RSMeM: Knowledge-Enhanced Memory Evolution for Remote Sensing Agents with Systematic Evaluation

Bingxian Wu[1,3*†], **Yu Zhang**[2†], **Zonghao Guo**[2‡], **Tang Liu**[1],
Chen Qian[1,3], **Yuxiang Lu**[4], **Xingbo Du**[6], **Yanghao Li**[2],
Yidan Zhang[3,5], **Chi Chen**[2], **Ling Yao**[3], **Chenghu Zhou**[1], **Maosong Sun**[2]
[1]Institute of Geographic Sciences and Natural Resources Research, CAS [2]Tsinghua University
[3]University of Chinese Academy of Sciences [4]China University of Geosciences, Beijing
[5]Aerospace Information Research Institute, CAS [6]Shanghai Jiao Tong University
wubingxian24@mails.ucas.edu.cn, guozonghao96@outlook.com

Abstract

Geoscience research requires complex analysis and domain expertise, with remote sensing (RS) observations as a key foundation. However, existing RS agents built on general-purpose LLMs remain largely domain-agnostic, resulting in brittle and error-prone workflows. Moreover, these failures are seldom consolidated into a reusable experience for subsequent analyses. To address this issue, we introduce RSMeM, a knowledge-enhanced memory evolution mechanism that bootstraps RS agents with pre-distilled domain knowledge and iteratively integrates online experience for robust multi-step tool execution. RSMeM is composed of two components: (i) Hierarchical Knowledge Grounding, which performs taxonomy-aware retrieval over a hierarchical domain corpus to guide planning and tool selection; and (ii) Failure-Aware Experience Refinement, which distills failure-annotated tool-use traces into reusable constraints for next-round tool execution. By iteratively employing these two processes, RS agents can evolve to absorb task-level domain knowledge and effectively translate it into instance-level execution experience. Extensive experiments on EarthBench demonstrate that RSMeM consistently improves tool-use performance and end-to-end answer across a diverse set of LLM backbones. Notably, RSMeM achieves a 6% accuracy improvement on DeepSeek-V3.2 with less than 1% additional experience tokens, demonstrating the strong knowledge density of our distilled experience. All codes and models will be released to support reproducible research.

1 Introduction

Advances in large language models (LLMs) and tool-augmented agents have demonstrated promising capabilities in task planning, reasoning, and external tool use (Su et al., 2024; Tang et al., 2025). In parallel, geoscience research has witnessed growing interest in deploying RS agents (Xu et al., 2025a; Shabbir et al., 2025; Feng et al., 2025) for in-depth analysis of remote sensing imagery.

One line of RS agents (Shabbir et al., 2025; Feng et al., 2025) leverages general-purpose LLMs with ReAct-style tool-use to perform multi-step analysis. Yet, because these agents are largely domain-agnostic, they often struggle with domain-specific concepts (e.g., NDVI, BSI or land-surface temperature interactions), which makes their tool-use trajectories brittle and prone to cascading errors, such as premature termination or incorrect tool invocation, shown in Fig. 1(a). The other paradigm

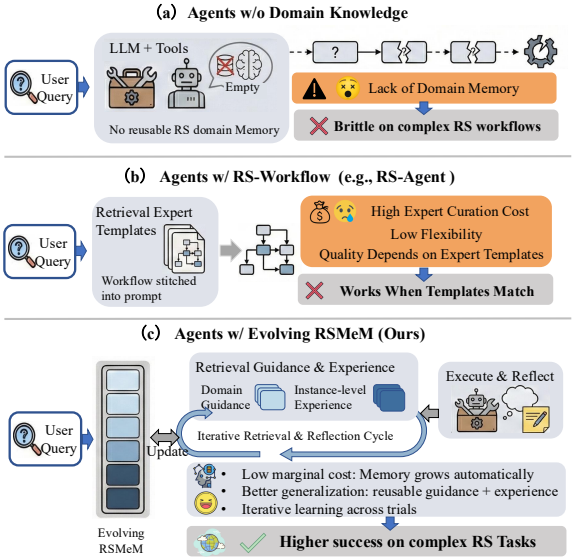

Figure 1: Comparison of RS agent architectures. (a) Agents w/o Domain Knowledge lack reusable memory, leading to brittle workflows. (b) Agents w/ RS-Workflow rely on static expert templates, which are inflexible and costly to curate. (c) Agent w/ Evolving RSMeM (Ours) integrates hierarchical grounding with failure-aware reflection, enabling iterative knowledge evolution and robust execution for complex RS tasks.

*Work done during his internship at AI9Stars.
†Equal contribution
‡Corresponding authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32899–32915
July 2-7, 2026

adopts semi-automated, expert-authored solution templates or workflow programs (Xu et al., 2025a; Chen et al., 2025) to guide LLMs planning and tool-calling, as shown in Fig. 1(b). While effective for narrowly scoped tasks, such workflow-centric designs are labor-intensive to construct and maintain. Moreover, these task-specific procedural details largely serve as external references, rather than enabling the agent to internalize corrections from execution failures and improve its behavior over time.

To enable agents to learn from mistakes on concrete tasks, prior work has explored reflection-driven improvements (Shinn et al., 2023; Zhang et al., 2025b; Fang et al., 2025; Zhang et al., 2025a). However, in geoscience settings, reflection alone is limited, as agents without domain knowledge often misdiagnose failures and fail to turn errors into procedural fixes.

