SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2026, Vol. 69, Iss. 8, 180302:1–180302:17
https://doi.org/10.1007/s11432-026-5026-5

Special Topic: Large Model-Driven Aerospace Microwave Imaging

RS-Agent: automating remote sensing tasks through intelligent agent

Wenjia XU[1*†], Zijian YU[1†], Boyang MU[1], Jiuniu WANG[2], Zhiwei WEI[3*] & Mugen PENG[1]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Department of Computer Science, City University of Hong Kong, Hong Kong 999077, China*
[3]*School of Geographic Sciences, Hunan Normal University, Changsha 410081, China*

Received 13 January 2026/Revised 7 May 2026/Accepted 30 June 2026/Published online 27 July 2026

Abstract The unprecedented advancements in multimodal large language models (MLLMs) have demonstrated strong potential for language-vision interaction in remote sensing tasks such as visual question answering and scene understanding. However, these models are largely constrained to basic instruction-following or descriptive tasks, and struggle with real-world remote sensing scenarios where multi-source data, fine-grained spatial semantics, and expert knowledge are indispensable. To address these limitations, we propose RS-Agent, a domain-adapted intelligent agent that bridges user intent and professional remote sensing workflows through structured task planning and tool orchestration. RS-Agent is built upon four key components that explicitly follow the typical workflow of remote sensing applications: an LLM-based Central Controller for user intent understanding and analytical process planning, a dynamic toolkit for tool execution, a Solution Space for task-specific expert guidance, and a Knowledge Space for domain-level knowledge support. To further enhance the performance of RS-Agent, we introduce two novel mechanisms: Task-Aware Retrieval, which improves task planning by explicitly inferring task types and retrieving expert-defined procedural solutions, rather than relying on query-level similarity or ad-hoc tool chaining; and DualRAG, a weighted dual-path retrieval-augmented generation method, which enhances the relevance and completeness of retrieved domain knowledge. RS-Agent natively supports multiple imaging modalities, including optical and SAR imagery. For SAR tasks in particular, the agent plans and orchestrates dedicated SAR processing and analysis tools into executable workflows, improving reliability and automation under underspecified requests. Extensive experiments across 9 datasets and 18 remote sensing tasks demonstrate that RS-Agent significantly outperforms state-of-the-art MLLMs, achieving over 95% task planning accuracy and delivering superior performance in tasks such as scene classification, object counting, and remote sensing visual question answering. These results validate the effectiveness of a dedicated remote sensing agent that fuses LLM reasoning with domain expertise for geospatial intelligence. Our work presents RS-Agent as a robust and extensible framework for advancing intelligent automation in remote sensing analysis. Our code will be available at https://github.com/IntelliSensing/RS-Agent.

Keywords remote sensing, large language model, AI agent, retrieval-augmented generation

Citation Xu W J, Yu Z J, Mu B Y, et al. RS-Agent: automating remote sensing tasks through intelligent agent. Sci China Inf Sci, 2026, 69(8): 180302, https://doi.org/10.1007/s11432-026-5026-5

1 Introduction

By aligning visual and textual information, multi-modal large language models (MLLMs) [1,2] have advanced remote sensing image interpretation, including captioning [3], visual question answering [4], and scene understanding [5,6], making remote sensing data more accessible to non-expert users.

However, remote sensing presents unique challenges. Its multi-modal, multi-temporal, and multi-resolution characteristics—spanning optical, SAR, and multi-spectral imagery—demand expert-level interpretation that MLLMs struggle to achieve. Moreover, most existing MLLMs are designed for instruction-following or descriptive tasks [7–10], and thus fail to deliver the analytical reasoning required for geospatial interpretation. These limitations call for domain-aware intelligent systems that understand user intent and autonomously orchestrate specialized models.

AI Agents [11], autonomous systems that interpret user intent, plan multi-step tasks, invoke tools, and refine decisions from intermediate outcomes [12–14], offer a promising pathway for remote sensing intelligence. Recent advancements in agent frameworks—often powered by large language models—have demonstrated strong potential

* Corresponding author (email: xuwenjia@bupt.edu.cn, trentonwei@whu.edu.cn)
† These authors contributed equally to this work.

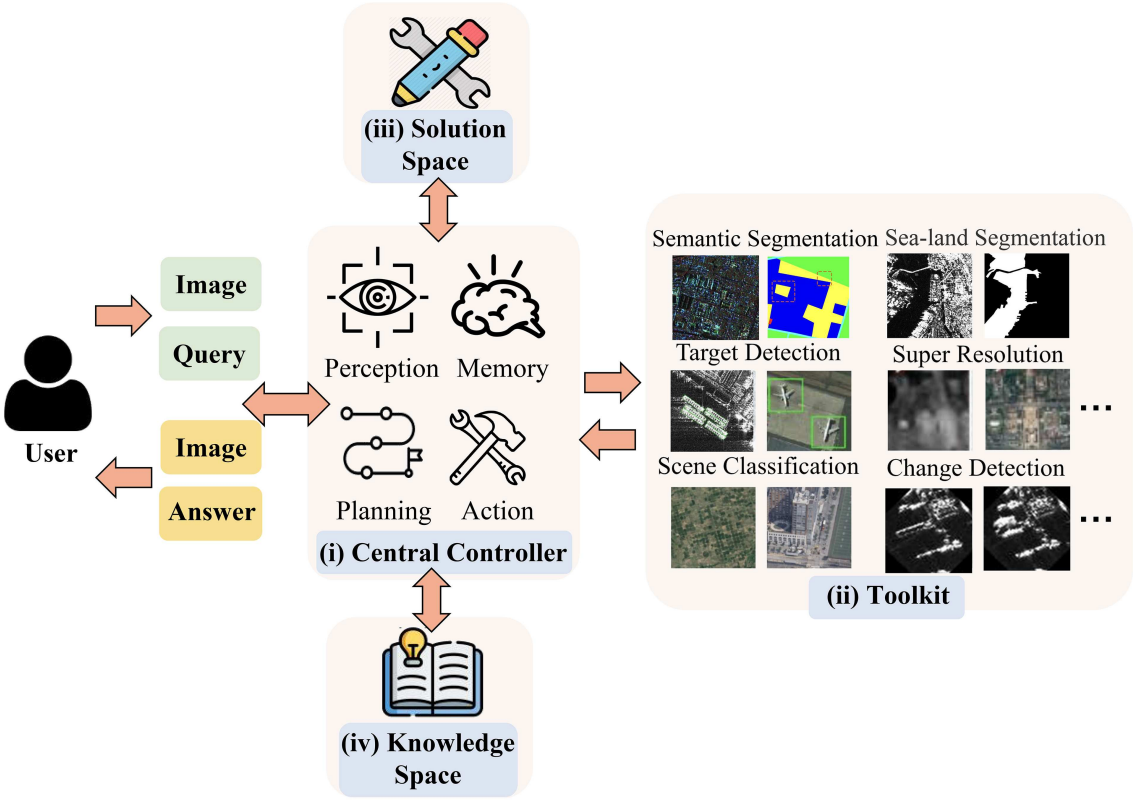


Figure 1 (Color online) The schematic diagram of the RS-Agent. RS-Agent consists of four main components: a Central Controller based on LLM, which comprehends user queries, maintains historical context, plans tool usage, and aggregates results; a Solution Space that provides well-defined solutions to user-submitted problems; a Knowledge Space offering domain-specific knowledge in remote sensing; and a Toolkit integrating state-of-the-art methods for remote sensing tasks.

in coordinating multiple tools, decomposing complex tasks, and flexibly adapting to diverse user goals through modular design [11, 15], with successful adoption in domains such as next-generation wireless networks [16, 17].

Although a few studies in the remote sensing field use the term "agent" in their paper [18–20], most of them are designed for narrowly scoped tasks and lack a unified architecture that can generalize across heterogeneous remote sensing applications. A representative example is RS-ChatGPT [21], which integrates ChatGPT [22] with a collection of pre-trained remote sensing networks to perform tasks such as object detection, scene classification, and image captioning. However, RS-ChatGPT encounters difficulties when scaling to more diverse tools or addressing problems that require structured procedural reasoning and specialized domain knowledge, revealing the absence of a unified, domain-aware agent framework for real-world remote sensing analysis.

In this paper, we propose RS-Agent, a domain-specific AI agent designed to bridge this gap by integrating LLM reasoning with expert knowledge for professional remote sensing applications. As shown in Figure 1, RS-Agent comprises four components that mirror the typical remote sensing workflow: (i) an LLM-based Central Controller that interprets queries, plans tasks, executes tools, and aggregates results; (ii) a Toolkit of SOTA methods covering tasks such as denoising, super-resolution, detection, captioning, and scene classification; (iii) a Solution Space that provides task-specific solutions to guide tool selection; (iv) a Knowledge Space that supplies domain expertise for remote sensing tasks.

