Journal of Advances in Information Technology, Vol. 17, No. 5, 2026

LLM-Based Multi-agent Collaborative Spectrum Cognition Method

Lihao Liu, Jianhui Wang *, Jingjing Li, Nan Hu, Wenlin Zhang, and Da Zhang
School of Information and Communication Engineering, Information Engineering University, Zhengzhou, China
Email: liulihao@139.com (L.L.); wjhxxgc@163.com (J.W.); jing_57@163.com (J.L.);
hn19911029@outlook.com (N.H.); wenlinzzz@163.com (W.Z.); zhangdajh@163.com (D.Z.)
*Corresponding author

***Abstract*—Current spectrum cognition methods based on mathematical models and deep learning excel at single tasks with clear inputs and outputs, such as signal detection and classification. However, they generally lack autonomous task decomposition, logical reasoning, and tool coordination capabilities. Consequently, it is difficult for them to directly transform raw signal data into valuable information in complex and dynamic wireless communication environments. Multi-Agent Systems (MAS), due to their autonomy, distributivity, and collaborative nature, show broad application prospects in interdisciplinary fields like communications and artificial intelligence. This paper proposes an MAS-based spectrum cognition method, implemented through a "master-slave" MAS architecture jointly driven by the Qwen3 and MiniCPM4 Large Language Models (LLMs). The system consists of a Master Agent, a Signal Perception Sub-Agent, a Knowledge Retrieval Sub-Agent, and a Cognitive Reasoning Sub-Agent, working together to accomplish collaborative spectrum cognition tasks. In experiments, focusing on Wi-Fi signal analysis, we constructed a signal sample database using Universal Software Radio Peripheral (USRP) platforms. The proposed method automated data analysis, feature extraction, historical data querying, and preliminary signal emitter direction estimation for the database. The results demonstrate that the proposed method achieves autonomous task decomposition, signal sample feature identification, and spectrum cognition for Wi-Fi signal emitters. It highlights the potential of MAS in communication signal analysis and offers a new technical approach for intelligent communication detection, automated signal monitoring, and smart emitter identification.**

***Keywords*—multi-agent system, communication signal analyzer, spectrum cognition, large language model**

I. INTRODUCTION

Spectrum Cognition (SC) is a core technology for dynamic spectrum management. It aims to accurately detect available frequency bands and their usage states in complex electromagnetic environments. For 6G and future networks, SC must achieve high throughput, low latency, and high reliability simultaneously [1].

Traditional model-based methods, such as energy detection and matched filtering, struggle in congested real-world conditions [2, 3]. They face challenges like poor performance under low SNR, dependency on prior signal knowledge, and high computational complexity [4].

Deep Learning (DL) provides a data-driven alternative [5]. By converting raw I/Q data into images or sequences, CNNs excel in modulation classification [6, 7]. RNNs and LSTMs capture temporal dependencies for signal detection. Transformers enhance performance in low-SNR scenarios by modeling long-range dependencies [8]. In Specific Emitter Identification (SEI), DL models learn hardware imperfections for device fingerprinting [9, 10]. However, DL remains confined to a "one model, one task" framework. It lacks autonomous task decomposition and dynamic collaboration capabilities for complex cognitive tasks [11].

The integration of LLMs with MAS offers a transformative approach. LLMs serve as a universal interface for task understanding and planning. Frameworks such as AutoGen [12] and MetaGPT [13] have demonstrated the potential of LLM-based multi-agent collaboration. In particular, MetaGPT introduces a structured software-engineering paradigm with role-based agent collaboration, which complements AutoGen's flexible conversation-based orchestration. In a MAS, a master agent decomposes high-level natural language instructions into subtasks [14, 15]. Specialized sub-agents then collaborate to execute these tasks. This process mimics sophisticated cognitive functions, similar to how brain networks handle complex information. The LLM-driven MAS paradigm overcomes key limitations of previous methods. It enables flexible task planning, collaborative problem-solving, and intuitive human-computer interaction [16, 17]. This makes it highly suitable for open and complex spectrum environments, advancing the development of autonomous and intelligent spectrum cognition systems.

Therefore, this paper designs and implements a new spectrum cognition method named SpecAgent. Built on the AutoGen framework, SpecAgent employs two LLMs, Qwen3 [18] and MiniCPM4 [19], for task understanding

---

Manuscript received November 24, 2025; revised December 4, 2025; accepted March 22, 2026; published May 25, 2026.

doi: 10.12720/jait.17.5.1015-1027

and scheduling, while deep learning models handle signal feature analysis. Automated MAS have demonstrated superior performance in complex reasoning tasks by leveraging specialized roles, such as AgentInit [20]. A critical challenge in MAS design is ensuring that agents possess the specific expertise required for complex tasks. Previous works have addressed this through dynamic role verification mechanisms. For instance, AgentDropout [21] employed a "Selector" to validate generated expert roles before deployment. The system operates as a master-slave MAS comprising four specialized agents: The master agent decomposes complex problems and assigns tasks. The signal perception sub-agent manages signal collection, feature analysis, and formatted storage. The knowledge Retrieval sub-agent converts natural language into database queries. The cognitive reasoning sub-agent conducts deep analysis and logical reasoning on existing data. The main contributions of this paper are:

(1) This paper proposes a multi-agent collaborative spectrum cognition method, introducing an LLM-based MAS paradigm to communication signal processing. The SpecAgent method offers a novel approach for complex spectrum cognition tasks.

(2) This paper implements a cloud-edge collaborative system, deploying LLM-driven agents on resource-constrained edge nodes. This combines edge computing's low latency with LLM intelligence, validating feasibility for real-time applications.

(3) This paper validates functionality across multiple scenarios. Using Wi-Fi signal analysis and a USRP-based sample database, experiments in autonomous spectrum scanning, device fingerprint identification, and signal source localization demonstrate SpecAgent's advantages in automation, flexibility, and user-friendly interaction.

II. RELATED WORK

A. Spectrum Cognition and Sensing Methods

Spectrum cognition has traditionally relied on model-based approaches, such as energy detection and cyclostationary feature detection. While these methods are computationally efficient for specific signal types, they lack adaptability to complex, non-stationary electromagnetic environments.

The advent of Deep Learning (DL) has significantly advanced this field. Convolutional Neural Networks (CNNs) have been extensively applied to modulation classification and automatic modulation recognition (AMR), demonstrating superior feature extraction capabilities compared to statistical methods. Furthermore, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [22] have been utilized to capture temporal dependencies in signal sequences. In the domain of Specific Emitter Identification (SEI), deep architectures have achieved high accuracy by learning subtle hardware impairments from raw I/Q samples.

Recent advancements in spectrum cognition have increasingly leveraged deep learning architectures, particularly Transformers, to address complex and dynamic signal environments. Regarding Cooperative Spectrum Sensing (CSS), Fang *et al.* [23] proposed a hybrid CNN-Transformer architecture that integrates local feature extraction with global dependency modeling, significantly enhancing sensing accuracy in cognitive radio networks. To tackle the challenges of 3D dynamic scenarios involving Autonomous Aerial Vehicles (AAVs), Chen *et al.* [24] developed a Deep Denoising Transformer-based Spectrum Sensing Framework (DDTSF), incorporating depthwise separable convolutions and soft-thresholding to maintain robust performance under random primary user activities. In the domain of Automatic Modulation Classification (AMC), Jang *et al.* [25] introduced a Meta-Transformer framework based on meta-learning, enabling rapid adaptation to new modulation types even with limited training samples. Furthermore, Zhai *et al.* [26] explored the extraction of cross-domain features through a Dual-Path Signal Transformer, which simultaneously captures time- and frequency-domain characteristics to improve identification robustness against complex interference. These studies demonstrate that the fusion of spatio-temporal correlations and cross-domain information is becoming a pivotal trend in advancing spectrum sensing capabilities.