To tackle this challenge, we present RSMeM, a memory-evolution framework that seeds RS agents with pre-distilled domain knowledge and continually incorporates online experience to enable robust multi-step tool execution, shown in Fig. 1(c). Following prior task taxonomies in geoscience (Kucharczyk and Hugenholtz, 2021; Ma et al., 2024; Yuan et al., 2020; Ma and Mei, 2021), we construct a set of geoscience knowledge corpus that covers 3 application domains, 12 subdomains and 64 task formulations. We further refine the corpus with manual, professional feedback to form a distilled knowledge base for bootstrapping RSMeM. RSMeM comprises two processes. **(i) Hierarchical Knowledge Grounding** conducts taxonomy-aware retrieval over a hierarchical corpus by encoding each knowledge item into retrievable units (keywords, task definitions, and semantic embeddings) and conditioning retrieval on the user query to inform LLM planning and tool selection. **(ii) Failure-Aware Experience Refinement** leverages a failure-critique environment to identify erroneous tool-use trajectories and distill them into reusable constraints that are stored as experience memory. By iterating between knowledge grounding and experience refinement, RSMeM injects both retrieved knowledge and failure-derived constraints into subsequent planning and tool invocation, enabling RS agents to progressively acquire instance-level execution competence from task-level domain knowledge. Extensive experiments on EarthBench (Feng et al., 2025) demonstrate that RSMeM consistently improves tool-use performance and end-to-end answer quality across diverse LLM backbones, including DeepSeek V3.2 (DeepSeek-AI et al., 2025), Kimi-K2 (Team et al., 2025), Qwen3-32B (Yang et al., 2025) and Qwen3-8B (Yang et al., 2025). Notably, RSMeM yields a 6.07% absolute accuracy improvement on DeepSeek-V3.2 with less than 1% additional experience tokens, whose efficiency arises from its ability to distill raw execution traces into high-density, reusable geoscience expertise. Rather than relying on resource-intensive context expansion, RSMeM prioritizes the synthesis of optimized workflow experiences, ensuring a superior performance-to-cost ratio for complex earth observation tasks.

We further observe that stronger backbones benefit more from RSMeM, as they exhibit greater capacity to summarize domain knowledge and diagnose execution errors, surfacing domain-level corrections during iterative refinement.

We summarize our contributions as follows:

- We propose a taxonomy-driven construction and retrieval scheme for structured geoscience knowledge, enabling RS agents to ground their reasoning in domain taxonomy thereby perform more reliable geoscience inference.
- We propose a knowledge-enhanced memory evolution framework to iteratively couple knowledge grounding with failure-aware experience refinement, enabling RS agents to transform domain knowledge into reusable execution experience.
- We propose RSMeM, a plug-and-play framework that can be seamlessly applied to diverse LLM backbones and consistently improves both tool-use and end-to-end performance on challenging remote sensing tasks.

2 Related Works

Remote Sensing Agents. Recently, agent-based Earth observation research has gained significant attention (Luo et al., 2025; Kim et al., 2025). Tool-augmented agents excel in tasks such as code generation (Huang et al., 2024), multi-app collaboration (Wang et al., 2024), and video understanding (Fan et al., 2024); however, their application in Earth observation remains in its infancy (Kao et al., 2025a). Early works, such as Change-Agent (Liu et al., 2024), focused on bi-temporal remote sensing change detection. In

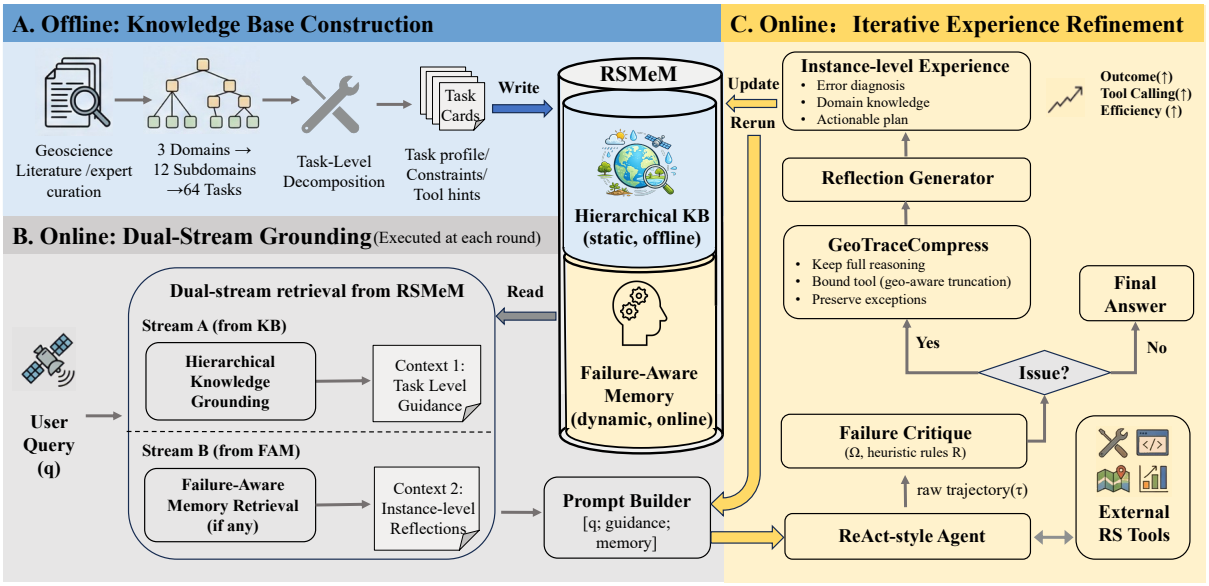

Figure 2: Overview of the RSMeM framework. The architecture consists of three main stages: (1) Offline Knowledge Base Construction: Systematically distilling geoscience literature into a hierarchical domain knowledge base and task-specific cards. (2) Online Task Solving via Dual-Stream Retrieval: The RS-Agent processes user queries and RS observations by retrieving task-level guidance through Hierarchical Knowledge Grounding (Stream A) and instance-level reflections from the Failure-Aware Memory (FAM) (Stream B). (3) Iterative Self-improvement: Based on the tool-use trace, a reflection generator synthesizes feedback to update the FAM, enabling the agent to evolve its experience and achieve higher accuracy and more reliable trajectories in subsequent tasks.

contrast, RS-ChatGPT (Guo et al., 2024) and RS-Agent (Xu et al., 2025a) integrated large language models with remote sensing tools for scene classification and object detection. Recent methods like ThinkGeo (Shabbir et al., 2025) introduced geospatial computing workflows, and UnivEarth (Kao et al., 2025b) incorporated geo-environmental encoding for spectral analysis; however, high failure rates limit their practical use. To overcome the overreliance on RGB data and underutilization of spectral information, Earth-Agent (Feng et al., 2025) integrates multispectral and hyperspectral data into a unified multimodal pipeline, supporting complex tasks like parameter inversion and time-series analysis, along with the Earth-Bench benchmark for evaluation. For structured geospatial workflows, the HTAM framework (Li et al., 2025) employs a task-aware hierarchical architecture to enable multi-step problem decomposition, instantiated as EarthA-gent and evaluated through GeoPlan-Bench with metrics for tool selection and logical consistency. The CangLing-KnowFlow (Chen et al., 2025) introduces a Process Knowledge Base (PKB) and an evolutionary memory module to mitigate hallucinations.