The core functionality of RS-Agent lies in interpreting user instructions and converting them into executable workflows. To this end, we design a structured agent architecture that combines modular prompting with dynamic context tracking, enabling the Central Controller to plan workflows adaptively from current inputs, tool outputs, and contextual memory [23, 24]. In the Solution Space, to enable RS-Agent to reason like a remote sensing analyst, we build a Solution Database storing professional knowledge for tool use, and propose Task-Aware Retrieval, which infers task types from user queries and retrieves expert-defined solution procedures for accurate tool selection and step-by-step decomposition. To address the domain knowledge limitations of general-purpose models [22, 25, 26], we develop a dedicated Knowledge Database, tailored to the remote sensing domain. Building on this, we propose DualRAG, which combines weighted keyword-aware retrieval with global semantic search to enable the agent to retrieve more relevant and accurate domain-specific knowledge.

We conduct extensive experiments across 9 datasets and 18 remote sensing tasks. RS-Agent achieves over 95% task planning accuracy, substantially outperforming the SOTA baseline RS-ChatGPT [21], and consistently surpasses single-model performance on scene classification, RS visual question answering, and object counting. The proposed DualRAG further improves domain knowledge retrieval. RS-Agent is LLM-agnostic; we validate its generality with ChatGPT [22], LLaMA [25], Qwen [27], and DeepSeek [28].

The contributions of this paper are summarized as follows.

• **RS-Agent: a comprehensive agent framework for remote sensing.** We present RS-Agent, an architecture designed for remote sensing scenarios that interprets user queries and orchestrates diverse tools for accurate task execution. Its four components—Central Controller, Toolkit, Solution Space, and Knowledge Space—work in concert to enable robust performance across applications.

• **Domain-adapted task planning Solution Space with Task-Aware Retrieval method.** To enhance the agent's task planning accuracy, we propose a domain-adapted Task-Aware Retrieval method. It exploits the structured nature of remote sensing workflows by first inferring a task type and then retrieving expert-curated solution templates as a planning prior. Leveraging domain-specific task knowledge, RS-Agent emulates the decision-making and tool selection processes of professional remote sensing analysts for complex and multi-step tasks.

• **Knowledge Space with DualRAG to retrieve professional domain knowledge.** To strengthen RS-Agent's remote sensing domain knowledge, we propose DualRAG, a domain-adapted retrieval augmented generation method that assigns weights to extracted keywords and performs dual path retrieval, thereby enhancing the accuracy and relevance of knowledge retrieval for geospatial analysis tasks. Its dual-path design addresses the heterogeneous nature of remote sensing queries that require both macro-level context and fine-grained technical details.

• **Superior performance on various applications.** Extensive experiments demonstrate that RS-Agent consistently surpasses prior SOTA MLLMs across remote sensing applications and substantially boosts task planning accuracy, marking a step forward in adapting AI agents to remote sensing.

2 Related work

2.1 Remote sensing multimodal large language models

MLLMs have transformed human-AI interaction in earth observation by aligning visual and semantic features for natural-language interpretation of remote sensing imagery. Early work focused on captioning [29,30] and VQA [31], but lacked flexibility for open-ended queries, motivating general-purpose models. RSGPT [5] and GeoChat [6] adapted LLaVA to remote sensing, supporting multiturn dialogue and visual geo-localization, while LHRS-Bot [32] introduced a multi-level vision-language alignment for SOTA perception. EarthVQANet [33] and SkySenseGPT [34] added multi-task learning and knowledge-graph integration to enhance reasoning. For temporal dynamics, ChangeCLIP [35] and SkyEyeGPT [36] extended to change captioning and video understanding, and UniRS [37] unified single-frame, bi-temporal, and video inputs.

However, the end-to-end paradigm has intrinsic limitations: current MLLMs struggle with deep reasoning, often hallucinate on quantitative tasks such as object counting, and rely on static knowledge that requires prohibitive retraining to update [6]. RS-Agent addresses these issues by shifting to agentic orchestration—leveraging LLM planning to invoke specialized tools and retrieve dynamic knowledge without retraining.

2.2 LLM-based agents in remote sensing

LLM-based agents extend static MLLMs with autonomous planning and tool orchestration, perceiving environments and refining decisions via feedback loops [13, 38]. In remote sensing, such agents have evolved from single-task specialists to general-purpose frameworks. Early studies targeted vertical applications: TreeGPT [18] tailors an expert system for forestry with tree segmentation and ecological parameter estimation, while Change-Agent [19] enables interactive change detection and captioning. Moving toward broader applicability, RS-ChatGPT [21] pioneered tool chaining by linking ChatGPT with discriminative models for tasks like detection and classification. More recently, ThinkGeo [39] and EarthAgent [40] established comprehensive benchmarks for assessing agentic capabilities.

However, a structural gap remains: most baselines rely on generic agent paradigms (e.g., standard ReAct loops) that lack cognitive structures for remote sensing's procedural complexity, and depend on the LLM's static parametric memory rather than dynamic professional knowledge. RS-Agent bridges these gaps with a Solution Space for expert-level procedural guidance and a DualRAG mechanism for structured knowledge retrieval.

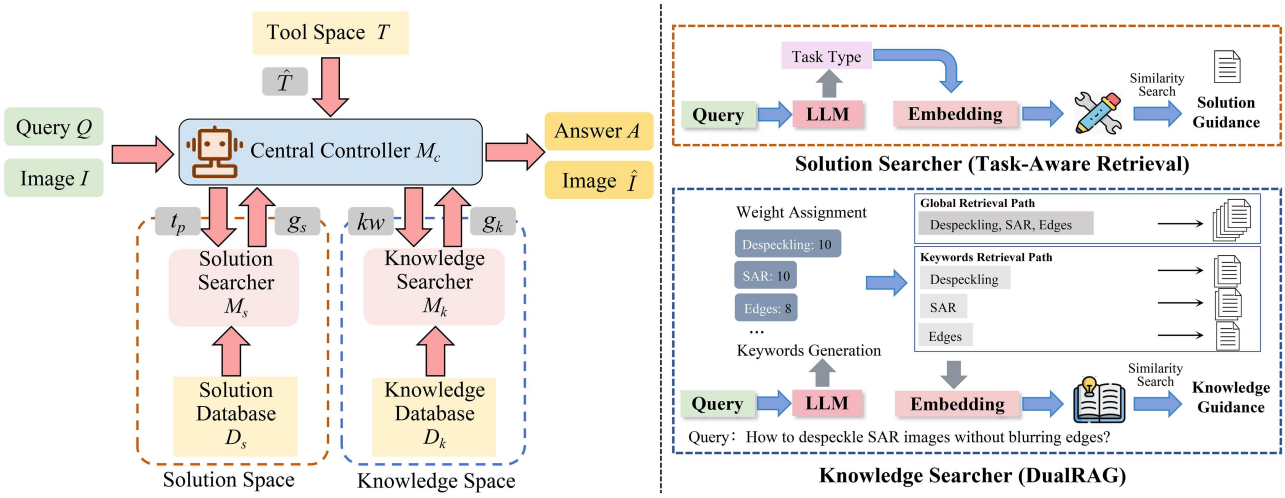


Figure 2 (Color online) Framework of RS-Agent. The left side shows the overall architecture, including key components like the Central Controller, Solution Space, Knowledge Space, and Toolkit. The right side details the Solution Searcher M_s and Knowledge Searcher M_k. The M_s uses the Task-Aware Retrieval method for solution guidance g_s generation. The M_k adopts a DualRAG strategy to retrieve relevant documents.

2.3 Retrieval-augmented generation

Retrieval-augmented generation (RAG) addresses LLMs' lack of domain-specific training data [41]. Standard dense or sparse retrieval [42–44] works for general queries but falters on multi-hop or global-aggregation tasks [45]. Graph-RAG approaches [46,47] integrate entity-relation structures; notably, LightRAG [48] combines graph indexing with keyword-guided queries, but builds a single retrieval vector by concatenating all keywords with equal weight. In remote sensing, where queries mix critical technical specifications (e.g., sensor types or resolutions) with generic terms, this uniform concatenation dilutes key constraints. We therefore propose DualRAG, which combines holistic query-based and decomposed keyword-based retrieval with a dynamic weight allocation mechanism.

3 Remote sensing agent (RS-Agent)

In this section, we formally define the problem addressed by RS-Agent and detail its architectural design. We then elaborate on the four core components that enable robust geospatial reasoning.

3.1 Overall architecture

3.1.1 *Problem formulation*

RS-Agent serves as an intelligent interface that maps a user query Q and a remote sensing image I to a textual answer A and, optionally, a processed visual output $\hat{I}$. Unlike general vision-language tasks, remote sensing queries are often under-specified (e.g., "Assess the damage" implies change detection, area calculation, and severity grading) and domain-dependent (requiring sensor knowledge), so standard MLLMs $f(Q, I)$ fail to capture the implicit procedural requirements. RS-Agent therefore formulates the problem as multi-stage reasoning, introducing three latent variables: task type t_p (analytical objective), solution guidance g_s (expert procedural workflows), and domain knowledge g_k (geospatial context and specifications), which together bridge user intent and tool execution.