Although existing deep learning-based methods have achieved high accuracy in specific spectrum sensing tasks, they are inherently designed as single-task, isolated modules. They lack the semantic reasoning capability to autonomously adapt to unforeseen dynamic spectrum environments. In contrast, our proposed work leverages the cognitive abilities of LLMs to dynamically decompose complex spectrum tasks rather than relying on static, pre-trained models.

B. Large Language Models in Wireless Communications

The integration of Large Language Models (LLMs) has emerged as a transformative frontier for achieving comprehensive intelligence in 6G wireless communications. Recent research has established a foundational framework for the lifecycle of LLMs within network architectures, highlighting their potential to automate complex network Operations and Maintenance (O&M) and optimization tasks [27]. To bridge the gap between the high computational demands of LLMs and the resource constraints of mobile devices, advancements in mobile edge intelligence have introduced critical strategies such as model compression, split learning, and active inference-based offloading, enabling efficient generative AI services at the network edge [28]. Furthermore, the utility of LLMs has extended into structural network design, where specialized frameworks like the Large Language Model-based Combinatorial Optimizer (LMCO) have demonstrated superior performance over traditional meta-heuristic algorithms in solving complex wireless planning problems, such as optimal access point placement [29]. Building upon these foundations, current research in 6G Symbiotic IoT environments further leverages active inference to manage the complexities of task offloading for LLMs, ensuring that resource-constrained IoT devices can effectively utilize generative AI capabilities while navigating the dynamic demands of future communication networks [30]. To address the limitations of traditional data augmentation methods in capturing the inherent time-frequency characteristics of radio signals for Automatic Modulation Classification (AMC), Chen *et al.* [31] proposed a novel augmentation framework based on the Wavelet Transform (WT). By decomposing original IQ sequences into approximation and detail coefficients, the authors introduced Random Noise Sequence Replacing (RNSR) and its Multi-Wavelet variant (RNSR-MW) to effectively generate diverse augmented samples. Their findings demonstrate that this wavelet-domain augmentation strategy significantly enhances the recognition robustness of deep learning models in complex channel environments, offering a promising solution for few-shot scenarios in signal detection and parameter estimation. For resource-constrained wireless Internet of Things (IoT) environments, Han *et al.* [32] proposed a lightweight Automatic Modulation Classification (AMC) model based on the Informer architecture. This research innovatively utilizes 2D cross-section Spectral Correlation Function (SCF) curves as model inputs, effectively extracting statistical characteristics of non-stationary signals while enhancing noise resistance. To reduce computational overhead on embedded platforms, the architecture incorporates the ProbSparse self-attention mechanism and self-attention distilling techniques. These features significantly minimize memory consumption while maintaining long-distance robustness for signals across various Signal-to-Noise Ratios (SNRs). Experimental results demonstrate that this method achieves superior classification accuracy compared to state-of-the-art deep learning models, even with limited hardware resources. Collectively, these developments signify a shift toward more autonomous, self-optimizing, and edge-native wireless communication systems driven by generative intelligence. However, the majority of these studies are confined to high-level network management or theoretical simulations. They fail to address the critical gap of connecting LLM reasoning with low-level, real-time deterministic hardware constraints. To overcome this limitation, our method introduces a bridge between abstract LLM planning and low-level hardware execution.

These developments are aligned with earlier foundational discussions on AI-native 6G and edge intelligence, where communication systems are expected to evolve toward deeply integrated sensing, computing, and intelligence paradigms [33, 34].

C. Multi-Agent Systems for Complex Problem Solving

The evolution toward 6G wireless communications is increasingly characterized by the integration of Large Language Models (LLMs) across various network layers. At the orchestration level, Sun *et al.* [35] introduces a framework for managing complex tasks within integrated space-air-ground-sea environments. In vertical applications such as intelligent transportation, Fu *et al.* [36] establishes a foundational taxonomy for LLM-based perception and decision-making. As explored by Wang *et al.* [37] addressed the complexities of mission planning and cooperative search for diverse agent formations. Furthermore, the efficiency of these intelligent systems depends on optimized communication protocols; Liang *et al.* [38] demonstrates how attention-based aggregation of limited information can mitigate partial observability while preventing the dimensional explosion of communication data. Together, these works highlight a multi-tiered intelligence strategy—spanning from global network orchestration to efficient local agent communication—that is essential for the realization of smart wireless ecosystems.

The advancements in Multi-Agent Systems (MAS) have focused on optimizing agent organization. Notably, Tian *et al.* [20] proposed AgentInit, which utilizes an optimization-based method to dynamically generate agent roles and execution plans tailored to specific problem contexts. This approach contrasts with traditional static role assignments by enhancing adaptability in general logical reasoning tasks. However, the applicability of such dynamic initialization mechanisms in the Radio Frequency (RF) domain—characterized by high data volatility and specific hardware constraints—remains to be validated. A critical challenge in deploying LLM-based systems is the high computational overhead and latency associated with token generation. Wang *et al.* [21] introduced AgentDropout, a framework designed to identify and prune redundant agent interactions to reduce token consumption without compromising task performance. This method highlights the potential for optimizing inter-agent communication protocols. While existing works focus on general efficiency, this paper primarily investigates the functional feasibility of LLM-MAS in spectrum cognition, accepting initial latency overheads to ensure interpretability and execution completeness.

Beyond dynamic role initialization, recent studies have also explored broader orchestration and scalability mechanisms for multi-agent systems, including dynamic resource/task allocation and agent auto-scaling strategies. In this context, the DRTAG approach [39] provides a useful perspective on automatic scaling for agent-based systems, which is relevant to the future evolution of SpecAgent toward more adaptive and scalable deployments.

While current Multi-Agent Systems (MAS) excel in software-centric domains such as software engineering, direct application of these frameworks to spectrum cognition is hindered by the domain's strict latency and hardware execution requirements. Unlike existing literature that mentions MAS in isolation, our framework uniquely designs SpecAgent to synthesize MAS with physical SDR operations.

A fundamental challenge neglected by prior works is the semantic gap between the probabilistic nature of LLM outputs and the deterministic operational requirements of Software-Defined Radio (SDR) peripherals. Existing spectrum cognition AI models operate in purely digital, simulated environments. Our research synthesizes multi-agent reasoning with physical layer sensing. Specifically, we design our agents not just as text-generators, but as embodied controllers. By providing the agents with specialized tools (tool-use) and predefined API constraints for SDR platforms, our framework ensures that the high-level cognitive intent (e.g., "detect anomalies in the 2.4 GHz band") is accurately translated into executable, deterministic hardware commands (e.g., adjusting sample rates, tuning center frequencies). This transition from virtual simulation to physical hardware execution sets our proposed method apart from current literature.

III. METHODOLOGY

To achieve autonomous, collaborative, and intelligent processing of communication signals, this paper designs and implements a multi-agent collaborative spectrum cognition method named SpecAgent. This section provides a detailed description of SpecAgent from three aspects: the overall architecture, the functional design of each agent, and the collaborative workflow based on AutoGen. While recent advancements in MAS advocate for dynamic role generation, the study adopts a static Master-Slave topology with four predefined specialized agents. This architectural choice is driven by the deterministic constraints required for RF hardware control. Unlike pure software tasks, spectrum cognition involves direct interactions with SDR peripherals. A fixed topology ensures a verifiable command chain and minimizes the risk of generating hallucinated or unsafe hardware instructions. This static framework serves as a foundational baseline for evaluating the feasibility of LLM-integrated spectrum sensing before introducing complex dynamic adaptation mechanisms.The specific details are as follows. Although generic multi-agent frameworks like AutoGen provide the communication backbone, they lack the intrinsic capability to interpret the physical properties of electromagnetic signals. We selected AutoGen as the base orchestration layer due to its flexible conversational topology, which allows us to inject spectrum-specific domain knowledge into individual agents without retraining the core LLMs. The core contribution of SpecAgent lies in the Domain Adaptation Layer constructed atop the generic framework. Unlike standard software engineering agents, SpecAgent must bridge the gap between abstract user intent (e.g., "find the anomaly") and rigorous mathematical operations (e.g., FFT, filtering). We designed a specialized prompt structure that encapsulates signal processing constraints (bandwidth, sampling rate) into the agents' working memory, enabling them to handle invisible RF data effectively.