Memory-Augmented Experience Learning. Memory enables language agents to retain and reuse information across interactions, substantially improving adaptability and task performance (Fang et al., 2025; Xu et al., 2025b). Prior work has explored various memory designs inspired by human cognition, aiming to support coherence, personalization, and continual learning in LLM-based agents (Chhikara et al., 2025). Existing approaches can be broadly grouped into end-to-end memory systems, external memory mechanisms, and hierarchical memory architectures (Hu et al., 2025; Tang et al., 2025). Across these paradigms, experiences are typically stored as textual summaries or embedding representations and retrieved via semantic similarity, with memory update and forgetting strategies used to maintain relevance (Shinn et al., 2023). Memory is closely tied to experience learning, where agents enhance decision-making through repeated interactions with the environment (Tan et al., 2025; Fang et al., 2025; Zhang et al., 2026). While experience-driven learning has been explored through reinforcement and imitation learning (Cai et al., 2025; Sun et al., 2024), general-purpose agents lack RS-specific frameworks that unify offline expertise with online instance-level experiences. We bridge this gap with RSMeM, which distills execution errors into

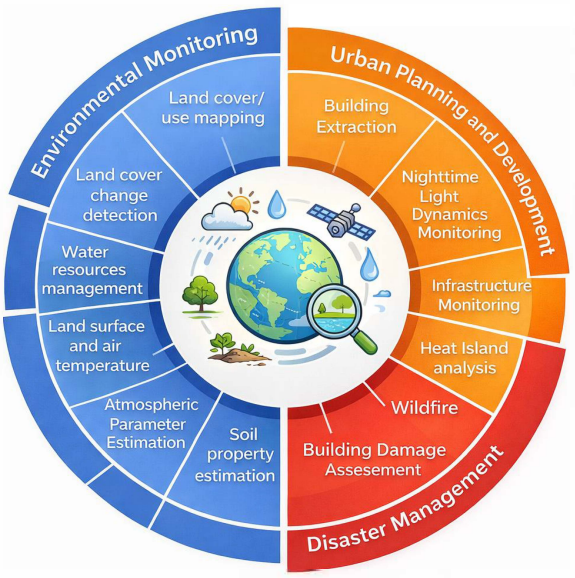

Figure 3: Hierarchical structure of the geoscience knowledge base. The outer ring corresponds to high-level domains, and the inner ring captures fine-grained sub-domains.

actionable constraints to ground reasoning in a progressively refined, domain-specific memory.

3 Methodology

As depicted in Fig. 2, RSMeM augments general LLMs with geoscientific expertise by coupling pre-distilled geoscience knowledge with iterative experience refinement. This iterative experience evolution follows a dual-stream mechanism that combines hierarchical knowledge grounding with experience refinement, which are operationalized via the prompt templates in Appendix A.1.

3.1 Hierarchical Knowledge Base Construction

To provide a robust foundation for grounding, we construct a hierarchical geoscience knowledge base (KB) curated by domain experts. This expert-driven design ensures analytical validity and terminological consistency across complex RS tasks.

Hierarchical Taxonomy The KB follows a three-level hierarchy consisting of 3 high-level *Geoscience Domains*, which are further decomposed into 12 *Sub-domains* and 64 *Atomic Analytical Tasks*, as shown in Fig. 3. This structure allows the agent to navigate from broad application areas to specific operational steps.

Task Card Construction Each atomic task is encapsulated in a **Task Card**, represented as a structured triplet $s_i = \{T_i, D_i, S_i\}$. Here, T_i denotes the task title, D_i provides a domain definition, and S_i specifies the standardized suggestion or hints required for execution.

We provide comprehensive task card samples and details on the expert-driven construction process in Appendix A.2 and A.3, respectively.

3.2 Dual-Stream Online Grounding

During inference, the agent assembles its input prompt from two complementary retrieval streams, appended after the system prompt and the user query:

Stream A: Hierarchical Knowledge Grounding (HKG). Given a user query q, we retrieve the most relevant Task Card via a hybrid scoring function that combines neural similarity with taxonomy-aware lexical signals:

$$\begin{aligned}\mathrm{Score}(q, s_i) = \cos(\mathbf{v}_q, \mathbf{v}_i) + \alpha \cdot \mathbb{1}_{\mathrm{T}}(q, T_i) \\ + \beta \cdot \textstyle\sum_{j=1}^{m} \mathbb{1}_{\mathrm{K}}(q, k_j) \\ + \gamma \cdot \mathbb{1}_{\mathrm{C}}(q, S_i),\end{aligned} \tag{1}$$

where $\mathbf{v}_q$ and $\mathbf{v}_i$ are sentence-transformer embeddings of the query and the contextualized Task Card, respectively. The indicator terms $\mathbb{1}_{\mathrm{T}}$, $\mathbb{1}_{\mathrm{K}}$, and $\mathbb{1}_{\mathrm{C}}$ fire when query tokens lexically overlap with the task title T_i, curated domain keywords $\{k_j\}_{j=1}^{m}$, and the task-card content S_i, respectively. These lexical bonuses (α=0.2, β=0.08, γ=0.1) compensate for cases where geoscience terminology (e.g., "TVDI", "emissivity") is underrepresented in the embedding space. We return the top-1 candidate exceeding a minimum threshold τ=0.2.

