand intent and generate solutions opens up new possibilities for improving a wide range of applications for 6G communications. The main features of LAMs and how they overcome the challenges of discriminative AI in 6G can be described as follows:

Excellent Global Perspective and Decision-Making

Future communication systems are expected to operate in rapidly changing environments, due to a variety of factors, such as the movement of devices and network traffic fluctuations. However, the traditional discriminative AI/ML mainly relies on learning local and short-term features, leading to trapping local extreme values or having difficulties in learning the long-term dependency of the dynamic network as well as achieving stable operation in a scalable way.

LAMs, with even trillions of parameters, enable the capturing of a large number of features from a global perspective and it can memorize and capture spatio-temporal dependencies in changing environments at different scales. This mechanism enables the LAM to generate stable and timely responses, unlike traditional neural networks that require retraining to adapt to environmental changes. For example, the multi-head self-attention of LAMs enables comprehensive learning of dynamic factors in the network, such as user mobility and traffic fluctuations. This mechanism avoids the long-term forgetting effect caused by dynamic environments, leading to accurate global traffic prediction and resource allocation [2].

Remarkable Robustness and Generalizability

Future communication systems will support a variety of devices, for exam ple, Internet of Things (IoT) or Unnamed Aerial Vehicles (UAVs), as well as provide various management strategies, like

Visit the project homepage: https://github.com/jiangfeibo/CommLLM.git

Digital Object Identifier: 10.1109/MWC.016.2300600

Feibo Jiang and Yubo Peng (corresponding author) are with Hunan Normal University, China; Li Dong (corresponding author) is with Hunan University of Technology and Business, China and also with Xiangjiang Laboratory, China; Kezhi Wang is with Brunel University London, UK; Kun Yang is with Nanjing University, China; Cunhua Pan is with Southeast University, China; Dusit Niyato is with Nanyang Technological University, Singapore; Octavia A. Dobre is with Memorial University, Canada.

beamforming design, user association, and edge resource allocation. However, the traditional discriminative AI/ML is mainly based on learning task-specific features, for example, only focusing on one type of task [3].

LAMs are typically trained on various types of data and tasks to enhance their generalization. The vast amounts of data allow LAMs to robustly capture intricate network patterns and nuances from heterogeneous devices, imbalance data and different tasks during training. For example, by learning the Channel State Information (CSI), the constraints for computation, communication, and storage resources of various edge devices and edge servers, it is possible to design a universal offloading model that achieves offloading optimization and resource scheduling for different system models or optimization objectives using prompts, without the need for retraining LAMs.

Astounding Understanding and Creativity

Future communication systems need to provide customized solutions for different application scenarios. For example, in autonomous driving services, the system requires extremely low latency and high reliability transmission, whereas in IoT applications, it must support a massive number of connections. Traditional discriminative AI/ML consists of small models trained for specific application scenarios, limiting them to those particular contexts.

Based on their exceptional understanding capabilities, LAMs can proactively analyze user demands and preferences in 6G networks, enabling the comprehension of various application scenarios and the provision of personalized computing and communication services for different applications. Leveraging their astonishing creativity, LAMs can dynamically plan, configure, and optimize the future communication networks through self-learning and self-adaptation abilities.

Leveraging their astonishing creativity, LAMs can dynamically plan, configure, and optimize the future communication networks through self-learning and self-adaptation abilities.

In this article, we describe the possible roles of Large Language Models (LLMs) in future communication networks. To overcome the current challenges of applying LLMs to 6G, we propose CommLLM, an LLM enhanced multi-agent system. This system is equipped with customized communication knowledge and tools, leveraging the collaboration and interaction among multiple agents to optimize task-solving capabilities in 6G networks. Specifically, users express their task requirements by natural language firstly. Then, Multi-agent Data Retrieval (MDR) is proposed to query and summarize domain-specific knowledge in 6G communications from private data. Next, a novel Multi-agent Collaborative Planning (MCP) decomposes the original task based on retrieved communication knowledge, generates multiple feasible sub-task chains and solves them. Subsequently, the Multi-agent Evaluation and Reflection (MER) is proposed to evaluate, reflect, and improve the current feasible solutions. As a whole, these form a self-learning and adaptive multi-agent system for solving communication-related problems by natural language. Finally, we validate the effectiveness of the multi-agent system through a case study.

How LLMs Support 6G Communications

LLMs constitute the most significant category of LAMs. On one hand, LLMs exhibit stronger comprehension, decision-making, and robustness compared to traditional ML models widely used in wireless networks. This higher level of intelligence brings new opportunities for sensor, communication, and computation in 6G wireless networks, enabling efficient and scalable general-purpose wireless intelligence. On the other hand, the massive and diverse wireless data, along with ubiquitous wireless devices in 6G wireless networks, provide powerful support in terms of data and computational resources for LLMs [1]. Therefore, the application of LLMs to enhance the intrinsic intelligence of 6G networks holds significant importance.