A. *Overall Architecture*

The core design philosophy of SpecAgent is to deconstruct complex signal cognition tasks into a series of subtask sequences, which are executed through a collaborative framework of multi-role agents. Physically, the architecture comprises two collaborative entities: a Computing and Data Center on a central node, hosting the Master Agent, Knowledge Retrieval Sub-Agent, and Cognitive Reasoning Sub-Agent with backend LLMs as their cognitive core; and a Signal Perception Frontend on an edge node, hosting the Signal Perception Sub-Agent and lightweight perception models connected to the USRP and omnidirectional antenna, as shown in Fig. 1. Logically, the architecture follows a three-layer model:

(1) Physical Layer: This layer serves as the system's interface with the electromagnetic environment. It consists of the omnidirectional antenna and the USRP. Its core function is to sense and capture signals in specified frequency bands. It then digitizes these signals into raw I/Q data streams, providing the data foundation for upper-layer analysis.

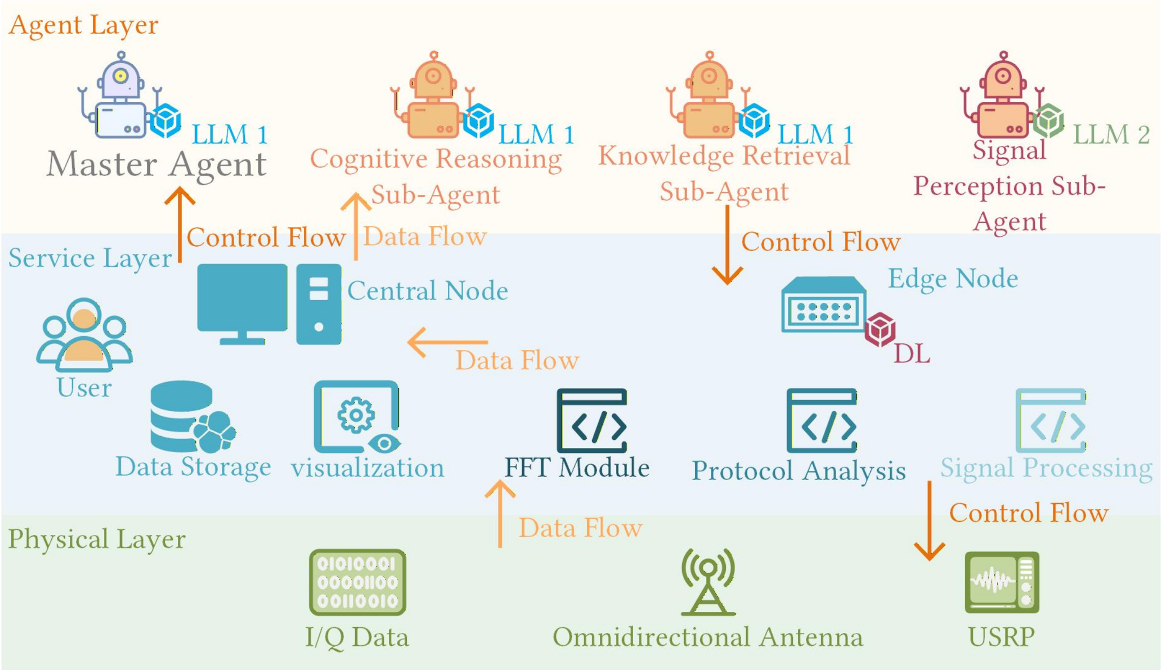


Fig. 1. SpecAgent overall architecture diagram.

(2) Service Layer: This layer provides core functional modules and data support for the Agent Layer. The central node deploys an Elasticsearch database for structured information storage and indexing, along with a signal visualization module. The edge node deploys a signal processing toolkit, which includes functions like FFT and Wi-Fi protocol parsing.

(3) Agent Layer: This layer provides SpecAgent with task planning, logical reasoning, and collaborative scheduling capabilities. It consists of one Master Agent and three functional Sub-Agents. Driven by models like Qwen3 and MiniCPM4, these agents collaboratively complete complex tasks by interactively calling tools from the Service Layer.

B. Functional Design of Each Agent

1) Master agent function design

The Master Agent acts as the planner and central dispatcher within SpecAgent. Powered by a backend LLM, it serves as the primary entry point for tasks and is responsible for decomposing complex problems and assigning subtasks. Its core capabilities are divided into four key areas:

(1) Task Planning Capability: The Master Agent receives and comprehends complex tasks expressed by users in natural language. For example, a user might input: “Scan the surrounding environment, identify the Wi-Fi device with the strongest signal, and estimate its approximate direction”. Leveraging the LLM’s semantic understanding, it breaks down such ambiguous user requests into a sequence of clear, executable subtasks and assigns them to appropriate sub-agents.

(2) Sub-Agent Dispatching Capability: Based on the nature of the decomposed subtasks, the Master Agent selects the most suitable sub-agent for execution. For instance, scanning tasks are assigned to the Signal Perception Sub-Agent, data query tasks to the Knowledge Retrieval Sub-Agent, and data analysis tasks to the Cognitive Reasoning Sub-Agent.

(3) Workflow Management Capability: The Master Agent maintains the context of the entire complex task and oversees the execution process. It manages dependencies, ensuring the output from one agent seamlessly becomes the input for the next, thereby guaranteeing the smooth progression of the complex task.

(4) Summarization and Presentation Capability: After all subtasks are completed, the Master Agent consolidates the results from various sub-agents. It then generates a well-structured and coherent final report presented to the user.

2) Functional design of signal-perceiving sub-agents combining hardware and software

The Signal Perception Sub-Agent acts as the radio engineer and serves as the bridge between SpecAgent and the physical world. It can call upon signal processing tools from the Service Layer. Its core capabilities are divided into four areas:

(1) Spectrum Scanning Capability: Based on instructions from the Master Agent (e.g., specified center frequency, bandwidth, scan duration), it invokes the USRP driver and GNU Radio to control the omnidirectional antenna for environmental signal acquisition, generating raw I/Q data.

(2) Wi-Fi Signal Parsing Capability: It processes the captured raw I/Q data in real-time, focusing on Wi-Fi signal analysis. This includes filtering Wi-Fi channels and using a binary data parsing module to decode and identify 802.11 frames. Key fields are extracted, such as Media Access Control (MAC) address, Service Set Identifier (SSID), Received Signal Strength Indicator (RSSI), and data frame type.

(3) Data Cleansing and Storage Capability: The extracted information is organized into structured JSON format, with automatically added identification fields. After data cleansing and structuring, the information, along with its metadata, is stored in the Elasticsearch database.

(4) Real-time Data Visualization Capability: Using a visualization module, it generates spectrograms for the corresponding time periods and saves them as image files, providing intuitive data for model analysis.

3) The functional design of the Knowledge Retrieval Sub-Agent, based on DSL rules, is as follows

The Knowledge Retrieval Sub-Agent acts as the data custodian and the bridge between SpecAgent and the Elasticsearch database. Its core function is to translate natural language queries into precise database query statements. Its core capabilities are divided into three areas:

(1) Natural Language Understanding Capability: It receives data requests in natural language from the Master Agent or users. For example, a request might be: “Find all records from the last 10 min for the device with MAC address XX:XX:XX:XX:XX:XX”.

(2) Domain Specific Language (DSL) Query Generation Capability: It utilizes the code generation ability of LLMs to translate natural language requests into Elasticsearch Query DSL statements. To improve the accuracy of generated DSL queries, its prompt is designed with built-in database Schema information and several query examples.

(3) Data Query and Transport Capability: It executes the generated DSL query against the database, retrieves the results from Elasticsearch, and returns them to the previous agent in an easy-to-process format (such as Pandas DataFrame or a JSON list). Alternatively, based on the prompt design, it can forward the results to the next task stage.