Stream B: Failure-Aware Memory (FAM) Retrieval. For each instance i, the agent retrieves its most recent reflection from the FAM via a deterministic instance-indexed lookup:

$$c_i^{(t)} = \mathrm{Latest}\big(\mathcal{M}_i^{(t-1)}\big), \tag{2}$$

where $\mathcal{M}_i^{(t-1)}$ is the reflection memory accumulated for instance i up to round $t-1$, and $\mathrm{Latest}(\cdot)$ selects the most recent entry. This design is deliberate: by binding reflections to the specific execution trace that produced them, we ensure that corrective guidance is maximally precise and free from interference by unrelated failure patterns. Retrieved reflections provide negative constraints and historical fixes distilled from previous tool-use traces, preventing the repetition of execution errors (detailed in Appendix A.4).

Table 1: Tool-use Performance Comparison on Trajectory and Outcome. Best results per backbone are **bolded**; "↑" indicates improvement over EarthAgent (R@3).

Backbone	Method	Cfg.	Tool Calling Accuracy				Outcome
			AO	IO	EM	WS	Acc.
Proprietary Models							
GPT-5	EarthAgent	R@1	69.11	58.25	45.57	–	65.59
Gemini-2.5	EarthAgent	R@1	57.96	45.44	31.86	–	54.66
GPT-4o	EarthAgent	R@1	65.73	50.70	46.17	–	45.34
Open-Source Models							
Qwen3-Max	EarthAgent	R@1	69.56	53.28	37.02	–	50.20
LLaMA-4	EarthAgent	R@1	16.51	2.45	1.70	–	44.94
InternVL-3.5	EarthAgent	R@1	8.83	3.87	2.02	–	26.72
Main Evaluation							
Qwen3-8B	EarthAgent	R@1	43.91	16.68	3.79	0.0408	31.98
	EarthAgent	R@3	48.34	21.82	4.14	0.0455	39.27
	RSMeM	R@3	**49.50**	**23.18**	**5.82**	**0.0600**	**42.11**↑
Qwen3-32B	EarthAgent	R@1	38.66	14.75	5.42	0.0569	38.87
	EarthAgent	R@3	46.68	26.80	8.81	0.0930	44.94
	RSMeM	R@3	**46.94**	**30.75**	**13.88**	**0.1469**	**48.58**↑
Kimi-K2	EarthAgent	R@1	72.79	**48.50**	28.72	0.3975	36.44
	EarthAgent	R@3	**73.39**	48.47	29.12	0.3887	43.32
	RSMeM	R@3	70.88	47.41	**30.81**	**0.3872**	**48.99**↑
DeepSeek-V3.2	EarthAgent	R@1	77.27	55.02	26.56	0.4275	48.18
	EarthAgent	R@3	78.02	55.47	**27.25**	**0.4342**	51.82
	RSMeM	R@3	**79.20**	**55.49**	27.01	0.4231	**57.89**↑

3.3 Iterative Experience Refinement

The core of RSMeM is a self-improvement loop that progressively refines task-level knowledge into precise instance-level experience.

Failure Critique. A symbolic critic Ω (Fig. 2) applies a set of heuristic rules $\mathcal{R}$ to the agent's tool-use trajectory $a_i^{(t)}$ to determine whether the attempt should be considered failed:

$$\text{Issue}(a_i^{(t)}) = \mathbb{I}\big(\exists\, \rho \in \mathcal{R} \text{ s.t. } \rho(a_i^{(t)})\big), \quad (3)$$

where each $\rho \in \mathcal{R}$ is a pattern-matching rule targeting a specific failure mode: (i) missing or malformed answer tags, (ii) empty responses, (iii) invalid option letters, (iv) refusal or incompleteness signals, and (v) hedging language indicative of guessing. This lightweight symbolic critique operates without access to ground-truth labels.

GeoTraceCompress. Traces of RS agents often contain verbose geospatial I/O (e.g., lengthy file-path lists, large metadata dictionaries) that would overwhelm the reflection prompt. We apply a *type-aware* compression strategy: list-valued outputs retain only the first three and last two items; dictionary outputs are reduced to two representative key-value pairs; error messages receive an elevated budget since they carry critical diagnostic signals. The agent's reasoning text is preserved in full. This compression reduces trajectory length by an order of magnitude while retaining the information most relevant for failure diagnosis.

Experience Distillation. For each failed instance, a **Reflection Generator** R (instantiated as an LLM call with the same backbone) takes the original query q_i, the retrieved HKG context, and the compressed trajectory as input, and produces a structured reflection $r_i^{(t)}$—a 3–5 sentence diagnosis identifying the failure cause and proposing a corrective plan. The FAM is then updated:

$$\mathcal{M}_i^{(t)} = \mathcal{M}_i^{(t-1)} \oplus r_i^{(t)}, \quad (4)$$

where $\oplus$ denotes memory append. Once stored, this reflection becomes available to Stream B for subsequent attempts, closing the execution–reflection–adaptation loop.

4 Experiment

4.1 Experimental Setup

Datasets: To evaluate our method, we employ Earth-Bench (Feng et al., 2025), a specialized

Table 2: Tool-use Performance Comparison (Efficiency). EarthAgent vs. memory-enhanced RSMeM. Best results per backbone are **bolded**; "↑" = improvement over EarthAgent (R@3).

Backbone	Method	Cfg.	Efficiency		
			TRI	Tok. (%)	ED
Qwen3-8B	EarthAgent	R@1	1.50	4.12	–
	EarthAgent	R@3	**1.20**	4.97	–
	RSMeM	R@3	1.26	**5.30**	41.75
Qwen3-32B	EarthAgent	R@1	**0.60**	4.99	–
	EarthAgent	R@3	0.70	5.58	–
	RSMeM	R@3	0.65	**6.07**	47.90
Kimi-K2	EarthAgent	R@1	1.39	4.55	–
	EarthAgent	R@3	1.33	5.25	–
	RSMeM	R@3	**1.20**	**6.02**	48.81
DeepSeek-V3.2	EarthAgent	R@1	2.03	5.99	–
	EarthAgent	R@3	1.92	6.23	–
	RSMeM	R@3	**1.86**	**6.97**	57.69

benchmark designed for tool-augmented Earth Observation agents in the context of real-world geoscientific analysis. This benchmark comprises 248 expert-annotated instances spanning three distinct modalities: RGB, Spectrum, and Earth Products.