The Roles of LLMs in 6G Communications

In 6G communications, LLMs can play the following roles.

Data Generator: LLM is a powerful generative AI, which can generate specific types of data based on their domain knowledge. Some novel generative structures (e.g., autoregression decoder and diffusion model) are introduced to LLMs for creating data efficiently. For instance, LLMs can generate high-quality synthetic CSI data without identification information for network optimization, encompassing aspects such as positioning, bandwidth allocation, and network architecture design. This data can assist operators in more effectively planning and developing their 6G networks without violating privacy.

Knowledge Organizer: LLMs can reprocess or mine raw data for knowledge extraction and analysis, which take both user requirements and raw data processed as input and utilize the extensive domain knowledge of LLMs to deduce new information. For example, LLMs can be introduced as the knowledge base to assist the encoder of SC and can reduce ambiguity and enhance semantic understandings [4].

Task Scheduler: LLMs can understand instructions and schedule algorithms or protocols to collectively address complicated communication tasks. By using LLMs as a bridge between communication requirements and solutions, LLMs can manage and invoke proper algorithms, and collaboratively fulfill task requirements while producing results. For instance, LLMs can autonomously allocate service areas to different UAVs, plan their trajectories, guide UAVs to avoid obstacles, and provide critical communication links and computational resources in an emergency environment [5].

System Designer: LLMs possess powerful natural language understanding and deduction capabilities, enabling them to design communication systems based on given system requirements. They exhibit strong multi-task processing and multi-module design capabilities within the boundaries of various training data. For instance, utilizing its intrinsic AI knowledge, LLMs can automatically design a federated learning system based on a given functional description of Federated Averaging (FedAvg) algorithm, and continuously optimize the system through prompt engin eering [6].

Challenges of Applying LLM in 6G Communications

Challenges arise when applying LLMs to 6G, and they can be categorized as follows.

Challenge 1: Untimely Technical Data: LLMs are typically trained on static datasets, while data and information in 6G communication may constantly change. Especially, a vast amount of data

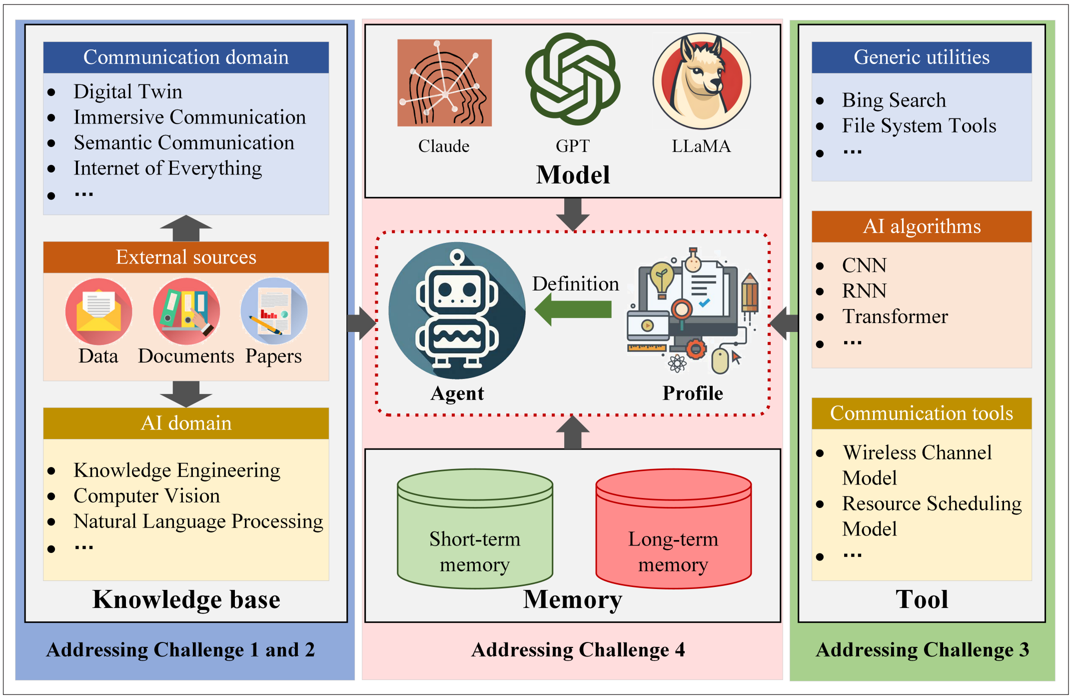

FIGURE 1. LLM enhanced agent system.

can be generated daily at the edge, and a significant portion of this data comprises private and technical information, rendering it inaccessible for public training purposes. As a result, the LLMs may not capture the latest data and concepts, emerging protocols and standards in a timely manner. This can lead to outputs that are less accurate or not fully aligned with the current communication system.