4) Functional design of the cognitive reasoning sub-agent based on Bayesian inference

The Cognitive Reasoning Sub-Agent acts as a data analyst in SpecAgent. It is the key component that transforms raw data into actionable information. Unlike traditional rule-based reasoning, SpecAgent equips this sub-agent with Bayesian inference capabilities for quantitative analysis. A probabilistic decision model based on Bayesian inference is embedded as a tool within the sub-agent. This model incorporates uncertainties in single-point RSSI measurements and outputs a probability distribution of the signal source's distance. This provides a quantitative basis for qualitative descriptions like "near" or "far," ultimately forming logically coherent analytical conclusions. Its core capabilities are divided into three areas:

(1) Data Correlation Analysis Capability: It receives data from the Knowledge Retrieval Sub-Agent and processes it based on analysis objectives set by the Master Agent. For example, it can rank a series of RSSI values to identify the device with the strongest signal or analyze trends in RSSI changes of a device over time.

(2) Attribute Discrimination Capability: It performs more complex judgment tasks. For instance, it compares a detected device's MAC address against a known "whitelist" or "blacklist" to determine whether the device is authorized or a potential threat.

(3) Position Reasoning Capability: Signal measurement values (such as RSSI) are subject to interference from various random factors including multipath effects, shadowing, and noise. Decision-making approaches based on deterministic thresholds—for instance, classifying RSSI $\geq -55dB$ as active—exhibit limited robustness in practical applications. Such methods may yield erroneous judgments due to transient signal fluctuations. To enhance the cognitive capabilities of the SpecAgent method in uncertain environments, the study has designed a probabilistic decision model based on Bayesian inference for the Cognitive Reasoning Sub-Agent. The core concept involves transforming the spectrum cognition problem from a deterministic rule-matching task into a statistical decision problem involving probabilistic inference about environmental states under given observational evidence. By integrating physical-layer observation data with higher-level positional reasoning, the study takes the initial determination of signal source location as an example. The study defines the discrete set of positional states for the signal source as $D = \{d_1, d_2, \dots, d_i\}$, where d_i represents a specific positional category. Using the observational evidence E provided by the Signal Perception Sub-Agent, namely the RSSI measurements, the model calculates the posterior probability $P(d_i|RSSI)$ of the signal source being in each possible position state. According to Bayesian inference, the data expression is given by Eq. (1).

$$P(d_i|E) = \frac{P(E|d_i) \cdot P(d_i)}{P(E)} \tag{1}$$

The components of Eq. (1) and their physical and statistical significance are as follows:

$P(d_i|E)$ denotes the posterior probability, representing the probability distribution of determining that the signal source is located at a distance when the observed value is E. This serves as the direct basis for model decision-making. It indicates the likelihood of the signal source being at different distances under the current observation. The Cognitive Reasoning Sub-Agent may take the peak of this probability distribution (Maximum A Posteriori Probability, MAP) for distance estimation and thereby provide a qualitative conclusion.

$P(E|d_i)$ denotes the likelihood function, representing the probability that the RSSI observation value is obtained when the distance d_i to the signal source is known. Due to factors such as multipath effects, obstacles, and dynamic interference, the observed RSSI value fluctuates even at a fixed distance d_i. This function therefore models such variation to describe the uncertainty inherent in wireless channels.

$P(D)$ denotes the prior probability, representing the prior probability over the possible distance states in D to the signal source prior to any RSSI measurements, determined based on prior knowledge of the physical environment.

$P(E)$ denotes the marginal probability, representing the total probability of observing a given RSSI value. In classification decision tasks, it serves as a normalizing constant, calculated via the total probability formula to ensure the sum of posterior probabilities across all location states equals one. The mathematical expression is given in Eq. (2).

$$P(E) = \sum_{j=1}^{k} P(E|d_i) \cdot P(d_i) \tag{2}$$

After calculating the posterior probabilities for all possible position states, the position state with the highest posterior probability is selected as the current optimal cognitive judgment result d. The mathematical expression is given in Eq. (3).

$$\hat{d} = arg \max_{d_i \in D} P(d_i|E) = arg \max_{d_i \in D} P(E|d_i) \cdot P(d_i) \tag{3}$$

During implementation, the Cognitive Reasoning Sub-Agent invokes Bayesian inference modelling tools to compute existing RSSI values and derive posterior probabilities for all potential positional states. Subsequently, the Cognitive Reasoning Sub-Agent employs the maximum a posteriori probability criterion for positional inference. By incorporating this Bayesian inference model, SpecAgent's cognitive reasoning process achieves two enhancements: firstly, the model handles uncertainty in observational data by describing signal characteristics through probability distributions, thereby strengthening SpecAgent's robustness against environmental noise and signal fluctuations; secondly, the model outputs a complete probability distribution vector $[P(d_1|E), P(d_S|E), \dots, P(d_k|E)]$, providing the Cognitive Reasoning Sub-Agent with richer decision-making information.

C. AutoGen-Based Collaborative Workflow and Case Study

SpecAgent's collaborative workflow, built on Microsoft's AutoGen framework, uses GroupChatManager and Conversable Agent classes to create a dynamic, dialogue-driven MAS. The process starts when the Master Agent receives a user task, decomposes it into logical subtasks, and initiates a group chat. It assigns the first subtask to a specialized sub-agent, which reasons about the task, calls an external tool, and posts the result back to the chat. The Master Agent monitors the dialogue, dynamically assigning subsequent subtasks based on progress until all are complete. It then aggregates the results into a final user report. This paradigm provides high flexibility, enabling dynamic planning and real-time interaction to tackle complex cognitive challenges. A schematic of the workflow is shown in Fig. 2. The specific details of each step are as follows. To demonstrate the efficacy of the proposed static architecture and the strict "Chain of Command" protocol, we detail a complete execution cycle for a specific source localization task. In this scenario, the user inputs the instruction: "Locate the distance of the unknown signal detected at 433.92 MHz". The workflow proceeds through four distinct phases, involving all specialized agents, as illustrated in Table I.

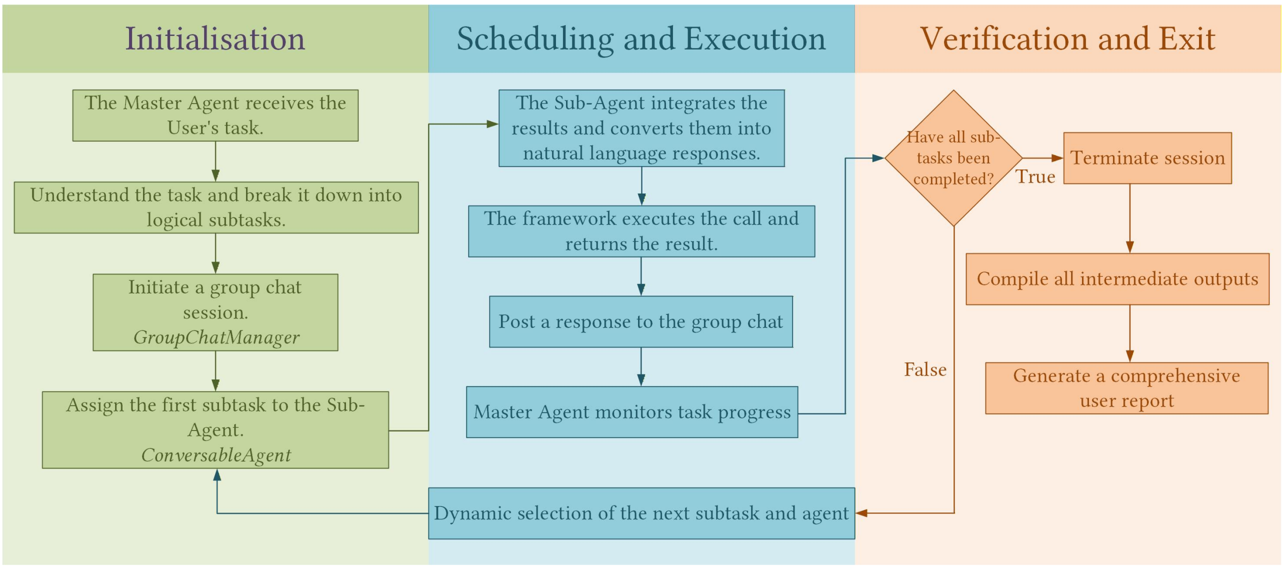


Fig. 2. SpecAgent workflow diagram.