Evaluated Models: We evaluate RSMeM with four representative LLM backbones—DeepSeek-V3.2 (DeepSeek-AI et al., 2025), Kimi-K2 (Team et al., 2025), and Qwen3 (32B / 8B) (Yang et al., 2025)—and compare against EarthAgent under the same protocol. All results are aggregated over three independent rounds (R@3), where instances that remain unsolved are re-run in subsequent rounds, and the final accuracy is computed by backward overwriting with the latest successful attempt.

Reflexion Baseline: We employ Reflexion (Shinn et al., 2023) as a gradient-free baseline that iteratively refines performance by storing verbal self-reflections in episodic memory. To bridge the gap between its original Docstore design and our tool-augmented EO setting, we adapt its feedback loop to our environment's specific failure modes and structured output requirements. Detailed implementation and prompt templates are provided in Appendix A.7.

Implementation Details: Following EarthAgent's (Feng et al., 2025) configuration, we utilize 104 tools but substitute the MSCN model (Ma et al., 2025) with the SiamCRNN model (Chen et al., 2020). Experiments are conducted on a remote sensing knowledge base across 3 application domains, 12 subdomains, and 64 tasks. We do not expose any ground-truth tool steps or answers to the agent during inference/refinement. Labels are reserved for evaluation only.

4.2 Evaluation Metrics

We evaluate RSMeM across three systematic dimensions: (i) Tool Calling Accuracy: Any-Order (AO), In-Order (IO), and Exact-Match (EM) for sequence matching, plus Wise-Score (WS) for prefix-sensitive trajectory and parameter precision; (ii) Outcome: End-to-end Accuracy (Acc); (iii) Efficiency: Trajectory Redundancy Index (TRI), Token Efficiency (Tok.), and Experience Density (ED) to quantify the cost-performance trade-off. Formal definitions are provided in Appendix A.5.

5 Experimental Results

5.1 Main Results and Analysis

We evaluate RSMeM in three dimensions (Tab. 1, 2): tool-use correctness, outcome accuracy and efficiency.

5.1.1 Tool-use Correctness across Granularities

As shown in Tab. 1, RSMeM consistently improves tool-use performance across AO, IO, EM, and WS, indicating gains at multiple levels of trajectory fidelity. The improvement in AO suggests that RSMeM enhances global tool coverage by grounding planning in taxonomy-aware domain knowledge, while gains in IO reflect improved sequencing consistency. More importantly, the consistent increase in EM demonstrates that RSMeM reduces compounding errors across steps, enabling the agent to complete entire tool trajectories without deviation. The rise in WS further confirms that these improvements are not limited to tool identity or order, but extend to parameter-level correctness under a prefix-sensitive evaluation. Since WS only rewards steps that follow a fully correct prefix, the observed gains imply that RSMeM stabilizes early-stage decisions, which are critical for downstream execution. Tab. 2 evaluates the efficiency of RSMeM across TRI, token overhead, and ED. Compared to EarthAgent, RSMeM introduces a marginal increase in token consumption—for instance, rising from 6.23% to 6.97% on DeepSeek-V3.2, attributable to the structured reflection and memory update stages. However, this overhead remains strictly constrained across all backbones.

RSMeM consistently achieves high ED scores, indicating an effective balance between perfor-

Table 3: Ablation study of **HKG** (Hierarchical Knowledge Grounding) and **FAR** (Failure-Aware Experience Refinement) across representative backbones. RSMeM denotes the full integrated framework. **Cfg.** indicates the evaluation configuration (e.g., number of rounds). All metrics are aligned with the main results (Tab. 1). Bold indicates the best performance within each backbone group.

Variant	Cfg.	Comp.		Tool Calling Accuracy				Outcome	Efficiency		
		HKG	FAR	AO	IO	EM	WS	Acc.	TRI	Tok. (%)	ED
Backbone: Qwen3-8B											
Baseline	R@1	–	–	43.91	16.68	3.79	0.0408	31.98	1.50	4.12	–
w/ HKG	R@1	✓	–	44.87	17.33	4.15	0.0461	34.41	1.48	4.42	–
w/ FAR	R@3	–	✓	48.36	23.35	**6.55**	**0.0763**	39.68	1.19	4.95	39.32
RSMeM	R@3	✓	✓	**49.50**	**23.18**	5.82	0.0600	**42.11**	**1.26**	**5.30**	**41.75**
Backbone: DeepSeek-V3. 2											
Baseline	R@1	–	–	77.27	55.02	26.56	0.4275	48.18	2.03	5.99	–
w/ HKG	R@1	✓	–	**79.50**	**56.51**	25.85	**0.4279**	51.82	1.99	6.44	–
w/ FAR	R@3	–	✓	76.94	54.73	26.75	0.4150	53.44	1.92	6.45	53.24
RSMeM	R@3	✓	✓	79.20	55.49	**27.01**	0.4231	**57.89**	**1.86**	**6.97**	**57.89**