Challenge 2: Lack of Communication Knowledge: 6G communication has specific technical requirements and constraints, such as extremely low latency and very high data rate. However, LLMs may be trained on general data sets, which lack an in-depth understanding of communication knowledge. For example, the latest GPT-3.5 does not have an inherent understanding of what the SC system is and its structure. Instead, it interprets the SC system as a mode of semantic interaction employed by humans. This can result in degraded performance of the LLM in communications, as it may struggle to accurately predict and optimize specific task parameters and system performance.

Challenge 3: Insufficient Logical Reasoning: 6G communication involves complex signal processing, optimization, and decision-making tasks. Although LLMs excel in language generation and comprehension, they have limitations in logical reasoning and inferential capabilities. For instance, tasks like wireless channel estimation or resource scheduling require intricate reasoning and decision-making abilities, which LLMs may struggle to perform accurately. This may result in logical errors or a lack of understanding of causal relationships, which is common in optimization tasks in 6G communications.

Challenge 4: Inadequate Evaluation and Refinement: The evaluation of outputs from LLMs by the single-instance result can be challenging due to the complexity and diversity of wireless communication environments. Assessing the LLM's performance and effectiveness in real-world scenarios requires considering multiple factors, such as channel conditions, user requirements, and mobility. Therefore, evaluating LLMs necessitates adopting multiple perspectives, considering various factors, and providing feedback to continuously refine the performance of LLMs to ensure reliability and usability.

Hence, we could improve the performance of LLMs in 6G communications by training on up-to-date datasets, incorporating communication knowledge and reasoning capabilities, and developing more comprehensive evaluation and refinement methods.

LLM Enhanced Multi-Agent Systems

LLM Enhanced Agent System

AI agents can address the above challenges, and they are computational entities designed to simulate or mimic intelligent behavior, possessing traits such as autonomy, reactivity, and communication capacity [7]. Due to their versatile and remarkable capabilities, LLMs are regarded as powerful tools for constructing AI agents. In the article, we define a typical LLM enhanced agent system in Fig. 1, which has the following components:

- *Knowledge base* represents the component that stores the latest private data, consisting of communication standards, documents, and papers that can be connected to the agent, and provides a way to index, query, and update domain-specific knowledge in 6G communications.
- *Tools* are interfaces that an agent can use to process tasks. They can be generic utilities (such as Bing search and file system tools), endogenous intelligence in the LLM (such as AI model), or customized communication models (such as the wireless channel model). Tools can also be defined by the user, or integrated from external sources.
- *Memory* is the component that stores the intermediate and final outputs of agents for evaluation and reflection, providing a way to manage,

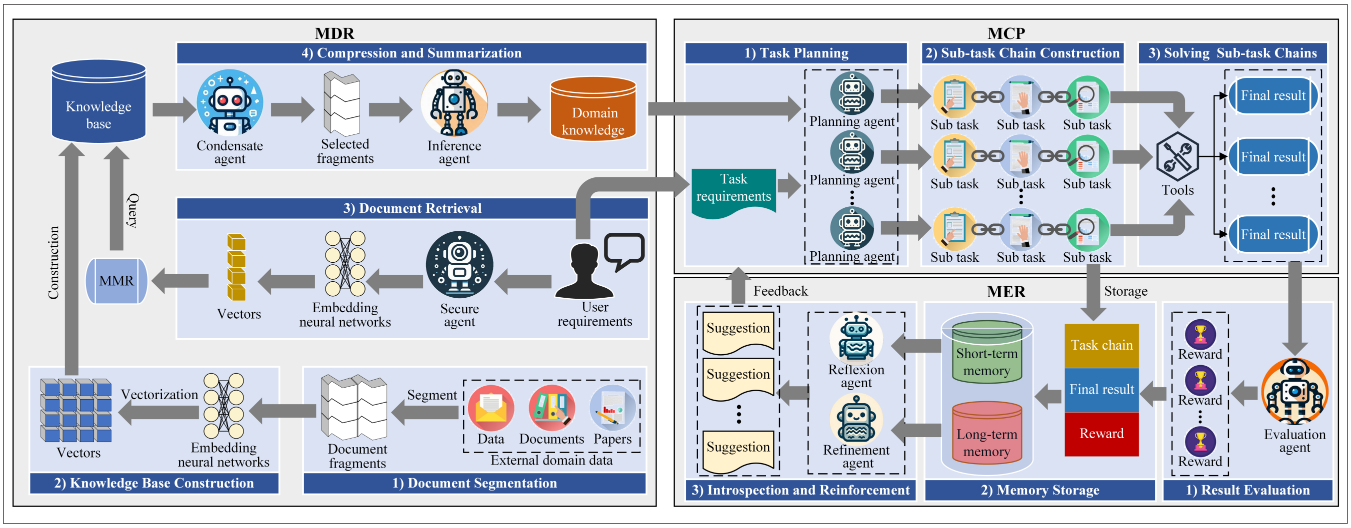

FIGURE 2. The proposed LLM enhanced multi-agent system.

retrieve, and modify these historical outputs.
- *Model* is introduced to comprehend and interpret human language inputs, extracting meaning, intent, and context. Agent system supports different LLMs such as OpenAI's GPT, Meta's LLaMA and Anthropic's Claude.
- *Agent* represents the entity that interacts with LLMs, memories, tools and knowledge base, providing a way to plan, control, introspect and communicate with other agents using LLM with a profile. The profile defines and manages the characteristics and behaviors of an agent by specific prompts. It encompasses a set of parameters and rules that describe various attributes of the agent, including its role, goals, capabilities, considerations, and behavioral patterns.