TABLE I. EXECUTION TRACE OF THE LOCALIZATION TASK

Step	Role	Action / Data Payload
1	User	Input: "Locate the distance of the unknown signal detected at 433.92 MHz"
2	Master Agent	Thought: Need RSSI first. Command: {"target": "SignalPerception", "payload": "Measure RSSI at 433.92MHz"}
3	Signal Perception Sub-Agent	Action: Executed Python Script (RTL-SDR). Output: {"RSSI": -72.5, "SNR": 15.0}
4	Master Agent	Command: {"target": "KnowledgeRetrieval", "payload": "Get path loss params for 433MHz"}
5	Knowledge Retrieval Sub-Agent	Output: {"n": 3.0, "A": -40.0, "env": "Indoor"}
6	Master Agent	Command: {"target": "CognitiveReasoning", "payload": "Calculate distance using Bayesian MLE"}
7	Cognitive Reasoning Sub-Agent	Reasoning: Calculated the estimated distance. Output: {"estimated_value": 12.11, "unit": "meters"}
8	Master Agent	Final Response: "Based on the signal strength... the estimated distance is... meters."

IV. EXPERIMENTAL ENVIRONMENT AND SCENARIO DESIGN

To systematically evaluate the feasibility, effectiveness, and intelligence level of SpecAgent, the study designed a series of experiments conducted in a real-world physical environment. This section first introduces the setup of the experimental environment. It then elaborates on the design of three progressive task scenarios. Finally, it defines the key metrics used to assess the system's performance. The details are as follows.

A. Experimental Environment Setup

The Wi-Fi network based on the IEEE 802.11 standard is one of the most widely used wireless communication technologies today. Its signals are present in nearly all human activity spaces, such as offices, homes, and urban areas. This makes the Wi-Fi environment a natural, complex, and dynamic representation of the electromagnetic spectrum. Therefore, the experiments were conducted in a typical indoor office environment. This environment contained multiple Wi-Fi Access Points (APs) operating in both the 2.4 GHz and 5 GHz bands. It also included dozens of client devices, such as smartphones and laptops. Potential co-channel interference sources, like Bluetooth devices and wireless mice, were also present. This indoor office setting effectively simulates a complex and dynamic electromagnetic spectrum scenario, thus meeting the study's experimental requirements. A schematic of the physical hardware deployment is shown in Fig. 3, and the hardware and software stack is detailed in Table II. The performance of LLM-based agents is highly sensitive to the specific phrasing of system prompts. To address the challenge of reproducibility in prompt-dependent systems and to facilitate comparative studies, we have rigorously documented the prompt design. To ensure reproducibility and facilitate further research on prompt sensitivity, the complete set of system prompts for the Master Agent and all Sub-Agents (Signal Perception, Knowledge Retrieval and Cognitive Reasoning) is provided in Supplementary Material. Each prompt includes the specific role definition and constraint instructions. We strictly adhered to these fixed prompts throughout the evaluation to ensure the consistency of the reported success rates and latency metrics.

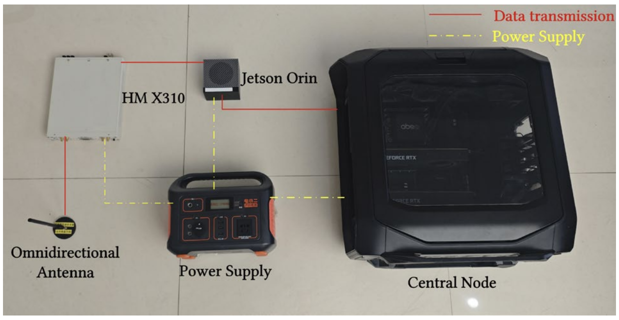


Fig. 3. SpecAgent hardware equipment deployment diagram.

TABLE II. SPECAGENT HARDWARE AND SOFTWARE STACK

Category	Component/Technology	Function and Description
Perception hardware	Omnidirectional antenna	360° reception of wireless signals within specified frequency bands.
Computing Platform	HM X310 USRP	High-performance software-defined radio platform, responsible for RF front-end processing and data digitization.
Computing Platform	Nvidia Jetson AGX Orin	Edge computing node, providing acceleration for LLMs and AI model inference.
Software Framework	Desktop computer equipped with Nvidia RTX 4090	For the initial download, quantisation processing and operation of LLM.
Software Framework	OS: Ubuntu 20.04	Provide a stable operating environment.
Software Framework	LLM: Qwen3	As the backend support for SpecAgent, implementing inference on computing central nodes.
Software Framework	MAS: Microsoft AutoGen	Construction and management of multi-agent dialogue and collaborative workflows.
Software Framework	Signal processing: GNU Radio, Python	For USRP control, data stream processing, and Wi-Fi protocol parsing; GNU Radio version 3.10.8.0.
Software Framework	Database: Elasticsearch 7.9	Storage, indexing, and querying of formatted signal data.
Software Framework	Runtime Env: Python 3.10	Primary development languages.
Software Framework	Edge Model: MiniCPM4/YOLO7	For signal preprocessing and detection of specific targets within spectrograms.

B. Task Scenario Design

1) Task 1: Autonomous spectrum environment scanning

This task validates the core automation capabilities of the SpecAgent method, focusing on the Master Agent's ability to interpret user instruction semantics and schedule tasks, alongside the Signal Perception Sub-Agent's proficiency in executing a full data lifecycle—acquisition, parsing, and persistence—autonomously. Upon receiving a high-level command like "Scan the 2.4 GHz Wi-Fi band and report a summary", SpecAgent initiates an end-to-end workflow: the Master Agent decomposes the instruction and delegates subtasks, prompting the Signal Perception Sub-Agent to acquire signals, parse protocols, and store structured data (e.g., SSID, BSSID, RSSI) into Elasticsearch. The Master Agent then directs the Knowledge Retrieval Sub-Agent to aggregate data, culminating in a formatted environmental summary report, demonstrating the system's capacity to translate high-level user intent into automated technical operations.

2) Task 2: Specific Wi-Fi device identification and fingerprint analysis

This task evaluates SpecAgent's capacity to process multi-step, composite queries by testing the collaborative workflow between its specialized agents. When a user requests analysis of a specific BSSID device—requiring existence confirmation, authorization status check, and signal strength trend examination—the Master Agent decomposes this complex instruction. It then dynamically coordinates the Knowledge Retrieval and Cognitive Reasoning Sub-Agents: the former retrieves device data and historical RSSI sequences, while the latter performs logical judgments and trend analysis. This process validates SpecAgent's ability to manage functionally diverse agents for handling interdependent, multi-dimensional queries, demonstrating effective collaboration for in-depth analysis.

3) Task 3: Preliminary signal source localization based on RSSI

This task tests the Cognitive Reasoning Sub-Agent's core ability to transform raw data (like RSSI measurements) into actionable information, such as estimating a signal's relative distance. When given a query like "identify the strongest signal and estimate its relative distance", the agent first identifies the strongest emitter quantitatively. Crucially, it then uses a Bayesian inference model to convert deterministic RSSI readings into a probabilistic estimate of spatial proximity, applying reasoning based on a defined formula. This demonstrates SpecAgent's key capability to derive high-level understanding from low-level physical measurements.