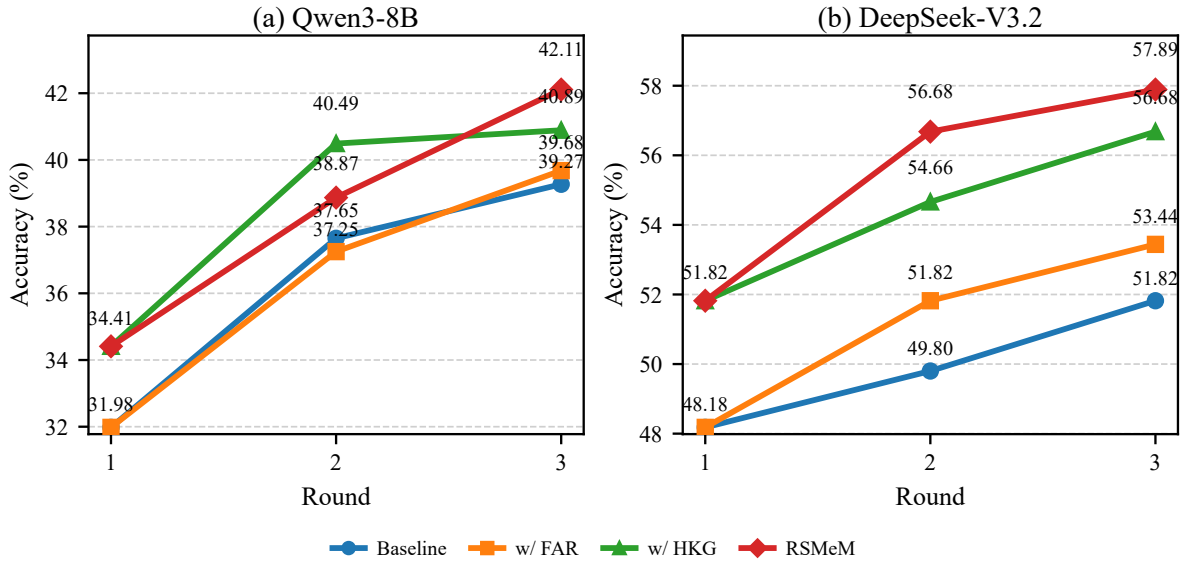

Figure 4: Ablation of RSMeM Components over Iterative Rounds (Accuracy across Models).

mance gains and computational cost. As backbone capacity increases, experience utility scales accordingly, suggesting that stronger models convert structured HKG grounding into more compact and actionable knowledge rather than redundant linguistic traces. Meanwhile, TRI values remain close to 1.0, showing that RSMeM avoids unnecessarily long or repetitive trajectories and instead concentrates computation on high-impact reflection, enabling accuracy improvements through refined planning rather than brute-force exploration.

We additionally compare RSMeM with the reflection-only baseline Reflexion. As shown in Appendix A.7 (Fig. 12), RSMeM achieves consistent gains over Reflexion across all four backbones on both outcome accuracy and experience density, suggesting improvements beyond generic linguistic reflection. To further isolate the contribution of structured grounding from generic retrieval, we compare RSMeM against flat embedding retrieval and web-based augmentation (Appendix A.11); RSMeM outperforms both by a substantial margin (+16.33 pp over Simple RAG in accuracy), confirming that taxonomy-aware domain knowledge is fundamentally more effective than unstructured retrieval for geoscientific tasks. A per-domain breakdown (Appendix A.12) reveals that RSMeM benefits most in domains with standardized multi-step workflows (Disaster Management: +10.25 pp), while gains are smaller in domains where residual errors stem

Table 4: Memory evolution on DeepSeek-V3.2. The baseline refers to the framework equipped with GRF

Setting	EM (%)	Tok. (%)	ED	Acc. (%)	Δ
GRF-only	25.85	0.0644	–	51.82	–
Evolutionary Stages					
Stage I: V1	26.19	0.0645	55.72	55.87	↑ 4.05
Stage II: V2	**27.01**	**0.0697**	**57.69**	**57.89**	↑ **6.07**

from upstream perception limits.

5.1.2 Component ablation (HKG, FAR)

Tab. 3 demonstrates that HKG and FAR provide complementary benefits to the tool-use process. HKG primarily optimizes global planning, increasing AO from 77.27% to 79.50% and IO from 55.02% to 56.51% on DeepSeek-V3.2. Conversely, FAR significantly enhances trajectory robustness, as evidenced by the WS improvement from 0.0408 to 0.0763 on Qwen3-8B. While individual components like HKG or FAR alone yield partial gains, their growth often stabilizes early or plateaus as shown in Fig. 4.

The full RSMeM framework achieves peak performance across all models, reaching outcome accuracies of 42.11% and 57.89% on Qwen3-8B and DeepSeek-V3.2, respectively. Despite a marginal increase in token usage, RSMeM consistently attains the highest ED, reaching 41.75 on Qwen3-8B and 57.69 on DeepSeek-V3.2—validating that the integration of both components ensures that performance gains are driven by high-density corrective knowledge rather than redundant computation.

5.1.3 Memory Evolution and Iterative Effects

Tab. 4 further analyzes the evolutionary trajectory of memory refinement. Starting from a GRF-only baseline, EM increases from 25.85% to 27.01%, with outcome accuracy improving by +6.07 points after two evolution stages. Notably, the ED rises from 55.72 to 57.69. This upward trend suggests that memory evolution enhances "information purity," where subsequent stages distill coarse failure patterns into more concentrated, high-utility corrective knowledge rather than merely accumulating tokens. Fig. 4 visualizes this trend across iterative rounds. RSMeM exhibits steeper and more stable accuracy improvements than both the baseline and single-component variants on Qwen3-8B and DeepSeek-V3.2. While DeepSeek-V3.2 shows a rapid performance surge in the second round, followed by a marginal plateau, the integration of HKG and FAR enables improvements to persist where individual components (e.g., FAR-only) often stagnate. This indicates that structured grounding and failure-aware refinement reinforce each other, maintaining a high level of Experience Density even as accuracy gains stabilize. Overall, these results demonstrate that RSMeM enhances tool use not by increasing trajectory length or trial-and-error, but by stabilizing early planning and converting execution failures into reusable experiences. This mechanism ensures a favorable effectiveness–efficiency trade-off, achieving consistent gains through superior experience density.

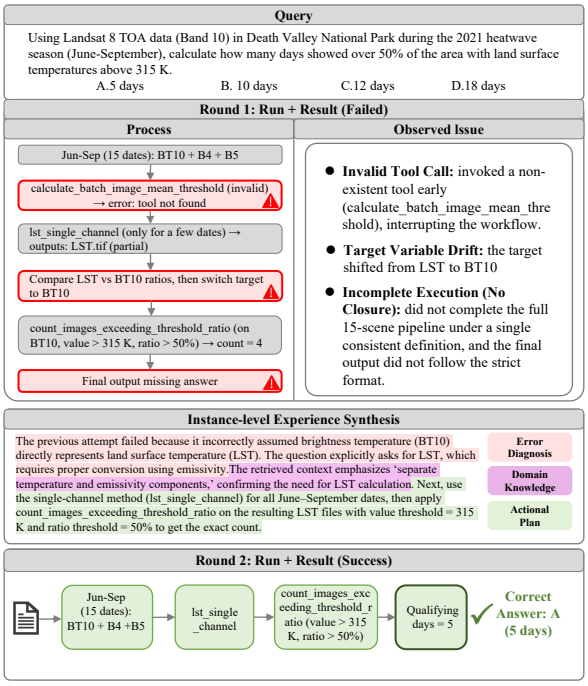

Figure 5: **Failure-aware memory evolution in Landsat thermal time-series analysis.** In Round1, the agent fails due to an invalid tool call and a semantic mismatch between BT10 (TOA brightness temperature from Landsat Band10) and LST (emissivity-corrected land surface temperature). RSMeM (DeepSeek-V3.2) grounds domain knowledge while diagnosing execution errors to distill instance-level experience, which subsequently guides a corrected end-to-end LST pipeline and yields the correct count of five days.