3.1.2 *Architectural design*

To address procedural ambiguity and knowledge sparsity in remote sensing queries, RS-Agent adopts a task-centric architecture (Figure 2). Rather than mapping Q directly to tool actions, it first abstracts user intent into explicit task types; this Task Inference step decouples intent understanding from execution, enabling retrieval of expert-verified workflows from the Solution Space and geospatial context from the Knowledge Space before tool invocation. RS-Agent comprises four core components.

(1) **Central Controller (M_c)**: Serves as the decision-making core of the agent, built upon a large language model. It is responsible for four key functions. (i) **Perception**: Analyzing user instruction Q and image metadata I; (ii) **Planning**: Consulting the Solution Space to decompose abstract queries into executable workflows;

(iii) **Retrieval**: Querying the Knowledge Space for domain support when parameter or context ambiguity arises; and (iv) **Action**: Dynamically orchestrating tools and synthesizing the final response A based on execution results.

(2) **Toolkit (T)**: A modular and extensible collection of state-of-the-art remote sensing tools covering diverse modalities (e.g., optical imagery, SAR) and tasks spanning multiple analysis levels (e.g., low-level processing tasks such as denoising and super-resolution, and high-level semantic tasks such as detection, segmentation, classification, and change analysis). Unlike end-to-end MLLMs where tool capabilities are frozen in model weights, the Toolkit design is modular and extensible, supporting plug-and-play integration. It allows for the seamless integration of new SOTA models or specialized tools without requiring the retraining of the Central Controller, thereby ensuring the agent remains at the forefront of algorithmic performance.

(3) **Solution Space (D_s)**: A repository of expert-verified standard operating procedures addressing the procedural ambiguity of remote sensing tasks. A **Solution Searcher** (M_s) with a **Task-Aware Retrieval** mechanism conditions retrieval on the inferred task type (t_p) rather than the raw query, fetching solution guidance (g_s) that encodes professional tool execution sequences and ensures rigorous workflows over zero-shot planning.

(4) **Knowledge Space**: Complementing the Solution Space, the Knowledge Space stores structured remote sensing knowledge, such as definitions, operational principles, or region-specific metadata. During task planning, the Controller queries this space to resolve ambiguities or retrieve explanatory information. To ensure relevant knowledge access, this module features a **Knowledge Searcher** (M_k) driven by our proposed **DualRAG** strategy, which combines global retrieval and keyword-guided retrieval to capture both high-level contextual knowledge and task-relevant details.

Algorithm 1 summarizes the end-to-end workflow in six stages.

(1) Initialization: M_c receives the user query Q (e.g., "How many airplanes are parked in this image, and what are their categories?") and image I.

(2) Task inference: M_c analyzes Q to infer the underlying task type(s) tp by mapping the query to predefined remote sensing task categories. When the query intent is ambiguous or under-specified, M_c may optionally retrieve domain knowledge from D_k to guide accurate task classification.

(3) Solution retrieval: The inferred task type tp is passed to the Solution Searcher M_s, which retrieves solution guidance g_s from the Solution Database D_s. This guidance encodes task-specific procedural workflows—for instance, performing object detection before fine-grained category classification for the aircraft counting query.

(4) Pipeline orchestration: Conditioned on (Q, g_s), M_c constructs an executable tool chain $\hat{T} = \{T_1, T_2, \dots\}$ from the Toolkit T.

(5) Tool execution: M_c invokes the tools in $\hat{T}$ sequentially or in parallel as required, processing the input image I and intermediate results. This stage produces task-specific outputs and, when applicable, a processed visual result $\hat{I}$ (e.g., detection bounding boxes).

(6) Knowledge-grounded response generation: If specialized knowledge is needed, M_c invokes M_k to retrieve g_k from D_k via DualRAG, then integrates tool outputs and knowledge into the final answer A.

Algorithm 1 Task-centric workflow of RS-Agent.

Require: User query Q and input remote sensing image I.
Ensure: Final answer A and processed image $\hat{I}$ (if required).
Initialization
Initialize the Central Controller M_c with (Q, I);
Task inference
M_c infers and decomposes the latent task representation tp from (Q, I), optionally grounded by retrieved domain knowledge;
Task-to-solution grounding
M_c grounds tp to task-specific solution guidance g_s retrieved from the Solution Space D_s via the Solution Searcher M_s;
Pipeline orchestration
Based on (Q, tp, g_s), M_c constructs an ordered tool execution plan $\hat{T} = \{T_1, T_2, \dots\}$;
Tool execution
Execute tools in $\hat{T}$ to process the input image I and intermediate results, producing outputs and the processed image $\hat{I}$ when applicable;
Knowledge-grounded interpretation (if required)
M_c enriches outputs with domain knowledge retrieved via M_k from D_k;
Response generation
Generate final answer A;

3.2 Central Controller

The Central Controller M_c is the brain of RS-Agent, managing multi-step reasoning and tool execution through modular interfaces. Built on an LLM with LangChain [49], M_c supports task decomposition, context management,

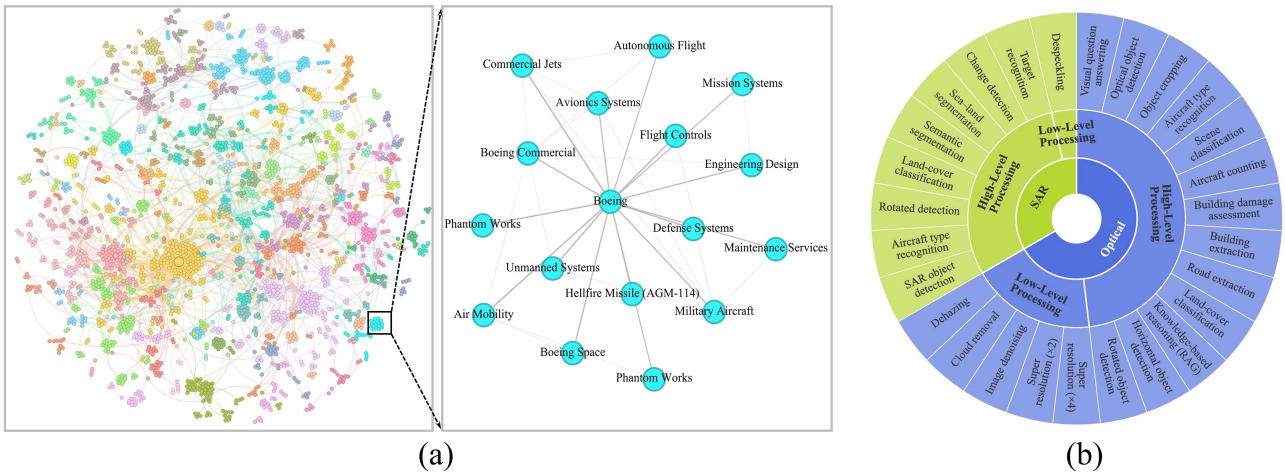


Figure 3 (Color online) Overview of the RS-Agent knowledge representation and toolkit organization. (a) RSaircraft knowledge graph. Nodes denote remote sensing knowledge entities, and edges represent typed semantic relations, forming a dense core with sparse peripheral components. (b) Hierarchical structure of the Toolkit T. Tools are organized into a three-level hierarchical structure and are accessed via standardized APIs for modular orchestration.

and dynamic tool invocation, and adopts a two-stage prompt-based reasoning process that embeds remote-sensing domain knowledge to guide task formulation.

Stage 1. Task inference and solution retrieval. Given a user query Q, M_c infers task type(s) tp by aligning Q with a predefined set of supported remote-sensing tasks via a structured classification prompt. Since user queries often describe high-level goals without specifying task formulation, scale, or temporal setting, the prompt guides M_c to distill remote-sensing attributes—sensor/modality, spatial granularity, temporal configuration, and analytical operation—and produce a compact, machine-readable task representation drawn exclusively from the supported task set.

The inferred tp is then forwarded to the Solution Searcher M_s, which retrieves candidate solution g_s—an expert-defined guideline for solving the task.

Stage 2. Task planning and tool execution. Leveraging the retrieved g_s and the original Q, the Central Controller M_c constructs a detailed task plan by selecting the most appropriate tools and arranging them into a logical, step-by-step workflow tailored to the user's intent. To facilitate the memory mechanism, M_c incorporates prior dialogue history to ensure accurate, context-aware reasoning.

3.3 Toolkit

The Toolkit T is the action engine of RS-Agent, providing a structured and extensible suite of analysis tools. Given remote sensing's strong modality dependency, multi-scale nature, and geometry-semantics coupling, T adopts a three-level hierarchy (Figure 3(b)): first, by sensing modality (optical vs. SAR); second, by analysis level (low-level processing such as denoising and super-resolution; high-level semantics such as classification and interpretation); third, by geometric and semantic characteristics including target semantics, spatial scale, resolution, and analysis granularity. All tools are accessed via standardized APIs for modular orchestration, and the Toolkit is extensible to new tools as needed. The framework currently includes 27 tools, with a representative subset shown in Figure 4. A complete list of all supported tools is provided in Table S3.