C. Performance Evaluation Metrics

Given that this study focuses on validating the conceptual framework of the SpecAgent method, this study adopts a composite indicator system that combines traditional performance metrics with system-level evaluations:

(1) Task Completion Success Rate (TCSR): Task Completion Success is defined as the system's ability to generate syntactically correct executable code that successfully invokes SDR hardware APIs and returns a non-null spectral analysis result. This metric assesses the architectural viability (pipeline connectivity) rather than the signal processing optimality compared to ground truth. The mathematical expression is given in Eq. (4).

$$TSCR = Rate_{success} = {}^{Num_{success}}/_{Num_{all}} \qquad (4)$$

(2) End-to-End Task Latency (ETL): This metric measures the total time elapsed from when the user inputs an instruction to when SpecAgent outputs the final report. It reflects the responsiveness of the SpecAgent system. The mathematical expression is given in Eq. (5).

$$ETL = T_{thinking} + T_{Operating} + T_{reasoning} \tag{5}$$

(3) Human Interaction Turns (HIT): This metric counts the number of times a user needs to provide clarification, corrections, or follow-up questions during a single task. An HIT value of 0 indicates full autonomous completion. A higher HIT value indicates poorer autonomy of the SpecAgent system. The mathematical expression is given in Eq. (6).

$$HIT = {}^{Num_{human}}/_{Num_{all}} \tag{6}$$

(4) Proof-of-Concept Evaluation (PCE): This is a qualitative metric. It involves assessing whether the task decomposition, tool invocation, and collaborative logic of the agents align with design expectations and demonstrate intelligence by examining the dialogue logs between the agents.

V. RESULTS

To the best of our knowledge, this work is among the first to explore LLM-based MAS specifically for raw RF signal processing. As this is a nascent intersection of Generative AI and wireless communications, standard benchmarks for 'Generative Spectrum Agents' have not yet been established. While individual sub-modules (e.g., a specific CNN for classification) could be compared with state-of-the-art models, comparing the entire autonomous workflow against manual or script-based methods is challenging due to the lack of standardized control groups. Therefore, our analysis primarily serves as a feasibility study, aiming to define the potential and limitations of this new architecture, rather than claiming superiority over specialized algorithms in narrow metrics.

After 50 rounds of testing, the results were collected from experimental data for the SpecAgent method across the three aforementioned tasks. The results are summarized in Table III and comparison results are summarized in Table IV, and a schematic diagram is shown in Fig. 4. The following section will provide a detailed discussion based on the performance metrics defined in Section III.C. The specific content is as follows.

(1) Automation and Collaboration Capabilities Were Effectively Verified: Task 1 achieved a 100% success rate with zero human interaction turns. This confirms SpecAgent can effectively translate high-level natural language instructions into automated software and hardware operations. The total time consisted of 60 s for signal scanning and on average 142.34 s for agent reasoning, data processing, and report generation. As this experiment requires offline deployment, it is not possible to use online LLM APIs. Moreover, the computational resources available are inherently limited, rendering it incapable of performing high-performance testing.

(2) Multi-Agent Collaboration Was Effectively Verified, but with Some Instability: Task 2 results were central to the experiment. In 96% of cases, the Master Agent successfully coordinated sub-agents for seamless task handover, validating the multi-agent architecture's ability to handle complex, multi-stage tasks. However, these two failures exposed the instability inherent in the system's network architecture: the Master Agent omitted the SSID parameter, resulting in task execution failure. This outcome suggests that SpecAgent has a limited self-correcting capability even without prior training, as it can halt operations and request human input when necessary. Future improvements may be achieved through enhanced prompt engineering or iterative validation mechanisms.

(3) Cognitive Reasoning Ability Was Effectively Verified: Task 3 demonstrated SpecAgent's advantage over traditional scripted tools. Beyond basic RSSI sorting, the Cognitive Reasoning Sub-Agent transformed complex signal data into interpretable, decision-ready information. The 94% success rate confirms the LLM's stable logical reasoning and rule-based judgment, showcasing human-like cognitive capabilities that surpass conventional systems. However, due to prompt engineering considerations, three instances of inaccurate estimation occurred. Analysis indicates that the Knowledge Retrieval Sub-Agent failed to limit the number of queries when generating DSL query statements, resulting in an inability to accurately estimate positional results during the initial experimental phase. We shall address this through subsequent refinements to the prompt engineering process.

(4) Latency Analysis: Analysis showed 70–80% of time was consumed by Qwen3 model inference (e.g., dialogue, reasoning, code generation). In contrast, signal processing and database queries completed in seconds. The LLM-driven approach introduces a reasonable time overhead (tens of seconds) in exchange for high automation, flexibility, and intelligence, effectively reducing human effort and costs for non-hard-real-time tasks like spectrum monitoring. Given the limited computational resources available for this experiment, it was not feasible to test larger-scale LLMs. However, the observed latency (e.g., 414 s in Task 2) and token consumption are attributed to the fully connected communication topology employed in this prototype. As analyzed in the context of AgentDropout, a significant portion of inter-agent interactions in static architectures may be redundant. The current latency values serve as a baseline for unoptimized LLM-MAS implementations in the RF domain. Future implementations can effectively mitigate this by applying redundancy-pruning algorithms analogous to AgentDropout.

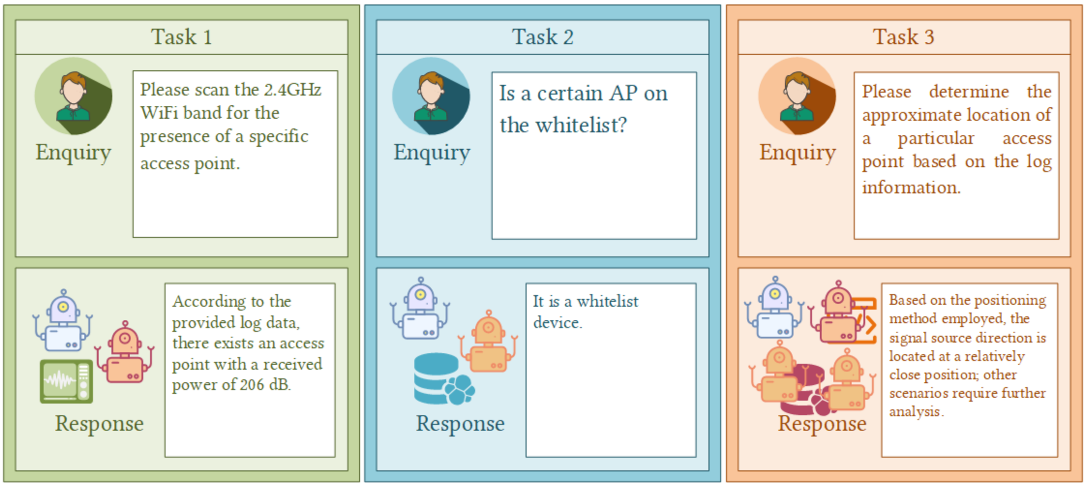
TABLE III. SPECAGENT PERFORMANCE EVALUATION RESULTS

Task	TCSR	ETL (Offline)	HIT	PCE	Analysis of Causes
1	100% (50/50)	142.34″	0	Highly compliant, task decomposition is clear, tool invocation is precise.	The extended ETL duration is attributable to the inclusion of a 60 s spectrum scan and perception time. As this experiment is conducted offline, the scanning duration is also incorporated into the statistical scope.
2	96% (48/50)	414.18″	0.04	Generally compliant, in 48 successful cases, agent collaboration proceeded smoothly.	In 2 failed cases, due to fluctuations in network transmission, the primary agent failed to correctly pass the SSID parameter to the inference agent, resulting in 1 and 2 interactions respectively during the experiment.
3	94% (47/50)	45.4″	0	Highly compliant, successful data conversion.	The three experiments failed to yield conclusive results due to the prompt design, as the Knowledge Retrieval Sub-Agent was unable to effectively generate corresponding DSL query statements, thereby preventing SpecAgent from delivering accurate conclusions.