Overview of the Proposed Multi-Agent System

We design CommLLM, an LLM enhanced multi-agent system for 6G communications, which constructs a specialized knowledge base and tools for 6G communications, and possesses planning, memory, tool utilization, and introspection capabilities beyond protogenetic LLMs. Moreover, to avoid biases and hallucinations caused by a single agent, we introduce multi-agent collaboration, which involves designing multiple agents to engage in multi-round cooperation.

As depicted in Fig. 2, the process begins with the MDR module querying communication knowledge from external data sources based on user requirements by natural language. This step involves several agents, including a *secure agent,* a *condensate agent*, and an *inference agent*. Once the communication knowledge is obtained along with task requirements, they are fed into the MCP module. Then, multiple planning agents and sub-task chains are employed to formulate solutions for the given task and retrieved knowledge. With assistance from the tools, we can derive the final results based on solving the sub-task chain. Subsequently, the MER module introduces an evaluation agent that assesses the final results of each sub-task chain and assigns corresponding rewards. Furthermore, it uses a *reflection agent* and a *refinement agent* to provide refined suggestions. Finally, these suggestions are looped back to the MER module, guiding it to re-plan and generate new sub-task chains and results. By iterating through this process, we eventually reach an optimal solution that is then delivered to the user by natural language.

Multi-Agent Data Retrieval

MDR enables LLMs to extract and summarize knowledge from external privacy data sources regarding the 6G communications. MDR can also use the LLM as a reasoning engine over new domain knowledge provided in the knowledge base. The steps of MDR are as follows.

Document Segmentation: External domain data, including the latest communication standards, documents, and papers in various formats, can be loaded and then segmented to improve the efficiency of data retrieval. The segmentation process involves dividing the documents into multiple coherent and meaningful fragments while maintaining semantic coherence and integrity.

Knowledge Base Construction: The segmented document fragments are transformed into numerical vectors using embedding neural networks. Document fragments with similar semantic content will have similar vectors in the numerical space. By comparing these vectors, we can identify similar text fragments. The segmented document fragments and their corresponding embeddings are stored in vector format to construct a knowledge base, facilitating subsequent retrievals.

Document Retrieval: A secure agent is employed to scrutinize user requirements, thereby preventing any unauthorized requests or potential injection attacks [8]. Once validated, these legal requirements are transformed into vectors through the use of embedding networks. These requirement vectors are then compared with document vectors already stored in the knowledge base, and the most closely matching vectors along with their corresponding document fragments are selected. To increase retrieval diversity and minimize redundancy during this querying process in the knowledge base, we employ the Maximum Marginal Relevance (MMR) for selecting document fragments.

Compression and Summarization: To reduce irrelevant information in the documents, a condensate agent is utilized to compress the documents, resulting in more accurate and focused results [8]. The selected fragments, along with the query, are

Agent	Function	Ref.
Secure agent	Prevents any unauthorized requests or potential injection attacks	[8]
Condensate agent	Compresses retrieved documents for more accurate and focused results	[8]
Inference agent	Concludes specialized domain knowledge corresponding to user's requirements	[9]
Planning agent	Decomposes the original task into a series of sub-tasks	[10]
Evaluation agent	Evaluates results of sub-task chains and calculate the rewards of planning agents	—
Reflection agent	Reflects the fine-grained information from the short-term memory	[11]
Refinement agent	Analyses coarse-grained information from the long-term memory	[12]

TABLE 1. Summary of agents.

then inputted into an inference agent to obtain specialized communication knowledge corresponding to the user's requirements by natural language [9].

Multi-Agent Collaborative Planning

MCP generates multiple feasible sub-task chains by constructing multiple planning agents. By combining their individual knowledge and planning capabilities from different perspectives, the quality of response in solving complex problems is enhanced. The steps of MCP are as follows.

Task Planning: Based on the current task requirements and retrieved communication knowledge, multiple planning agents are initialized. Each agent employs either the CoT [9] or Plan-and-Solve approach [10] to decompose the original task into a series of sub-tasks.

Sub-Task Chain Construction: Considering the order and dependency relationships of all sub-tasks, a sub-task chain is constructed by connecting the individual sub-tasks in a sequential or parallel manner, ensuring the coherence and uniformity of all sub-tasks.