5.2 Case Study

Fig. 5 illustrates a representative RS failure pattern in which errors are not isolated but instead cascade through the workflow. An early tool misuse, combined with a subtle semantic mismatch (e.g., treating BT10 as LST), jointly derails downstream steps, such that additional computation fails to recover correctness. This example highlights a

key limitation of generic tool-using agents in geoscience tasks: they may execute syntactically plausible operations while violating the *target variable contract* implicitly required by the task.

RSMeM improves robustness by converting such failures into persistent and reusable guardrails. Specifically, the experience refinement mechanism distills (i) a semantic constraint that enforces the correct variable definition (i.e., LST must be emissivity-aware), and (ii) a procedural constraint that maintains a consistent end-to-end pipeline with explicit completion requirements. The resulting behavioral change is not merely a repeated attempt, but a policy-level update that reduces the likelihood of recurring conceptual errors in subsequent thermal workflows. Overall, this case suggests that RSMeM's performance gains primarily stem from stabilizing early commitments—namely, variable semantics and tool validity—thereby preventing error propagation and improving end-to-end reliability with minimal additional overhead.

6 Conclusion

We present RSMeM, a memory-evolution mechanism designed to bridge the gap between general-purpose LLMs and domain-specific geoscience requirements. By integrating Hierarchical Knowledge Grounding (HKG) and Failure-Aware Experience Refinement (FAR), RSMeM enables agents to bootstrap from pre-distilled expertise and iteratively optimize execution logic through online experience. Evaluations on EarthBench demonstrate that RSMeM consistently enhances tool-use accuracy and end-to-end performance across various LLM backbones. Notably, the framework achieves significant gains with minimal reflection overhead, establishing structured memory evolution as an efficient paradigm for building expert-level RS agents.

Limitations

Despite its effectiveness, our work has several limitations. First, the performance of RSMeM in the initial stages is partially dependent on the quality and coverage of the hierarchical domain corpus; a sparse initial knowledge base may lead to a longer "cold-start" period for memory evolution. Second, while we evaluated the framework on the comprehensive EarthBench, its generalization to highly specialized or real-time streaming remote sensing tasks remains to be further explored. Lastly, the current experience refinement process primarily focuses on tool-use failures; incorporating successful but sub-optimal traces could potentially lead to even more nuanced strategic optimization, which we leave for future work.

Acknowledgments

We thank the anonymous reviewers and the area chair for their constructive comments. This work was supported in part by the National Natural Science Foundation of China under Grants 42501455 and 42471386, by the China Postdoctoral Science Foundation under Grant 2025M780333, and by the AI9Stars community.

References

Yuxuan Cai, Yipeng Hao, Jie Zhou, Hang Yan, Zhikai Lei, Rui Zhen, Zhenhua Han, Yutao Yang, Junsong Li, Qianjun Pan, Tianyu Huai, Qin Chen, Xin Li, Kai Chen, Bo Zhang, Xipeng Qiu, and Liang He. 2025. Building self-evolving agents via experience-driven lifelong learning: A framework and benchmark. *Preprint*, arXiv:2508.19005.

Hongruixuan Chen, Chen Wu, Bo Du, Liangpei Zhang, and Le Wang. 2020. Change detection in multisource vhr images via deep siamese convolutional multiple-layers recurrent neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2848–2864.

Zhengchao Chen, Haoran Wang, Jing Yao, Pedram Ghamisi, Jun Zhou, Peter M. Atkinson, and Bing Zhang. 2025. Cangling-knowflow: A unified knowledge-and-flow-fused agent for comprehensive remote sensing applications. *Preprint*, arXiv:2512.15231.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *Preprint*, arXiv:2504.19413.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. *Preprint*, arXiv:2403.11481.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2025. Memp: Exploring agent procedural memory. *Preprint*, arXiv:2508.06433.

Peilin Feng, Zhutao Lv, Junyan Ye, Xiaolei Wang, Xinjie Huo, Jinhua Yu, Wanghan Xu, Wenlong Zhang, Lei Bai, Conghui He, and Weijia Li. 2025. Earth-agent: Unlocking the full landscape of earth observation with agents. *Preprint*, arXiv:2509.23141.

Haonan Guo, Xin Su, Chen Wu, Bo Du, Liangpei Zhang, and Deren Li. 2024. Remote sensing chatgpt: Solving remote sensing tasks with chatgpt and visual models. *Preprint*, arXiv:2401.09083.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2025. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32779–32798, Vienna, Austria. Association for Computational Linguistics.

Dong Huang, Jie M. Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. 2024. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *Preprint*, arXiv:2312.13010.

Chia Hsiang Kao, Wenting Zhao, Shreelekha Revankar, Samuel Speas, Snehal Bhagat, Rajeev Datta, Cheng Perng Phoo, Utkarsh Mall, Carl Vondrick, Kavita Bala, and Bharath Hariharan. 2025a. Towards llm agents for earth observation. *Preprint*, arXiv:2504.12110.

Chia Hsiang Kao, Wenting Zhao, Shreelekha Revankar, Samuel Speas, Snehal Bhagat, Rajeev Datta, Cheng Perng Phoo, Utkarsh Mall, Carl Vondrick, Kavita Bala, and Bharath Hariharan. 2025b. Towards llm agents for earth observation: The univearth dataset. *arXiv preprint*.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. 2025. Towards a science of scaling agent systems. *Preprint*, arXiv:2512.08296.

Maja Kucharczyk and Chris H. Hugenholtz. 2021. Remote sensing of natural hazard-related disasters with small drones: Global trends, biases, and research opportunities. *Remote Sensing of Environment*, 264:112577.