TABLE IV. COMPARISON RESULTS

Project	Classification	Competency Assessment				
		Comprehension	Spectrum Awareness	Protocol Analysis (802.11)	Distance Estimation	Automation
Qwen3	LLM	√	×	×	×	×
MiniCPM	LLM	√	×	×	×	×
DLSpectSenNet [40]	DL	×	√	×	×	×
DLM [41]	DL	×	√	×	×	×
Sparrow-WiFi*	Programming	×	√	√	×	×
Aircrack-ng*	Programming	×	×	√	×	×
LSTM-CNN [24]	DL	×	√	×	√	×
RSSI	Programming	×	√	×	√	×
CompileAgent	MAS	√	×	×	×	√
SpecAgent	MAS	√	√	√	√	√

Note: Sparrow-WiFi*: https://github.com/ghostop14/sparrow-wifi.git
Aircrack-ng*: https://www.aircrack-ng.org/



Please scan the 2.4GHz WiFi band for the presence of a specific access point.

According to the provided log data, there exists an access point with a received power of 206 dB.

Is a certain AP on the whitelist?

It is a whitelist device.

Please determine the approximate location of a particular access point based on the log information.

Based on the positioning method employed, the signal source direction is located at a relatively close position; other scenarios require further analysis.

Fig. 4. SpecAgent schematic diagram of operational results.

In a typical Wi-Fi signal analysis scenario, the study systematically validated the feasibility and superiority of SpecAgent. The experimental results show that SpecAgent can not only accurately and efficiently complete the preset tasks but also demonstrates significant advantages over traditional signal processing workflows in three key aspects:

(1) Automation and Flexibility: It can understand the macro-intent of human experts (conveyed via natural language instructions) and translate it into machine-executable, multi-step operation sequences. This lowers the barrier to operation and improves task execution efficiency.

(2) Human-Computer Interaction: The dialogue-based interaction paradigm makes operating SpecAgent similar to communicating with a specialist assistant. Users can easily issue instructions, ask for details, and adjust tasks, achieving human-machine collaboration.

(3) Extensibility: By adding new agent roles or equipping existing agents with new tool-calling functions, SpecAgent’s capabilities can be conveniently expanded to handle more diverse signal types and analysis tasks.

In summary, this experiment successfully provided a proof-of-concept for SpecAgent’s design. It can be effectively applied in the field of communication signal analysis, transforming complex human expert knowledge and task flows into intelligent behaviors that machines can autonomously understand and execute. This work provides a solid technical prototype and practical basis for realizing next-generation autonomous spectrum cognition systems.

VI. DISCUSSION

A. *Scalability and Architectural Adaptation*

The current implementation of SpecAgent utilizes a predefined set of agents, which ensures stability but limits scalability when facing novel spectral tasks outside the designated capabilities. In the broader field of general-purpose MAS, Tian *et al.* [20] proposed AgentInit, a framework that employs an optimization-based approach to dynamically generate agent roles and execution plans based on specific problem contexts. The contrast between AgentInit's dynamic orchestration and SpecAgent's static topology highlights a critical trade-off: dynamic initialization offers superior adaptability for unstructured tasks, whereas static assignment provides higher control predictability for hardware-in-the-loop systems. Future iterations of SpecAgent should incorporate mechanisms similar to AgentInit to autonomously instantiate new agent roles (e.g., a specific protocol decoder) when the Planner Agent detects a capability gap, thereby moving from a fixed-role baseline to an adaptive ecosystem.

In addition, scalability-oriented orchestration methods such as DRTAG [39] may offer further guidance for enabling elastic agent scaling in larger and more dynamic spectrum cognition scenarios.

B. *Latency Analysis and Efficiency Optimization*

The experimental results indicate substantial latency (e.g., approximately 150 s for Task 1) and high token consumption, which presents a challenge for real-time deployment on edge devices. This latency is primarily attributed to the unoptimized redundancy in inter-agent communication and the sequential processing of large language models. Recent research by Wang *et al.* [21] on AgentDropout has demonstrated that a significant portion of agent interactions in static MAS structures can be redundant. By training a utility model to prune non-essential communication nodes, AgentDropout effectively reduces token costs and inference time without compromising task accuracy. Our analysis suggests that the high latency reported in this study is a consequence of the fully connected interaction graph where agents may exchange verbose confirmations. The integration of a redundancy-elimination mechanism, analogous to AgentDropout, provides a plausible pathway for optimizing SpecAgent. By selectively activating agents only when their specific expertise is required by the current reasoning step, the computational overhead on the Jetson Orin platform can be significantly mitigated.

C. *Interpretability versus Quantitative Precision*

Regarding the comparison with traditional methods (e.g., LSTM-CNN [23] or RSSI-based localization), it is essential to distinguish the distinct objectives of these paradigms. Deep learning models excel in quantitative precision (e.g., minimizing RMSE in localization) but operate as "black boxes" lacking explanatory power. In contrast, SpecAgent prioritizes interpretability and process automation. The "Task Completion Success" reported in this study reflects the system's ability to successfully translate natural language into executable hardware code, a capability absent in traditional signal processing pipelines. While SpecAgent currently entails higher latency and lower estimation precision compared to specialized supervised models, it provides a unique semantic interface that enables human operators to interact with RF systems using high-level logic. Therefore, SpecAgent should be viewed as a complementary decision-making layer rather than a direct replacement for low-level signal estimators.

VII. CONCLUSION

Facing the increasingly complex electromagnetic environment and the pressing need for autonomous and intelligent spectrum cognition, the study proposed and successfully implemented a multi-agent collaborative spectrum cognition method named SpecAgent. SpecAgent demonstrates the feasibility of utilizing LLM-based multi-agent systems for spectrum cognition, which can autonomously understand tasks, collaboratively execute them, and interact with the physical world in real-time. Although the SpecAgent currently exhibits higher latency compared to dedicated neural networks, the proposed architecture provides a unique semantic interface for RF systems. Future work will prioritize the integration of dynamic agent initialization and redundancy elimination mechanisms, such as AgentInit and AgentDropout, to address the identified efficiency bottlenecks. This work preliminarily addresses key issues in existing spectrum cognition approaches, such as fragile adaptability to dynamic environments, ineffective transformation of raw data into actionable information, and bottlenecks in human-computer interaction efficiency. The study shows the SpecAgent represents only a starting point. As LLM technology continues to evolve and deeply integrate with the communications field, a network composed of intelligent agents is poised to become essential infrastructure for future intelligent communications. However, as exploratory research, SpecAgent still has limitations, which also indicate directions for future work. Firstly, the latency of reasoning poses a challenge for real-time spectrum cognition tasks requiring microsecond-level responses, such as channel monitoring and tracking frequency-hopping signals. Secondly, the hallucination problem of LLMs requires attention: LLMs might generate erroneous code or make incorrect logical inferences when instructions are ambiguous, necessitating the design of robust error-checking and correction mechanisms. Future work will focus on addressing these aspects and conducting large-scale, multi-scenario testing.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

L.L. and N.H. primarily conducted the experiments and drafted the manuscript. W.Z. and D.Z. contributed to the preparation of antenna and equipment setup, construction of the experimental environment, and analysis of the experimental data. J.W. and J.L. revised and proofread the manuscript. All authors approved the final version of the manuscript.

FUNDING

This research was supported financially by the National Natural Science Foundation of China (62171468).

REFERENCES

[1] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021. doi: 10.1109/OJCOMS.2021.3057679

[2] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, "Autonomous chemical research with large language models," *Nature*, vol. 624, no. 7992, pp. 570–578, Dec. 2023. doi: 10.1038/s41586-023-06792-0

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surv. Tutor.*, vol. 11, no. 1, pp. 116–130, 2009. doi: 10.1109/SURV.2009.090109

[4] E. Axell, G. Leus, and E. G. Larsson, "Overview of spectrum sensing for cognitive radio," in *Proc. 2010 2nd International Workshop on Cognitive Information Processing*, Elba Island, Italy, IEEE, Jun. 2010, pp. 322–327. doi: 10.1109/CIP.2010.5604136

[5] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Nov. 2017. doi: 10.1109/CC.2017.8233654

[6] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, C. Jayne and L. Iliadis, Eds. Cham: Springer International Publishing, 2016, pp. 213–226.