Solving Sub-Task Chains: Each sub-task chain is then solved by invoking either the intrinsic general-purpose tools or external custom tools to address each sub-task separately until the final result of the sub-task chain is obtained.

Multi-Agent Evaluation and Reflection

MER is used to evaluate the quality of the results generated by MCP and then to reflect on the planning results by memory, thereby facilitating automatic learning, continuous improvement, and self-refinement. The steps of MER are as follows.

Result Evaluation: All sub-task chains and their results from MCP are collected and the rewards of all planning results are calculated using an evaluation agent.

Memory Storage: A comparison is made between the current sub-task chain and historical sub-task chains. Task chains with significant differences in semantic space are stored in the long-term memory, while task chains with similar semantics are stored in the short-term memory, along with their corresponding results and rewards.

Introspection: A *reflection agent* is employed to extract fine-grained information from the short-term memory, similar to how humans can recall recent details [11]. It allows for contemplating the performance of the current sub-task chains across historical schemes, providing valuable small-scale feedback for refinement.

Refinement: A refinement agent is utilized to reference coarse-grained information from long-term memory, similar to how humans extract important experiences from long-term decisions [12]. It involves contemplating the performance of the current sub-task chains from a global perspective and providing large-scale feedback for improving the sub-task chains.

The introspection and refinement enable us to examine the contents of sub-task chains from different scales, gaining an in-depth understanding of the effectiveness and applicability of each sub-task.

Based on the feedback of the suggestions proposed by introspection and refinement, the MCP is re-driven to generate new sub-task chains, which are then evaluated through MER. This iterative process continues until the optimal scheme is obtained. All the agents used in the CommLLM system, along with the related work that inspired their design, are summarized in Table 1.

Case Study

In this section, we introduce a case study in 6G communications, that is, CommLLM for SC, which is expected to become the key application of next-generation intelligent communication paradigms. Specifically, we utilize the proposed multi-agent system to automatically generate an SC model based on user requirements and resource constraints.

SC Model Designed by the Multi-Agent System

As illustrated in Fig. 3, we apply the multi-agent system to construct the SC model. In this study, we chose GPT-3.5 as our LLM. The input content is expressed in natural language, encompassing user requirements, design objectives and constraints, as well as evaluation metrics as follows:

Input: *Please produce Python code that implements an SC model for text transmission. The Additive Gaussian White Noise (AWGN) can serve as the physical channel. The Bilingual Evaluation Understudy (BLEU) score is adopted as the metric that finally evaluates the SC model. The total number of model parameters does not exceed 2,000,000 for the resource constraints of devices.*

It is worth noting that the current multi-agent system lacks any knowledge about the SC system beyond the input, as the native GPT-3.5 does not understand the structure and functionality of SC. The multi-agent system needs to autonomously learn from the knowledge base and address these issues. The output is the outline and implementation code for this SC model. In this study, the specific contents of the outline part are listed as follows:

Output: *The Python-based SC model has been successfully implemented, incorporating all necessary modules. The semantic encoder and decoder are realized using Long Short-Term Memory (LSTM) networks. The channel encoder and decoder are constructed based on the Multilayer Perceptron (MLP) architecture. The final SC model can achieve a 0.68 BLEU score when SNR is 10 dB. In addition, the total number of model parameters is 1,826,762.*