3.4 Solution Space

The Solution Space supports accurate task planning by compensating for LLMs' lack of domain-specific solution strategies, especially when task instructions are ambiguous, underspecified, or implicitly formulated [50, 51]—scenarios common in remote sensing. It comprises two components: the Solution Database D_s, storing expert-curated solutions, and the Solution Searcher M_s, which retrieves the most appropriate solution for a given user query.

3.4.1 *Solution database*

To provide RS-Agent with a reliable source of task-solving knowledge, we construct a curated Solution Database $D_s = \{d_s^1, d_s^2, \dots\}$, which stores expert-defined solutions for commonly encountered remote sensing tasks. This

Figure 4 (Color online) Qualitative result of RS-Agent. This figure shows several key tools that highlight the core capabilities of RS-Agent. The input image represents the image input by the user. The blue box shows the user's request, and the red box shows the RS-Agent's reply.

database acts as a structured knowledge base that guides task planning and tool invocation.

Solutions in D_s are built in three stages. First, domain experts manually design core solutions based on established analysis paradigms and tool capabilities, encoding task formulations, recommended tool chains, input requirements, and expected outputs. Second, this set is expanded via instruction-based LLM generation to produce paraphrased variants that improve retrieval robustness. Third, all generated variants are verified for correctness and executability by executing the prescribed tool chain on representative test inputs and checking that input/output formats are consistent across the pipeline.

Each solution document follows a unified structure: task description, applicable conditions, recommended tool sequence, input formats, and expected outputs. For example, a solution for object counting specifies an object-level detection tool followed by a counting step, translating a high-level request into an executable workflow. In the current implementation, D_s contains 34 solution documents in total: 27 single-tool templates and 7 multi-tool workflow templates. The 27 single-tool templates correspond one-to-one with the 27 tools in the Toolkit (Figure 3(b) and Table S3), each encoding the recommended invocation procedure for one tool. The 7 multi-tool templates are designed for RSVQA tasks, encoding composite pipelines that chain image preprocessing tools (e.g., denoising,

super-resolution) with visual question answering.

3.4.2 *Solution searcher with Task-Aware Retrieval*

To enable RS-Agent to accurately locate the most relevant solution for a given task, we design the Solution Searcher, a module dedicated to retrieving appropriate entries from the Solution Database. To enhance its effectiveness, we propose a Task-Aware Retrieval method, which augments the capabilities of large language models by integrating RAG into the solution selection process. This design leverages remote sensing's structured task taxonomy. Since user queries describe goals without explicit task specifications, Task-Aware Retrieval first infers the task type, then uses it as the retrieval key for more effective matching.

As shown in the top right of Figure 2, Task-Aware Retrieval has two steps: Task Inference, which identifies the task type from the query, and Solution Retrieval, which uses this type to fetch the most relevant solution.

Task inference: When a user query Q is received, the Central Controller M_c first interprets the user's intent and infers the underlying tp using a system prompt designed to guide task recognition. This step can be viewed as a query rewriting process, where the LLM reformulates or abstracts the input into a semantically meaningful task label, enabling more structured downstream retrieval. Critically, this prompt encodes remote sensing domain knowledge—including sensing modality distinctions, analysis-level hierarchies, and task granularity cues—enabling the LLM to resolve ambiguities that general-purpose task inference would miss. The complete prompt design is provided in Appendix A.1.

Solution retrieval: The inferred tp is then passed to the Solution Searcher M_s, which retrieves relevant strategies from D_s to support tool selection and reasoning. To achieve retrieval, each solution document in D_s and tp is encoded into dense vector representations using a sentence embedding model [52]. M_s then employs FAISS [53] as a high-performance vector similarity search function f_s to retrieve relevant documents and form the Solution Guidance g_s:

$$g_s = f_s(tp, D_s, k), \quad k = |tp|\,, \tag{1}$$

where $|tp|$ is the number of task types in tp. This yields a compact set of complementary solution documents covering the implied task requirements.

By combining retrieval and generation, Task-Aware Retrieval enables RS-Agent to leverage expert-defined tool solutions and reliably handle complex remote sensing tasks.

3.5 Knowledge Space

The Knowledge Space strengthens RS-Agent's ability to handle queries requiring specialized domain knowledge. It contains two components: a Knowledge Database D_k storing curated remote sensing knowledge, and a Knowledge Searcher M_k that retrieves relevant documents from D_k.

3.5.1 *Knowledge Database*

The Knowledge Database D_k serves as a structured repository of domain knowledge that supports RS-Agent in tasks requiring factual accuracy, technical terminology, or background-specific reasoning. To validate RS-Agent's capability in retrieving and generating domain-specific knowledge, D_k is designed as a general-purpose remote sensing knowledge corpus, covering concepts related to sensing modalities, platforms, targets, observation principles, and domain-specific semantics. The knowledge corpus is collected from Wikipedia and preprocessed through cleaning and normalization to form semantically coherent text chunks suitable for structured knowledge construction. To evaluate RS-Agent's capability in retrieving and generating domain-specific knowledge, we further instantiate a domain-focused subset of D_k as a case study. Specifically, as the current RS-Agent is initially developed to support aircraft-related remote sensing analysis, we construct a specialized knowledge subset, denoted as RSaircraft, which contains curated information on 65 aircraft categories, including fighter jets, commercial airliners, drones, and helicopters. This subset exemplifies how the general knowledge corpus can be adapted to other specific application domains. The Knowledge Database is designed for extensibility. Its knowledge graph storage represents content as entity-relation triples in vector spaces, decoupling the schema from any sub-domain ontology. The DualRAG retrieval logic is likewise decoupled from the stored content, so switching corpora requires no pipeline changes. A standardized API between the Knowledge Space and the Central Controller further ensures that new corpora can be ingested without modifying the agent logic. Consequently, expanding to new sub-domains requires only corpus ingestion and knowledge graph reconstruction. We adopt a knowledge graph-based storage structure [48], in which each text chunk is processed by an LLM to extract entities and relational triples. These entities and relations are embedded into separate vector spaces, enabling structure-aware semantic retrieval. This representation supports

multi-hop reasoning and domain-informed query expansion, facilitating more accurate and context-rich responses. A local subgraph of the constructed knowledge graph is illustrated in Figure 3(a).

3.5.2 *Knowledge Searcher with DualRAG*

The Knowledge Searcher retrieves relevant documents from D_k for domain-specific queries. Existing single-query or keyword-based methods struggle with complex remote sensing queries, where entities, relations, and constraints are implicit, unevenly emphasized, or scattered across fragments. We thus propose DualRAG, a domain-adapted retrieval method tailored for remote sensing knowledge retrieval, which introduces weighted keyword-aware query decomposition and a dual-path retrieval architecture.

As shown in Figure 2, DualRAG has two components: (1) keyword generation and weight assignment, which scores semantically important keywords to emphasize key entities and relations; and (2) dual-path retrieval, which fuses global semantic search with weighted keyword-level retrieval.

Keyword generation and weight assignment: Given a user query Q, the Knowledge Searcher M_k generates a ranked set of retrieval-oriented keywords and assigns an importance weight to each keyword based on prompt-defined prioritization and scoring rules to facilitate remote-sensing knowledge retrieval. In particular, M_k prioritizes RS domain-specific and discriminative terms while filtering out generic or low-information words, producing a compact weighted representation. This weighted keyword set acts as a structured query expansion that narrows the retrieval space and steers the downstream stage toward the most relevant remote-sensing knowledge.

$$\{(kw_i, w_i)\}_{i=1}^{m} = M_c(Q), \quad \text{where } w_i \in [1, 10]\,. \tag{2}$$

Each tuple (kw_i, w_i) represents a keyword kw_i and its corresponding importance weight w_i, where larger w_i indicates higher expected utility for retrieval. The weighting is governed by a domain-informed prioritization hierarchy that reflects the structure of remote sensing queries, so that retrieval-critical aspects are emphasized. The complete keyword extraction prompt is provided in Appendix A.2. Here, m denotes the total number of extracted keyword-weight pairs.

Dual-path retrieval: To enhance the relevance and coverage of the retrieved knowledge documents, our DualRAG adopts a dual-path retrieval strategy that combines global and keyword-level search. In the global retrieval path, the generated kw_i are concatenated to form a single string:

$$kw_{\text{global}} = \text{Concat}(kw_1, kw_2, \dots)\,. \tag{3}$$

This string retrieves the top-N documents from the Knowledge Database D_k via similarity search algorithm f_k:

$$d_{\text{global}} = f_k(kw_{\text{global}}, D_k, N)\,. \tag{4}$$

This path captures the overall semantic structure and inter-keyword dependencies.

In the keyword retrieval path, each kw_i is treated as an individual query. The number of the retrieved knowledge documents n_i is allocated based on the assigned weight w_i:

$$n_i = \left\lfloor \frac{w_i}{\sum_{j=1}^{m} w_j} \times N \right\rfloor, \tag{5}$$

where each n_i is obtained by floor division so that keywords with higher importance weight are allocated more documents. The remaining quota $N - \sum_i n_i$ is distributed one-by-one to keywords in descending order of w_i, ensuring $\sum_i n_i = N$. Then, each kw_i retrieves its own top-n_i documents:

$$d_{\text{keyword}} = \bigcup_{i=1}^{m} f_k(kw_i, D_k, n_i)\,, \tag{6}$$

where m is the number of keywords. This path retrieves fine-grained information complementing the global path.