[7] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, "Deep learning for RF signal classification in unknown and dynamic spectrum environments," in *Proc. 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Newark, NJ, USA, IEEE, Nov. 2019, pp. 1–10. doi: 10.1109/DySPAN.2019.8935684

[8] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018. doi: 10.1109/TCCN.2018.2835460

[9] S. Kokal, M. Vanamala, and R. Dave, "Deep learning and machine learning, better together than apart: A review on biometrics mobile authentication," *J. Cybersecurity Priv.*, vol. 3, no. 2, pp. 227–258, Jun. 2023. doi: 10.3390/jcp3020013

[10] D. R. Reising, J. H. Tyler, M. K. M. Fadul, M. R. Hilling, and T. D. Loveless, "Improved RF fingerprint-based identity verification in the presence of an SEI mimicking adversary," *J. Cyber Secur. Mobil.*, pp. 887–916, Sep. 2024. doi: 10.13052/jcsm2245-1439.1354

[11] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Med.*, vol. 15, no. 11, e1002683, Nov. 2018. doi: 10.1371/journal.pmed.1002683

[12] Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," in *Proc. ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

[13] Y. Zhou, L. Song, B. Wang, and W. Chen, "MetaGPT: Merging large language models using model exclusive task arithmetic," in *Proc. the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1711–1724. doi: 10.18653/v1/2024.emnlp-main.102

[14] Y. Yang, Z. Li, Q. Dong, H. Xia, and Z. Sui, "Can large multimodal models uncover deep semantics behind images?" in *Proc. Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, 2024, pp. 1898–1912. doi: 10.18653/v1/2024.findings-acl.113

[15] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, and K. Narasimhan, "React: Synergizing reasoning and acting in language models," in *Proc. 11th International Conference on Learning Representations, ICLR 2023*, Kigali, Rwanda, 2023.

[16] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proc. the 36th Annual ACM Symposium on User Interface Software and Technology*, San Francisco CA USA, ACM, Oct. 2023, pp. 1–22. doi: 10.1145/3586183.3606763

[17] C. Qian *et al.*, "ChatDev: Communicative agents for software development," in *Proc. the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, Association for Computational Linguistics, 2024, pp. 15174–15186. doi: 10.18653/v1/2024.acl-long.810

[18] A. Yang *et al.*, "Qwen3 technical report," arXiv preprint, arXiv:2505.09388, 2025.

[19] M. Team *et al.*, "MiniCPM4: Ultra-efficient LLMs on end devices," arXiv preprint, arXiv:2506.07900, 2025.

[20] C. Tian *et al.*, "AgentInit: Initializing LLM-based multi-agent systems via diversity and expertise orchestration for effective and efficient collaboration," arXiv preprint, arXiv:2509.19236, 2025.

[21] Z. Wang *et al.*, "AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration," in *Proc. the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria, Association for Computational Linguistics, Jul. 2025, pp. 24013–24035. doi: 10.18653/v1/2025.acl-long.1170

[22] Q. Tian, R. Cai, Y. Luo, and G. Qiu, "DOA estimation: LSTM and CNN learning algorithms," *Circuits Syst. Signal Process.*, vol. 44, no. 1, pp. 652–669, Jan. 2025. doi: 10.1007/s00034-024-02866-0

[23] X. Fang, M. Jin, Q. Guo, and T. Jiang, "CNN-transformer-based cooperative spectrum sensing in cognitive radio networks," *IEEE Wirel. Commun. Lett.*, vol. 14, no. 5, pp. 1576–1580, May 2025. doi: 10.1109/LWC.2025.3550536

[24] Y. Chen, Y. Wang, S. Peng, J. Li, and R. Xiao, "Deep denoising and transformer-based 3D spectrum sensing," *IEEE Commun. Lett.*, vol. 29, no. 12, pp. 2894–2898, Dec. 2025. doi: 10.1109/LCOMM.2025.3618686

[25] J. Jang, J. Pyo, Y.-I. Yoon, and J. Choi, "Meta-transformer: A meta-learning framework for scalable automatic modulation classification," *IEEE Access*, vol. 12, pp. 9267–9276, 2024. doi: 10.1109/ACCESS.2024.3352634

[26] L. Zhai, Y. Li, Z. Feng, S. Yang, and H. Tan, "Learning cross-domain features with dual-path signal transformer," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 2, pp. 3863–3869, Feb. 2025. doi: 10.1109/TNNLS.2024.3350609

[27] S. Long *et al.*, "6G comprehensive intelligence: Network operations and optimization based on large language models," *IEEE Netw.*, vol. 39, no. 4, pp. 192–201, Jul. 2025. doi: 10.1109/MNET.2024.3470774

[28] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Commun. Surv. Tutor.*, vol. 27, no. 6, pp. 3820–3860, Dec. 2025. doi: 10.1109/COMST.2025.3527641

[29] K. Qiu, S. Bakirtzis, I. Wassell, H. Song, J. Zhang, and K. Wang, "Large language model-based wireless network design," *IEEE Wirel. Commun. Lett.*, vol. 13, no. 12, pp. 3340–3344, Dec. 2024. doi: 10.1109/LWC.2024.3462556

[30] X. He *et al.*, "Large language model offloading using active inference in 6G symbiotic IoT," *IEEE Internet Things J.*, 2025. doi: 10.1109/JIOT.2025.3588832

[31] T. Chen, S. Zheng, K. Qiu, L. Zhang, Q. Xuan, and X. Yang, "Augmenting radio signals with wavelet transform for deep learning-based modulation recognition," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 6, pp. 2029–2044, Dec. 2024. doi: 10.1109/TCCN.2024.3400525

[32] J. Han, Z. Yu, and J. Yang, "A lightweight deep learning architecture for automatic modulation classification of wireless internet of things," *IET Commun.*, vol. 18, no. 18, pp. 1220–1230, Nov. 2024. doi: 10.1049/cmu2.12823

[33] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019. doi: 10.1109/JPROC.2019.2918951

[34] E. C. Strinati *et al*., "6G: The next frontier: from holographic messaging to artificial intelligence using sub terahertz and visible light communication," *IEEE Veh. Technol. Mag*., vol. 14, no. 3, pp. 42–50, 2019. doi: 10.1109/MVT.2019.2921162

[35] G. Sun *et al.*, "Toward AI-native task orchestration for collaborative computing in SAGSINs," *IEEE Commun. Mag.*, vol. 63, no. 12, pp. 112–118, Dec. 2025. doi: 10.1109/MCOM.004.2400304

[36] C.-A. Fu, C.-J. Peng, H.-Y. Huang, W.-H. Cheng, I.-B. Liao, and Y.-H. Li, "AI agents for autonomous driving: An overview," *IEEE Consum. Electron. Mag.*, pp. 1–17, 2025. doi: 10.1109/MCE.2025.3636090

[37] S. Wang, N. Niu, Z. Wang, Y. Lv, and J. Zhang, "Heterogeneous multi-agent task planning method in complex marine environment," *IEEE Access*, vol. 13, pp. 84202–84216, 2025. doi: 10.1109/ACCESS.2025.3567503

[38] Q. Liang, J. Liu, Z. Jiang, J. Yin, K. Xu, and H. Li, "Limited information aggregation for collaborative driving in multi-agent autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6624–6631, Jul. 2024. doi: 10.1109/LRA.2024.3410159

[39] R. Perera, A. Basnayake, and M. Wickramasinghe, "Auto-scaling LLM-based multi-agent systems through dynamic integration of agents," *Front. Artif. Intell.*, vol. 8, 2025. doi: 10.3389/frai.2025.1638227

[40] V. Roopa and H. Shekhar Pradhan, "Deep learning based intelligent spectrum sensing in cognitive radio networks," *IETE J. Res.*, vol. 70, no. 12, pp. 8425–8445, Dec. 2024. doi: 10.1080/03772063.2024.2386599

[41] A. Kumar, N. Gaur, S. Chakravarty, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "Analysis of spectrum sensing using deep learning algorithms: CNNs and RNNs," *Ain Shams Eng. J.*, vol. 15, no. 3, 102505, Mar. 2024. doi: 10.1016/j.asej.2023.102505