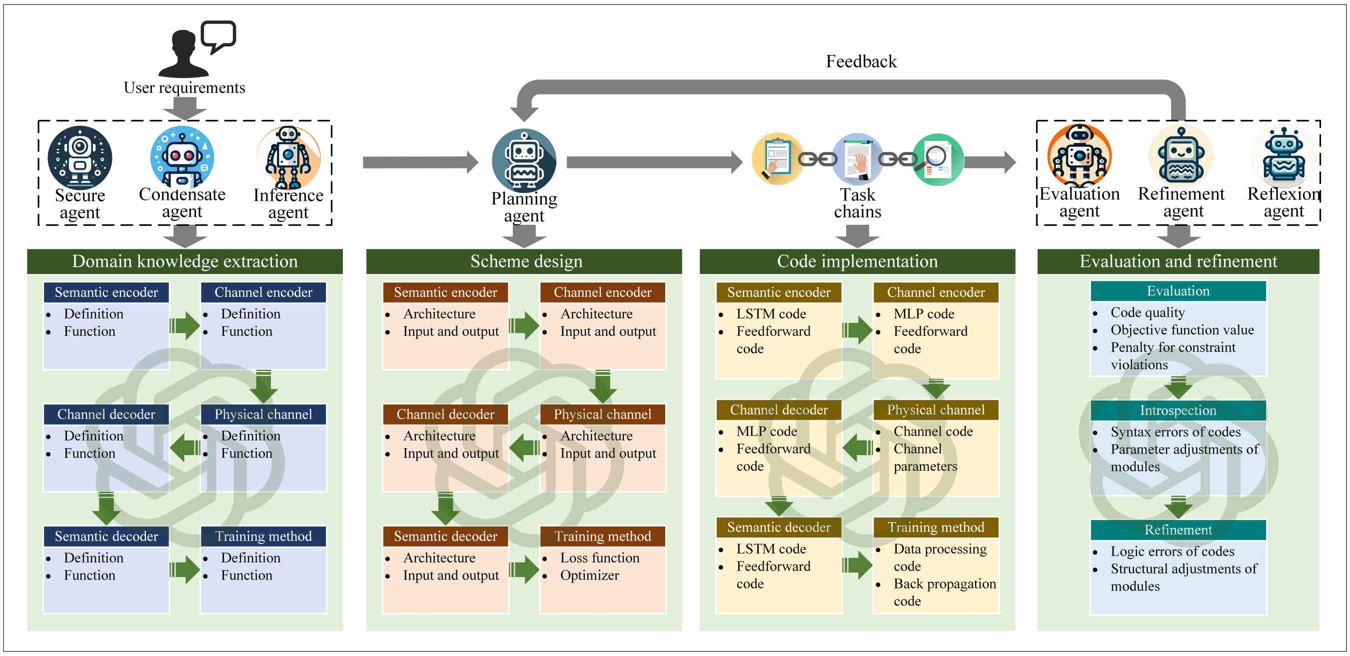

FIGURE 3. The illustration of implementing the SC model based on the LLM-enhanced multi-agent system.

We have gathered a selection of research papers from arXiv that are relevant to 6G communications to build a basic communication knowledge base. In addition, we have included essential communication models, such as channel models, as specialized communication tools. Next, the specific progress of applying the multi-agent system to design the SC model is as follows:

- The *secure agent* is utilized to validate the legitimacy of the input, while the relevant arXiv papers (e.g., [13]) in the knowledge base are retrieved. Then, the *condensate agent* is applied to refine the retrieved SC knowledge in papers. Next, the *inference agent* amalgamates user input and SC knowledge to distill relevant SC knowledge for constructing the SC model, which includes the definitions and functions of each module in the SC.
- Once the necessary SC knowledge is obtained, each *planning agent* formulates a specific subtask chain for the SC model. This chain delineates the architecture of each module in the SC model, including their respective inputs and outputs, network structure, as well as training settings (e.g., loss function and optimizer).
- Sub-task chains are then handled by the endogenous AI code generation tool of the LLM. Each sub-task chain manages one feasible scheme of the SC model including semantic encoder/decoder, channel encoder/decoder, and other codes (e.g., data processing, feedforward and backpropagation). The wireless channel code is generated by predefined communication tools (e.g., channel models). This results in obtaining precise Python code.
- The *evaluation agent* subsequently assesses the quality of generated codes. The evaluative score encompasses three components: the quality of the generated code, the value of the objective function, and the penalty for constraint violations.
- Based on these evaluation results, the reflection agent gives fine-grained introspective comments (e.g., syntax errors or parameter adjustments), and the *refinement agent* gives coarse-gained reinforced comments (e.g., logic errors or structural adjustments) taking into account both strengths and weaknesses of the current SC model.
- These proposed improvement suggestions are then fed back into the planning agent for introspection and refinement purposes. This process undergoes iterative optimization until satisfactory results are obtained.

Simulation Results

All schemes and experimental codes are generated by GPT-3.5, and all parameter optimizations are performed automatically by the multi-agent system. We only assign two planning agents in the multi-agent system. This implies that two independent SC models, each based on a unique scheme, are concurrently generated. We also set the number of iterations to four. Figure 4 presents simulation results for the SC model with different structures. It displays evaluative scores for both schemes. Notably, as the iterations advance, there is a distinct enhancement in evaluative scores. Initially, Scheme 2 trailed behind Scheme 1 in terms of evaluation scores. However, in the second iteration, the semantic encoder-decoder of Scheme 2 underwent coarse-grained refinement and transformed into the LSTM structure. As a result, Scheme 2 eventually caught up and surpassed Scheme 1. The adopted network architectures in the final SC models of the two schemes are described as follows.

Scheme 1: An MLP with 3 layers is employed as the semantic encoder and decoder. The MLP with 2 layers is integrated into the channel encoder and decoder. The SC model is trained using the SGD optimizer.

Scheme 2: An LSTM with 4 layers is utilized for the semantic encoder and decoder. Similarly, the MLP with 2 layers is used for both channel encoding and decoding. Training of this SC model is based on the Adam optimizer.