The retrieved documents form the Knowledge Guidance $g_k = d_{\text{global}} \cup d_{\text{keyword}}$, which provides the LLM with domain-specific knowledge. Duplicate documents appearing in both paths are removed during the union operation, and each document is retained at most once in g_k.

By combining both paths, DualRAG balances semantic coherence and informational diversity. Unlike LightRAG [48]'s single pipeline, DualRAG's dual-path structure separates global and local retrieval scopes—capturing both macro-level semantics and fine-grained entity details—and is particularly effective for complex remote sensing queries involving multiple entities.

Table 1 Comparison (%) of task planning accuracy with SOTA RS-ChatGPT. The best results are in bold.

Task	gpt-3.5-turbo-1106		gpt-3.5-turbo		gpt-4o-mini	
	RS-ChatGPT	RS-Agent (Ours)	RS-ChatGPT	RS-Agent (Ours)	RS-ChatGPT	RS-Agent (Ours)
Object counting	47.37	94.74	97.74	100	100	100
Object detection	78.95	84.21	52.63	89.47	42.11	94.74
Land use segmentation	90.48	95.24	95.24	95.24	90.48	90.48
Image captioning	42.86	57.14	85.71	61.90	100	95.24
Scene classification	90.48	85.71	80.95	100	100	90.48
Instance segmentation	57.89	78.95	78.95	68.42	89.47	100
Edge detection	100	100	84.21	100	94.74	100
Average accuracy	72.66	**84.89**	82.01	**87.77**	88.49	**95.68**

4 Experiments

In this section, we present comprehensive experiments to validate the effectiveness of RS-Agent across multiple dimensions. We organize our evaluation as follows: (i) implementation details (Subsection 4.1); (ii) task planning accuracy versus SOTA agents and across open-source and proprietary LLMs (Subsection 4.2); (iii) performance over single-model baselines on scene classification, visual question answering, and object counting (Subsection 4.3); (iv) ablation studies on Task-Aware Retrieval and DualRAG (Subsections 4.4 and 4.5); (v) qualitative results on multi-step reasoning (Subsection 4.6).

4.1 Implementation details

To ensure the flexibility and generalizability of RS-Agent, we build the Central Controller with the LangChain framework [49] and general-purpose large language models. Unless otherwise specified, RS-Agent uses the Qwen 2.5 32B-Instruct model [27], balancing the model capability with inference speed.

For Task-Aware Retrieval, we employ the open-source m3e-base [52] embedding model and index vectors with FAISS [53]. For DualRAG, we return the top-N documents with $N = 30$.

RS-Agent is equipped with a toolkit of specialized tools designed to support diverse remote sensing tasks, ranging from low-level vision operations to high-level semantic tasks. For performance evaluation, we instantiate the toolkit with a representative set of 18 tasks to benchmark RS-Agent under a controlled and reproducible setting.

4.2 Task planning accuracy

A core capability of RS-Agent is its ability to accurately plan tasks and invoke the appropriate tools in response to user queries. In this section, we evaluate the task planning accuracy of RS-Agent.

4.2.1 *Task planning accuracy compared to SOTA agent*

Dataset. We construct a dataset for task planning, measuring accuracy by whether the first invoked tool matches the correct one. For fair comparison with RS-ChatGPT [21], we adopt its seven core tasks: scene classification, land use segmentation, object detection, image captioning, edge detection, object counting, and instance segmentation. We curate ∼20 query-solution pairs per task, mixing clear and ambiguous instructions, and use the same backbone LLMs as RS-ChatGPT.

Results. As shown in Table 1, RS-Agent consistently outperforms RS-ChatGPT across all backbone LLMs. With gpt-3.5-turbo-1106, accuracy rises from 72.66% to 84.89% (+12.23%); similar gains hold for gpt-3.5-turbo (+5.76%) and gpt-4o-mini (+7.19%). These results demonstrate that RS-Agent exhibits superior task comprehension and decision-making ability, allowing it to reliably select appropriate tools and execute tasks more effectively.

4.2.2 *Task planning accuracy with different LLMs*

We evaluate RS-Agent's adaptability when paired with closed-source (GPT series) and open-source LLMs.

Datasets. To comprehensively evaluate RS-Agent's task planning accuracy across different LLM configurations, we construct a new dataset tailored to the full toolkit currently supported by our system. Specifically, we define a set of 18 distinct tasks and design 20 query-solution pairs for each, resulting in a total of 360 evaluation instances. These queries span a diverse range of task types and complexity levels, and include both well-specified and ambiguous instructions to assess the model's robustness in understanding intent and selecting the appropriate tool. Unlike

Table 2 Task planning accuracy (%) of RS-Agent with different LLMs on 10 representative tasks. Complete results for all tasks are reported in Table S2. "B" denotes the number of parameters in billions. Numbers in parentheses (t/s) indicate model inference speed in tokens per second on NVIDIA RTX4090 GPUs.

Task	ChatGPT			LLaMa 3.1		Qwen2.5			DeepSeek
	3.5-turbo-1106 (87.71 t/s)	3.5-turbo (65.03 t/s)	4o-mini (58.87 t/s)	8B (100.78 t/s)	70B (17.71 t/s)	14B (69.61 t/s)	32B (36.77 t/s)	72B (16.24 t/s)	r1:70B (18.25 t/s)
Image captioning	55.00	45.00	90.00	15.00	60.00	70.00	80.00	80.00	10.00
Object detection	75.00	60.00	95.00	30.00	90.00	90.00	85.00	100	85.00
Scene classification	20.00	90.00	100	80.00	90.00	90.00	100	100	50.00
SAR detection	30.00	100	100	75.00	95.00	100	100	100	100
SAR plane classification	100	100	100	100	100	100	100	100	90.00
Knowledge search	100	100	100	100	80.00	100	100	100	10.00
Building extraction	10.00	70.00	100	55.00	100	100	100	100	100
Rotated detection	15.00	35.00	100	85.00	90.00	100	100	100	100
Semantic segmentation	60.00	100	100	80.00	100	100	100	100	80.00
Land use classification	15.00	100	100	75.00	100	100	100	95.00	95.00
Average accuracy	48.00	80.00	98.50	69.50	90.50	95.00	96.50	97.50	72.00

the previous dataset, which was based on the RS-ChatGPT toolkit for comparative evaluation, this dataset fully leverages RS-Agent's expanded capabilities and provides a more comprehensive assessment of its performance under different LLM configurations.

Results. Table 2 reports results on 10 representative tasks selected to cover optical, SAR, and cross-modal categories (complete 18-task results are in Table S2). RS-Agent achieves robust tool selection across proprietary and open-source LLMs. Open-source models such as Qwen2.5-72B (97.50%) and 32B (96.50%) match or exceed the ChatGPT series. This result highlights that well-tuned open-source models can match the reasoning capabilities of state-of-the-art commercial models in task planning scenarios.

Within a model family, larger parameters generally improve accuracy: Qwen2.5 rises from 14B (95.00%) to 72B (97.50%), and LLaMa 3.1 from 8B (69.50%) to 70B (90.50%); but inference speed drops accordingly (Qwen2.5-72B: 16.24 t/s vs. 14B: 69.61 t/s). Balancing both, Qwen2.5-14B and 32B are practical choices, while LLaMa 3.1-8B underperforms (69.50%), indicating smaller variants may not generalize to tool planning.

In summary, from the above results, we can observe that the model performance varies significantly across different tasks, especially those with unclear tool boundaries. This highlights the need for clearer task definitions and more robust planning strategies in future work.

4.3 Performance on remote sensing tasks

Like MLLMs, agents perform diverse tasks through natural-language interaction. To highlight RS-Agent's advantages, we evaluate it on three representative remote-sensing applications: object counting, visual question answering (VQA), and scene classification.

4.3.1 *Object counting*

Object counting estimates the number of target objects in an image, supporting remote-sensing applications such as vehicle counting for traffic analysis and infrastructure density assessment for urban planning, where accurate quantification is critical for resource management. We employ YOLOv8x-obb [54], pre-trained on DOTAv1 [55], which excels at oriented object detection.

Datasets. We use the DOTAv1 [55] validation set (458 images, 1099 questions) with the prompt: "What is the number of the {object}? Answer the question using a number.", where {object} is one of 15 typical remote-sensing target types (e.g., plane, ship, bridge).

Metric. We use three metrics: absolute accuracy (percentage of predictions exactly matching the ground truth), interval-matching accuracy (a prediction is correct if it falls in the same predefined range as the ground truth, with intervals 0, 1–10, 11–100, 101–1000, and >1000), and relative error e_r, defined as

$$e_r = \frac{1}{N}\sum_{i=1}^{N}\log\left(1+\frac{|gt_i - p_i|}{gt_i}\right), \tag{7}$$

where gt_i and p_i are the ground truth and prediction of the i-th case, and N is the number of counting problems.