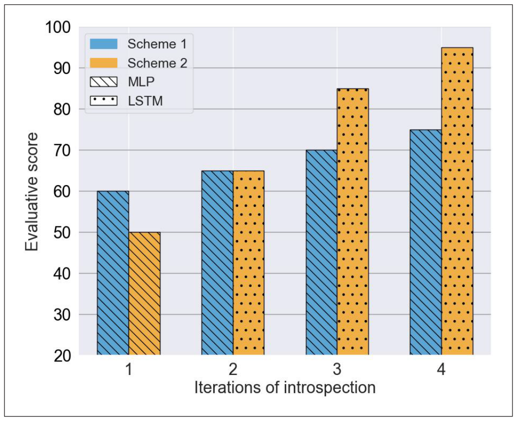

FIGURE 4. Evaluative score versus iteration number.

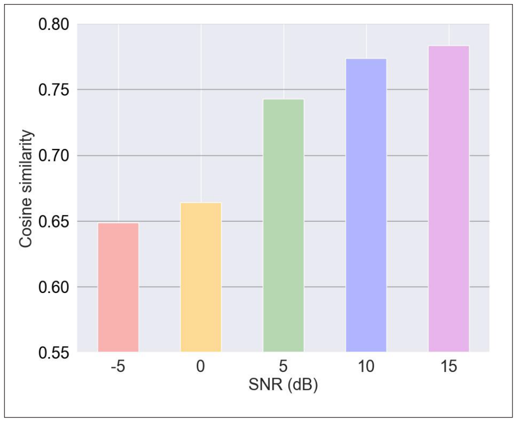

FIGURE 5. Cosine similarity versus SNR.

In summary, this simulation demonstrates the proposed multi-agent system's ability to autonomously generate an SC model while iteratively refining it through self-introspection and refinement.

To assess the effectiveness of the generated SC model by Scheme 2, we utilize the Cornell Movie-Dialogs Corpus dataset [14], a collection of dialogues from 617 film scripts, to train the SC model. Specifically, we use 8,000 dialogues for training and reserve 2,000 dialogues for testing. The training epoch is set at 50. As for the evaluation metric, we employ a BERT-based semantic evaluation method complemented by cosine similarity [15]. BERT is a pre-trained LLM capable of understanding generated text semantically and transforming the text into feature vectors, using the cosine similarity of the sentence-level vectors as an assessment metric more accurately reflects the semantic similarity between whole sentences. Subsequently, we compute the cosine similarity between these text encodings from raw and recovered text data. Figure 5 depicts the semantic similarity results of our SC model on the test set while varying the SNR. The figure clearly illustrates that the performance of our SC model improves as the SNR increases. These findings not only showcase the functionalities but also underscore the effectiveness of the SC model generated through the proposed multi-agent system.

Open Issues

Limited Resources

The multi-agent system relies heavily on the availability of LLMs and the private communication data at the edge. However, edge devices often have limited computing, storage and energy resources compared to powerful cloud servers. LLMs are resource-intensive models that may exceed the processing capabilities of edge devices, making it challenging to deploy multi-agent systems on the edge.

Cooperation and Competition

The proposed multi-agent system adopts a cooperative approach for all agents to accomplish the design of the SC system. The interaction mode among agents is important for the multi-agent system and LLMs facilitate diverse modes of interaction among agents. Exploring alternative modes of interaction, such as competition, and their application in 6G communications would be an intriguing research direction.

Real-Time Interaction

LLMs often suffer from slow response times, making them impractical for real-time, interactive 6G applications. Developing efficient and faster inference methods to enable real-time interaction with LLM-based agents is an ongoing challenge.

Conclusion

In this article, we proposed CommLLM, a multi-agent system that utilizes natural language to design solutions for 6G communications and provided a case study in SC tasks. The system leveraged multiple LLM-enhanced agents to collaborate, self-learn, self-improve, and effectively solve defined problems in 6G communications. Specifically, we first employed the MDR to query private data in the system and extracted communication knowledge relevant to the task requirements. Next, we utilized the MCP to generate feasible solutions from different perspectives. Subsequently, we employed the MER to evaluate and reflect on the current solutions, provide improvement suggestions, and guide MCP in enhancing the solutions. Through iterations, an optimal solution was obtained. Finally, we demonstrated and validated the effectiveness of the proposed multi-agent system through the SC case study.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 41604117, 41904127, and 62132004, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2024JJ5270, in part by the Open Project of Xiangjiang Laboratory under Grant 22XJ03011, in part by the Scientific Research Fund of the Hunan Provincial Education Department under Grant 22B0663, in part by the National Research Foundation, Singapore, and the Infocomm Media Development Authority under the Future Communications Research & Development Programme, the Defence Science Organisation (DSO) National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019 and FCP-ASTAR-TG-2022-003), and in part by the Singapore Ministry of Education (MOE) Tier 1 (RG87/22).

References

[1] Z. Chen, Z. Zhang, and Z. Yang, "Big AI Models for 6G Wireless Networks: Opportunities, Challenges, and Research Directions," arXiv preprint arXiv:2308.06250, 2023.

[2] R. Zhang *et al.*, "Generative AI for Space-Air-Ground Integrated

Networks (SAGIN)," arXiv preprint arXiv:2311.06523, 2023.
[3] L. Dong *et al.*, "Deep Progressive Reinforcement Learning-Based Flexible Resource Scheduling Framework for Irs and UAV-Assisted Mec System," *IEEE Trans. Neural Networks and Learning Systems*, 2024, pp. 1–13.
[4] F. Jiang *et al.*, "Large AI Model-Based Semantic Communications," *IEEE Wireless Commun.*, vol. 31, no. 3, 2024, pp. 69–75.
[5] J. Zhong *et al.*, "A Safer Vision-Based Autonomous Planning System for Quadrotor Uavs With Dynamic Obstacle Trajectory Prediction and Its Application With LLMs," arXiv preprint arXiv:2311.12893, 2023.
[6] Y. Shen *et al.*, "Large Language Models Empowered Autonomous Edge AI for Connected Intelligence," arXiv preprint arXiv:2307.02779, 2023.
[7] Y. Xiao *et al.*, "Reinforcement Learning Based Energy-Efficient Collaborative Inference for Mobile Edge Computing," *IEEE Trans. Commun.*, vol. 71, no. 2, 2022, pp. 864–76.
[8] O. Topsakal and T. C. Akinci, "Creating Large Language Model Applications Utilizing Langchain: A Primer on Developing LLM App. Fast," *Proc. Int'l. Conf. Applied Engineering and Natural Sciences*, vol. 1, no. 1, 2023, pp. 1050–56.
[9] J. Wei *et al.*, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24,824–37.
[10] L. Wang *et al.*, "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models," arXiv preprint arXiv:2305.04091, 2023.
[11] N. Shinn *et al.*, "Reflection: An Autonomous Agent With Dynamic Memory and Self-Reflection," arXiv preprint arXiv:2303.11366, 2023.
[12] A. Madaan *et al.*, "Self-Refine: Iterative Refinement With Self-Feedback," arXiv preprint arXiv:2303.17651, 2023.
[13] C. Wang *et al.*, "Multimodal Semantic Communication Accelerated Bidirectional Caching for 6G MEC," *Future Generation Computer Systems*, vol. 140, 2023, pp. 225–37.
[14] C. Danescu-Niculescu-Mizil *et al.*, "You Had Me at Hello: How Phrasing Affects Memorability," *Proc. 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 892–901.
[15] N. Reimers and I. Gurevych, "Sentence-Bert: Sentence Embeddings Using Siamese Bert-Networks," *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. Natural Language Processing*, 2019, pp. 3982–92.

Biographies

Feibo Jiang [SM] received his B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. He received his Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2014. He is currently an associate professor at the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, China. His research interests include semantic communication, federated learning, Internet of Things, and mobile edge computing.

Yubo Peng received his B.S. and M.S. degrees from Hunan Normal University, Changsha, China, in 2019 and 2024. He is pursuing a doctor's degree from the School of Intelligent Software and Engineering at Nanjing University. His main research interests include semantic communication and large models.

Li Dong received the B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. She received her Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2018. She is currently an associate professor at Hunan University of Technology and Business, China. Her research interests include machine learning, Internet of Things, and mobile edge computing.

Kezhi Wang [SM] received a PhD degree in Engineering from the University of Warwick, U.K. Currently, he is a Senior Lecturer at the Department of Computer Science, Brunel University London, U.K. His research interests include wireless communications, mobile edge computing, and machine learning.

Kun Yang [F] received his PhD from the Department of Electronic & Electrical Engineering of University College London (UCL), UK. He is currently a Chair Professor in the School of Intelligent Software and Engineering, Nanjing University, China. He is also an affiliated professor of University of Essex, UK. His main research interests include wireless networks and communications, communication-computing cooperation, and new AI (artificial intelligence) for wireless. He is a Member of Academia Europaea (MAE), a Fellow of IEEE, a Fellow of IET and a Distinguished Member of ACM.

Cunhua Pan [SM] received the B.S. and Ph.D. degrees from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2010 and 2015, respectively. He was a Research Associate with the University of Kent, Canterbury, U.K., from 2015 to 2016. He held a postdoctoral position with the Queen Mary University of London, London, U.K., from 2016 and 2019, and was a Lecturer from 2019 to 2021. He has been a Full Professor with Southeast University since 2021. He has published over 120 IEEE journal papers. His research interests mainly include reconfigurable intelligent surfaces (RIS), intelligent reflection surface, ultrareliable low-latency communication, machine learning, UAV, Internet of Things, and mobile-edge computing.

Dusit Niyato [M'09, SM'15, F'17] is a professor in the College of Computing and Data Science, at Nanyang Technological University, Singapore. He received B.Eng. from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand and Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada. His research interests are in the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

Octavia A. Dobre [F] is a professor and Canada Research Chair Tier 1 at Memorial University, Canada. She is a Fellow of the Canadian Academy of Engineering, Fellow of the Engineering Institute of Canada, and elected member of the European Academy of Sciences and Arts.