Table 3 Object counting accuracy on DOTA [55]. The best results are in bold.

Method	Absolute accuracy (%)	Interval match (%)	Relative error
GeoChat	17.65	74.61	0.65
LHRS-Bot	14.92	60.78	0.56
RS-Agent (Ours)	**33.30**	**75.98**	**0.28**

Table 4 Visual question answering accuracy (%) on RSVQA-LR datasets [31]. Top: general-purpose MLLMs; bottom: models tailored for remote sensing. The best results are in bold.

Method	Rural/urban	Presence	Comparison	Avg.
LLaVA-1.5 [56]	59.22	73.16	65.19	65.86
MiniGPTv2 [57]	60.02	51.64	67.64	59.77
InstructBLIP [58]	62.62	48.83	63.92	59.12
mPLUG-Owl2 [59]	57.99	74.04	65.04	65.69
QWen-VL-Chat [26]	62.00	47.65	54.64	58.73
RSVQA [31]	90.00	87.47	81.50	86.32
SkyEyeGPT [36]	88.93	88.63	75.00	84.16
LHRS-Bot [32]	89.07	88.51	90.00	89.19
GeoChat [6]	94.00	91.09	90.33	90.70
RS-Agent (Ours)	**97.00**	**91.07**	**90.58**	**90.88**

Results. As shown in Table 3, RS-Agent outperforms GeoChat [6] and LHRS-Bot [32] on all metrics: 33.30% absolute accuracy (vs. 17.65%/14.92%), 75.98% interval-matching (vs. 74.61%/60.78%), and 0.28 relative error (vs. 0.65/0.56), yielding more precise and consistent predictions and confirming the robustness of RS-Agent for object counting. This advantage stems from the agent's ability to decompose counting into detection and aggregation steps, leveraging a specialized detector that end-to-end MLLMs lack.

4.3.2 *Visual question answering*

Visual question answering (VQA) requires understanding visual content and answering natural-language questions, jointly testing visual perception, language comprehension, and cross-modal reasoning. We adopt a multi-tool approach: RS-Agent invokes super-resolution, denoising, and GeoChat for different question types in RSVQA, retrieving strategies from D_s via Task-Aware Retrieval.

Datasets. We use RSVQA-LR [31], with 772 satellite images and 77232 question-answer pairs across four question types: presence, comparison, rural/urban classification, and counting.

Compared methods. We compare RS-Agent with two groups of multimodal large language models: (1) general-purpose models trained on broad, non-specialized corpora, e.g., LLaVA-1.5 [56], MiniGPTv2 [57], InstructBLIP [58], mPLUG-Owl2 [59] and QWen-VL-Chat [26] and (2) remote sensing-specific models fine-tuned on domain-specific remote sensing datasets, e.g., RSVQA [31], SkyEyeGPT [36], LHRS-Bot [32] and GeoChat [6].

Results. As shown in Table 4, with multi-tool invocation in a zero-shot setting, RS-Agent reaches 97.00% on Rural/Urban, 91.07% on Presence, and 90.58% on Comparison, with an average of 90.88%—the highest among all tested models, surpassing MLLMs such as GeoChat and LHRS-Bot on RSVQA-LR. These results reflect the benefit of dynamically selecting and composing task-specific tools according to question type, a capability beyond end-to-end MLLMs.

4.3.3 *Scene classification*

Scene classification categorizes satellite or aerial images into predefined scene types, testing the ability to capture spatial patterns, textures, and contextual cues. As the scene classifier, RS-Agent uses a vision transformer (ViT-B16) [60] self-supervised pretrained on ImageNet [61] and fine-tuned on RSSDIVCS [62] (70 categories, 800 images each at 256×256, 80% for training).

Datasets. We evaluate on RSSDIVCS [62], AID [63] (Google Earth aerial images, 30 classes), and UCMerced [64] (2100 images, 21 land use categories).

Results. As shown in Table 5, RS-Agent achieves 98.00% on RSSDIVCS [62] (vs. GeoChat 51.43%, LHRS-Bot 63.85%), 96.88% on AID [63] (vs. 72.03%/91.29%), and 98.63% on UCMerced [64] (vs. 84.43%/96.63%), consistently

Table 5 Scene classification accuracy (%) on RSSDIVCS [62], AID [63], and UCMerced [64]. The best results are in bold.

Model	RSSDIVCS	AID	UCMerced
MiniGPTv2 [57]	–	52.60	62.90
Qwen-VL-Chat [26]	–	52.60	62.90
LLaVA-1.5 [56]	–	51.00	68.00
GeoChat [6]	51.43	72.03	84.43
LHRS-Bot [32]	63.85	91.29	96.63
RS-Agent (Ours)	**98.00**	**96.88**	**98.63**

Table 6 Ablation study of Task-Aware Retrieval: tool execution accuracy (%) on single-tool and multi-tool tasks. The best results are in bold.

Method	Object counting	Object detection	Land use segmentation	Image captioning	Scene classification	Instance segmentation	Edge detection	Single-tool accuracy	Multi-tool accuracy
Baseline	10.53	15.79	95.24	85.71	76.19	100	94.74	69.06	9.42
Baseline+T_{inf}	52.63	36.84	95.24	95.24	95.24	100	100	82.01	10.87
Baseline+S_{ret}	52.63	31.58	100	80.95	52.38	94.74	94.74	73.38	11.38
Baseline+T_{inf}+S_{ret}	**100**	**84.21**	90.48	**95.24**	**100**	84.21	**100**	**93.53**	**90.62**

outperforming all baselines. The large margins confirm that correctly routing images to a fine-tuned domain classifier via the agent significantly outperforms the implicit classification ability of end-to-end MLLMs.

4.4 Effectiveness of the Task-Aware Retrieval method

Task-Aware Retrieval derives g_s via task inference and solution retrieval (Subsection 3.4). To assess its contribution, we ablate four progressively-built variants. (1) **Baseline**: RS-Agent without the Solution Space; (2) **Baseline+T_{inf}**: only Task Inference enabled; (3) **Baseline+S_{retr}**: only Solution Retrieval enabled, with Q used directly as the retrieval query; (4) **Baseline+T_{inf}+S_{retr}**: the full RS-Agent.

Evaluation tasks. We test two task categories: single-tool tasks requiring one tool, and multi-tool tasks coupling several tools to assess complex workflows.

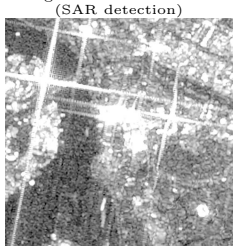
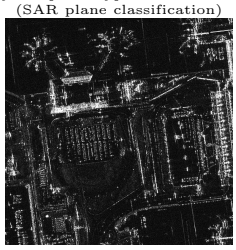
Metric. Task planning accuracy is reported as **single-tool accuracy** (whether the correct tool is selected for single-tool tasks) and **multi-tool accuracy** (whether all required tools are invoked in a valid order for multi-tool tasks). We additionally report end-to-end task success rates based on final-answer correctness in Table S1.

Results. As shown in Table 6, the Baseline achieves 69.06% single-tool accuracy but only 9.42% multi-tool task planning accuracy, as it frequently fails to invoke the correct tools (e.g., 10.53% on object counting). Adding Task Inference or Solution Retrieval alone yields moderate single-tool gains (82.01% and 73.38%) with limited multi-tool improvement (10.87% and 11.38%). Combining both achieves the best performance: 93.53% single-tool and 90.62% multi-tool accuracy, confirming a strong synergistic effect. Notably, the underlying toolset is identical across all variants; the gains are entirely attributable to the orchestration mechanism. Multi-tool end-to-end task success rates are reported in Table S1, and representative case studies are provided in Table 7.

To provide deeper insight into the roles of Task Inference and Solution Retrieval, we present representative case studies in Table 7. Each case compares the MLLM baseline (GeoChat [6]) with the four RS-Agent ablation variants. These cases further illustrate that the core value of the agent lies in orchestration: even when powerful specialized tools are available, incorrect tool selection or improper execution ordering leads to task failure, underscoring the necessity of the orchestration mechanism.

Analysis. These cases highlight three key findings. First, the end-to-end MLLM (GeoChat) [6] consistently fails on tasks requiring precise quantitative outputs, modality-specific processing, or pixel-level predictions, as it can only produce text-based descriptions without invoking specialized tools. In contrast, RS-Agent leverages tool orchestration to deliver actionable, accurate results. Second, within RS-Agent's ablation variants: (1) Task Inference is most critical for queries with ambiguous task boundaries and cross-modal disambiguation (e.g., SAR vs. optical); (2) Solution Retrieval provides precise procedural guidance but may retrieve mismatched templates when used alone; (3) occasional over-abstraction by Task Inference (e.g., mapping "landuse" to "segmentation") is corrected when combined with Solution Retrieval. These observations confirm both the advantage of the agentic paradigm over end-to-end MLLMs and the synergistic design of Task-Aware Retrieval. Third, the RSVQA Presence [31] case demonstrates RS-Agent's strength in multi-tool tasks: all single-component variants invoke only `geochat` and produce an incorrect answer, whereas the full system with both Task Inference and Solution Retrieval correctly plans

Table 7 (Color online) Case studies comparing the MLLM baseline (GeoChat) and RS-Agent ablation variants. Result: ✓ = correct, ✗ = incorrect.

Query	Variant	Tool execution\output	Analysis	Result
"What targets are in this SAR image?" (SAR detection) 	MLLM	"The image shows some bright spots..."	No SAR-specific capability; vague description instead of detection results.	✗
	Baseline	optical_det.(img) → optical results	Modality not distinguished; invoked optical tool on SAR data.	✗
	$+T_{\text{inf}}$	sar_det.(img) → correct detections	Correctly invoked sar_det.	✓
	$+S_{\text{retr}}$	sar_det.(img) → correct detections	Correctly invoked sar_det.	✓
	$+T_{\text{inf}}+S_{\text{retr}}$	sar_det.(img) → correct detections	Correctly invoked sar_det.	✓
"Generate a landuse map from this image." (Land use seg.) 	MLLM	"urban areas, vegetation, roads..."	Cannot generate pixel-level segmentation; text-only output.	✗
	Baseline	landuse_seg.(img) → correct map	Correctly invoked landuse_seg.	✓
	$+T_{\text{inf}}$	sem_seg.(img) → wrong output	Over-abstracted as "Semantic Segmentation"; wrong tool.	✗
	$+S_{\text{retr}}$	landuse_seg.(img) → correct map	Correctly invoked landuse_seg.	✓
	$+T_{\text{inf}}+S_{\text{retr}}$	landuse_seg.(img) → correct map	Correctly invoked landuse_seg.	✓
"Identify the plane type in this SAR image." (SAR plane classification) 	MLLM	"I can see an aircraft..."	No SAR understanding capability; generic guess based on optical priors.	✗
	Baseline	optical_plane_type(img) → incorrect	Misidentified as optical task; applied optical classifier to SAR image.	✗
	$+T_{\text{inf}}$	sar_plane_type(img) → "A220."	Correctly invoked sar_plane_type	✓
	$+S_{\text{retr}}$	sar_plane_type(img) → "A220."	Correctly invoked sar_plane_type	✓
	$+T_{\text{inf}}+S_{\text{retr}}$	sar_plane_type(img) → "A220."	Correctly invoked sar_plane_type	✓
"Is there a circular residential building in the image?" (RSVQA) 	MLLM	"Yes"	Directly answered without preprocessing; missed fine-grained details in low-quality image.	✗
	Baseline	geochat(Q, img) → "Yes"	Invoked geochat only; no image enhancement performed.	✗
	$+T_{\text{inf}}$	geochat(Q, img) → "Yes"	Correctly inferred "Presence"; but no preprocessing guidance retrieved.	✗
	$+S_{\text{retr}}$	geochat(Q, img) → "Yes"	Retrieved wrong guidance.	✗
	$+T_{\text{inf}}+S_{\text{retr}}$	denoise(img) → super_resolution(img) → geochat(Q, img) → "No"	Correctly invoked multi-tool pipeline; image enhancement enabled accurate answer.	✓

a three-step pipeline (`denoise` → `super_resolution` → `geochat`), improving image quality before visual reasoning and yielding the correct answer.

4.5 Effectiveness of the DualRAG method

To verify DualRAG's effectiveness, we compare RS-Agent with DualRAG against LightRAG [48].

Datasets. In this study, we utilize two datasets: the RSaircraft dataset, which we have specifically constructed for remote sensing applications, and the Mix dataset from the UltraDomain benchmark [65], a general-domain RAG dataset. The RSaircraft dataset focuses on aircraft-related information, tailored for remote sensing tasks, and was built by systematically collecting and processing structured data from publicly available sources.

Metric. Following LightRAG [48], gpt-4o-mini serves as an automated pairwise judge that selects the superior response between two candidates along four dimensions—comprehensiveness, diversity, empowerment, and overall quality—under three keyword usage modes: local, global, and hybrid.

Results. Table 8 reports win rates of LightRAG [48] vs. RS-Agent with DualRAG. RS-Agent consistently surpasses LightRAG, with notably larger gains on RSaircraft, leading in all dimensions and modes—local mode improves comprehensiveness by 17.00% (58.00% vs. 41.00%) and hybrid mode shows the largest gaps (+26.40%

Table 8 Win rates (%) of LightRAG vs. RS-Agent with DualRAG across two datasets and four evaluation dimensions. The better results are in bold.

Mode	Dimensions	RSaircraft		Mix	
		LightRAG	RS-Agent (Ours)	LightRAG	RS-Agent (Ours)
Local	Comprehensiveness	41.00	**58.00**	48.40	**51.60**
	Diversity	43.40	**56.40**	50.00	**50.00**
	Empowerment	43.40	**56.40**	**50.80**	49.20
	Overall	42.80	**57.20**	**50.80**	49.20
Global	Comprehensiveness	40.40	**59.60**	48.80	**51.20**
	Diversity	43.60	**56.40**	46.80	**53.20**
	Empowerment	39.20	**60.80**	**51.60**	48.40
	Overall	39.60	**60.40**	49.20	**50.80**
Hybrid	Comprehensiveness	36.80	**63.20**	46.40	**53.60**
	Diversity	37.00	**62.00**	44.80	**55.20**
	Empowerment	35.60	**64.40**	47.60	**52.40**
	Overall	36.00	**64.00**	47.20	**52.80**

comprehensiveness, +25.00% empowerment, +28.00% overall). Gains on Mix are smaller but consistent, confirming DualRAG's effectiveness and robustness across datasets and retrieval settings.

4.6 Qualitative results

Figure 4 illustrates representative tasks showcasing RS-Agent's core capabilities. Each tool demonstrates strong performance, producing fluent, accurate, and contextually appropriate responses. Outputs are not only visually and semantically aligned with the input queries, but also exhibit robustness across various image types and modalities. Beyond accuracy, RS-Agent also excels in interpreting natural language instructions, even when they are underspecified or ambiguous. For example, a vague query such as What do you see in this image? is correctly interpreted as an image description task, and the system autonomously invokes the appropriate `image captioning` model to produce a detailed and relevant response.

This reflects strong task planning capabilities that bridge user intent and technical execution through semantic understanding rather than keyword matching. The modular toolkit allows seamless integration of new tools, ensuring long-term scalability and adaptability across diverse remote sensing scenarios.

5 Conclusion

We propose RS-Agent, an AI agent tailored for remote sensing that seamlessly integrates diverse tools to support a wide range of tasks across multiple imaging modalities. The proposed architecture enables user query understanding, complex task planning, tool execution, and context-aware memory. Our Task-Aware Retrieval method significantly improves task planning accuracy, enhancing both efficiency and reliability in complex scenarios. In addition, the DualRAG mechanism provides access to specialized domain knowledge, enabling RS-Agent to handle complex technical queries. Extensive experiments demonstrate that RS-Agent consistently outperforms state-of-the-art multimodal large language models across multiple remote sensing benchmarks. Moreover, RS-Agent supports efficient integration of new tools while retaining previously learned capabilities. Overall, RS-Agent shows strong potential to streamline remote sensing workflows, reduce reliance on domain experts, and enable more accessible, high-quality analysis in real-world applications.

Despite its strong performance, RS-Agent has several limitations. The Task Inference module can occasionally over-abstract user queries, leading to incorrect tool selection, and the agent sometimes generates malformed tool arguments. Reinforcement learning from execution feedback or supervised fine-tuning on tool-calling traces could address these issues. In multi-tool workflows, upstream errors can propagate to downstream tools, calling for intermediate quality checks and adaptive re-planning. On the evaluation side, the current metrics focus on tool-selection correctness and final-answer accuracy but do not diagnose intermediate step quality at a fine-grained level; developing more comprehensive workflow-level evaluation schemes is an important direction for future work. Additionally, the Knowledge and Solution Databases currently have limited coverage and rely on manual curation; automated web-based knowledge collection could broaden their scope. Finally, domain-specific fine-tuning or distillation could reduce the system's dependence on large proprietary LLMs.

Supporting information Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62301063, 42501551) and Funds for National Key Laboratory of Inertial Measurement.

References

1 Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: Proceedings of International Conference on Machine Learning, 2021. 8748–8763
2 Sun Q, Fang Y, Wu L, et al. Eva-clip: Improved training techniques for clip at scale. 2023. ArXiv:2303.15389
3 Tao C, Xiao R, Wang Y, et al. A general transitive transfer learning framework for cross-optical sensor remote sensing image scene understanding. IEEE J Sel Top Appl Earth Observations Remote Sens, 2023, 16: 4248–4260
4 Bashmal L, Bazi Y, Melgani F, et al. Visual question generation from remote sensing images. IEEE J Sel Top Appl Earth Observations Remote Sens, 2023, 16: 3279–3293
5 Hu Y, Yuan J, Wen C, et al. RSGPT: a remote sensing vision language model and benchmark. ISPRS J Photogrammetry Remote Sens, 2025, 224: 272–286
6 Kuckreja K, Danish M S, Naseer M, et al. Geochat: grounded large vision-language model for remote sensing. 2023. ArXiv:2311.15826
7 Sun X, Wang P, Wang C, et al. PBNet: part-based convolutional neural network for complex composite object detection in remote sensing imagery. ISPRS J Photogrammetry Remote Sens, 2021, 173: 50–65
8 Li X, Du Z, Huang Y, et al. A deep translation (GAN) based change detection network for optical and SAR remote sensing images. ISPRS J Photogrammetry Remote Sens, 2021, 179: 14–34
9 Wang L, Li R, Zhang C, et al. UNetFormer: a UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. ISPRS J Photogrammetry Remote Sens, 2022, 190: 196–214
10 Zhu Q, Guo X, Deng W, et al. Land-use/land-cover change detection based on a Siamese global learning framework for high spatial resolution remote sensing imagery. ISPRS J Photogrammetry Remote Sens, 2022, 184: 63–78
11 Xi Z H, Chen W X, Guo X, et al. The rise and potential of large language model based agents: a survey. Sci China Inf Sci, 2025, 68: 121101
12 Reed S, Zolna K, Parisotto E, et al. A generalist agent. 2022. ArXiv:2205.06175
13 Park J S, O'Brien J, Cai C J, et al. Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023. 1–22
14 Wang L, Ma C, Feng X, et al. A survey on large language model based autonomous agents. Front Comput Sci, 2024, 18: 1–26
15 Ruan J, Chen Y, Zhang B, et al. Tptu: task planning and tool usage of large language model-based AI agents. 2023. ArXiv:2308.03427
16 Cui Q M, You X H, Wei N, et al. Overview of AI and communication for 6G network: fundamentals, challenges, and future research opportunities. Sci China Inf Sci, 2025, 68: 171301
17 Chen Z R, Zhang C Y, Liu C Y, et al. Towards wireless native big AI model: the mission and approach differ from large language model. Sci China Inf Sci, 2025, 68: 170303
18 Du S, Tang S, Wang W, et al. Tree-GPT: modular large language model expert system for forest remote sensing image understanding and interactive analysis. 2023. ArXiv:2310.04698
19 Liu C, Chen K, Zhang H, et al. Change-agent: toward interactive comprehensive remote sensing change interpretation and analysis. IEEE Trans Geosci Remote Sens, 2024, 62: 1–16
20 Zhu L, Wu J, Wang B, et al. Rs-agent: large language models guided agent system for remote sensing image generation. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2024. 7020–7024
21 Guo H, Su X, Wu C, et al. Remote sensing ChatGPT: solving remote sensing tasks with ChatGPT and visual models. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2024. 11474–11478
22 Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 33: 1877–1901
23 Wang C, Lv Y, Mao Y, et al. Robust fast adaptation from adversarially explicit task distribution generation. 2024. ArXiv:2407.19523
24 Lv Y, Wang Q, Liang D, et al. Theoretical investigations and practical enhancements on tail task risk minimization in meta learning. 2024. ArXiv:2410.22788
25 Touvron H, Lavril T, Izacard G, et al. Llama: open and efficient foundation language models. 2023. ArXiv:2302.13971
26 Bai J, Bai S, Yang S, et al. Qwen-VL: a frontier large vision-language model with versatile abilities. 2023. ArXiv:2308.12966
27 Yang A, Yang B, Zhang B, et al. Qwen2. 5 technical report. 2024. ArXiv:2412.15115
28 Guo D, Yang D, Zhang H, et al. Deepseek-r1: incentivizing reasoning capability in LLMs via reinforcement learning. 2025. ArXiv:2501.12948
29 Shi Z, Zou Z. Can a machine generate humanlike language descriptions for a remote sensing image? IEEE Trans Geosci Remote Sens, 2017, 55: 3623–3634
30 Wang Q, Huang W, Zhang X, et al. Word-sentence framework for remote sensing image captioning. IEEE Trans Geosci Remote Sens, 2020, 59: 10532–10543
31 Lobry S, Marcos D, Murray J, et al. RSVQA: visual question answering for remote sensing data. IEEE Trans Geosci Remote Sens, 2020, 58: 8555–8566
32 Muhtar D, Li Z, Gu F, et al. Lhrs-bot: empowering remote sensing with VGI-enhanced large multimodal language model. 2024. ArXiv:2402.02544
33 Wang J, Ma A, Chen Z, et al. EarthVQANet: multi-task visual question answering for remote sensing image understanding. ISPRS J Photogrammetry Remote Sens, 2024, 212: 422–439
34 Luo J, Pang Z, Zhang Y, et al. SkysenseGPT: a fine-grained instruction tuning dataset and model for remote sensing vision-language understanding. 2024. ArXiv:2406.10100
35 Dong S, Wang L, Du B, et al. ChangeCLIP: remote sensing change detection with multimodal vision-language representation learning. ISPRS J Photogrammetry Remote Sens, 2024, 208: 53–69
36 Zhan Y, Xiong Z, Yuan Y. SkyEyeGPT: unifying remote sensing vision-language tasks via instruction tuning with large language model. ISPRS J Photogrammetry Remote Sens, 2025, 221: 64–77
37 Li Y, Xu W, Li G, et al. Unirs: unifying multi-temporal remote sensing tasks through vision language models. 2024. ArXiv:2412.20742
38 Wu Q, Bansal G, Zhang J, et al. Autogen: enabling next-gen LLM applications via multi-agent conversation. 2023. ArXiv:2308.08155
39 Shabbir A, Munir M A, Dudhane A, et al. Thinkgeo: evaluating tool-augmented agents for remote sensing tasks. 2025. ArXiv:2505.23752
40 Feng P, Lv Z, Ye J, et al. Earth-agent: unlocking the full landscape of Earth observation with agents. 2025. ArXiv:2509.23141
41 Kenneweg T, Kenneweg P, Hammer B. Retrieval augmented generation systems: automatic dataset creation, evaluation and Boolean agent setup. 2024. ArXiv:2403.00820
42 Gao L, Ma X, Lin J, et al. Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023. 1762–1777
43 Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: a survey. 2023. ArXiv:2312.10997
44 Chan C M, Xu C, Yuan R, et al. Rq-rag: learning to refine queries for retrieval augmented generation. 2024. ArXiv:2404.00610

45 Tang Y, Yang Y. Multihop-RAG: benchmarking retrieval-augmented generation for multi-hop queries. 2024. ArXiv:2401.15391
46 Edge D, Trinh H, Cheng N, et al. From local to global: a graph rag approach to query-focused summarization. 2024. ArXiv:2404.16130
47 Zhu X, Xie Y, Liu Y, et al. Knowledge graph-guided retrieval augmented generation. 2025. ArXiv:2502.06864
48 Guo Z, Xia L, Yu Y, et al. Lightrag: simple and fast retrieval-augmented generation. 2024. ArXiv:2410.05779
49 Chase H. LangChain: building applications with LLMs through composability. Version 0.1. 2022. https://github.com/langchain-ai/langchain
50 Qin Y, Hu S, Lin Y, et al. Tool learning with foundation models. 2023. ArXiv:2304.08354
51 Shapira N, Levy M, Alavi S H, et al. Clever Hans or neural theory of mind? Stress testing social reasoning in large language models. 2023. ArXiv:2305.14763
52 Chen J, Xiao S, Zhang P, et al. Bge m3-embedding: multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. 2024. ArXiv:2402.03216
53 Jégou H, Douze M, Johnson J, et al. Faiss: similarity search and clustering of dense vectors library. Astrophysics Source Code Library, 2022, ascl:2210.024. http://ascl.net/2210.024
54 Ultralytics. YOLOv8: state-of-the-art real-time object detection. Version 8.0. 2023. https://github.com/ultralytics/ultralytics
55 Xia G S, Bai X, Ding J, et al. Dota: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3974–3983
56 Liu H, Li C, Li Y, et al. Improved baselines with visual instruction tuning. 2023. ArXiv:2310.03744
57 Chen J, Zhu D, Shen X, et al. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. 2023. ArXiv:2310.09478
58 Dai W, Li J, Li D, et al. Instructblip: towards general-purpose vision-language models with instruction tuning. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
59 Ye Q, Xu H, Xu G, et al. mplug-owl: modularization empowers large language models with multimodality. 2023. ArXiv:2304.14178
60 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: transformers for image recognition at scale. 2020. ArXiv:2010.11929
61 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
62 Li Y, Zhu Z, Yu J G, et al. Learning deep cross-modal embedding networks for zero-shot remote sensing image scene classification. IEEE Trans Geosci Remote Sens, 2021, 59: 10590–10603
63 Xia G S, Hu J, Hu F, et al. AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans Geosci Remote Sens, 2017, 55: 3965–3981
64 Yang Y, Newsam S. Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010. 270–279
65 Qian H, Zhang P, Liu Z, et al. Memorag: moving towards next-gen rag via memory-inspired knowledge discovery. 2024. ArXiv:2409.05591