Kaiyu Li, Jiayu Wang, Zhi Wang, Hui Qiao, Weizhan Zhang, Deyu Meng, and Xiangyong Cao. 2025. Designing domain-specific agents via hierarchical task abstraction mechanism. *Preprint*, arXiv:2511.17198.

Chenyang Liu, Keyan Chen, Haotian Zhang, Zipeng Qi, Zhengxia Zou, and Zhenwei Shi. 2024. Change-agent: Toward interactive comprehensive remote sensing change interpretation and analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16.

Peng Luo, Xiayin Lou, Yu Zheng, Zhuo Zheng, and Stefano Ermon. 2025. Geoevolve: Automating geospatial model discovery via multi-agent large language models. *Preprint*, arXiv:2509.21593.

Jingjing Ma, Wei Jiang, Xu Tang, Xiangrong Zhang, Fang Liu, and Licheng Jiao. 2025. Multiscale sparse cross-attention network for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–16.

Yuchi Ma, Shuo Chen, Stefano Ermon, and David B. Lobell. 2024. Transfer learning in environmental remote sensing. *Remote Sensing of Environment*, 301:113924.

Zhengjing Ma and Gang Mei. 2021. Deep learning for geological hazards analysis: Data, models, applications, and opportunities. *Earth-Science Reviews*, 223:103858.

Akashah Shabbir, Muhammad Akhtar Munir, Akshay Dudhane, Muhammad Umer Sheikh, Muhammad Haris Khan, Paolo Fraccaro, Juan Bernabe Moreno, Fahad Shahbaz Khan, and Salman Khan. 2025. Thinkgeo: Evaluating tool-augmented agents for remote sensing tasks. *Preprint*, arXiv:2505.23752.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Yu Su, Diyi Yang, Shunyu Yao, and Tao Yu. 2024. Language agents: Foundations, prospects, and risks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 17–24, Miami, Florida, USA. Association for Computational Linguistics.

Jingkai Sun, Qiang Zhang, Yiqun Duan, Xiaoyang Jiang, Chong Cheng, and Renjing Xu. 2024. Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16236–16242.

Xiaoyu Tan, Bin Li, Xihe Qiu, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. Meta-agent-workflow: Streamlining tool usage in llms through workflow construction, retrieval, and refinement. New York, NY, USA. Association for Computing Machinery.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. 2025. Agent kb: Leveraging cross-domain experience for agentic problem solving. *Preprint*, arXiv:2507.06229.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen,

Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. In *Advances in Neural Information Processing Systems*, volume 37, pages 2686–2710. Curran Associates, Inc.

Wenjia Xu, Zijian Yu, Boyang Mu, Zhiwei Wei, Yuanben Zhang, Guangzuo Li, and Mugen Peng. 2025a. Rs-agent: Automating remote sensing tasks through intelligent agent. *Preprint*, arXiv:2406.07089.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-mem: Agentic memory for llm agents. *Preprint*, arXiv:2502.12110.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qiangqiang Yuan, Huanfeng Shen, Tongwen Li, Zhiwei Li, Shuwen Li, Yun Jiang, Hongzhang Xu, Weiwei Tan, Qianqian Yang, Jiwen Wang, Jianhao Gao, and Liangpei Zhang. 2020. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 241:111716.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. 2025a. Memevolve: Meta-evolution of agent memory systems. *Preprint*, arXiv:2512.18746.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2025b. Agent learning via early experience. *Preprint*, arXiv:2510.08558.

Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. 2026. Expseek: Self-triggered experience seeking for web agents. *Preprint*, arXiv:2601.08605.

A Appendix

All artifacts are used under their respective open-source licenses for research purposes.

A.1 Prompt Engineering and Agent Constraints

The operational rigor and self-evolution capabilities of RSMeM are grounded in two core prompt templates. The *System Prompt* ensures geoscientific professionalism and strict output formatting, while the *Reflection Prompt* facilitates the distillation of failures into Failure-Aware Memory (FAM).

A.1.1 System Prompt (Core Operational Logic)

The system prompt, as shown in Fig. 6, defines the agent's persona and enforces strict rules regarding tool usage and data path integrity, which are essential for multi-step Earth observation workflows.

System Prompt Template

```
### Core Role & Instructions
You are a professional geoscientist
specializing in Earth observation data
analysis.  Follow these rules STRICTLY:
1.  Tool Usage:  Use designated tools
to solve the problem step by step; retry
ONLY ONCE if a tool returns an error.
2.  Path Rule:  When a tool returns
"Result saved at /path/to/file", you
must use the FULL path "/path/to/file"
in all subsequent tool calls.
3.  Answer Format:  Output ONLY
the correct choice in this format -
<Answer>Letter<Answer>.
- Do NOT add explanations or extra text
outside the tags.
- Do NOT change the answer format under
any circumstances.
```

Figure 6: System Prompt Template for the geoscientific agent.

A.1.2 Trajectory-Grounded Reflection Prompt (Stream B: Memory Generation)

We adopt a *Reflexion-style* self-critique prompt to distill failed tool-use trajectories into reusable, as shown in Fig. 7, instance-specific constraints. Concretely, the prompt conditions the agent on (i) the original multiple-choice question (including data paths), (ii) optionally retrieved domain context, and (iii) a compressed chronological trajectory containing `tool_use`, `tool_result`, and key assistant decisions. The agent then produces a schema-constrained JSON reflection—diagnosing the failure and proposing an actionable, question-specific recovery plan—which is written into the Failure-Aware Memory (FAM, Stream B) for subsequent retrieval.

A.1.3 Significance of the Dual-Prompt Strategy

As observed in our case studies Fig. 5, this dual-prompt strategy ensures that the agent is both a

disciplined executor and a reflective learner. The *System Prompt* maintains the precision required for LST retrieval, while the *Reflection Prompt* enables the agent to autonomously correct methodological mismatches between retrieved knowledge and the specific task at hand.

A.1.4 Reflection Prompt (Original Work)

This prompt as shown in Fig. 8, represents the standard reflection mechanism used in the original ReAct-based frameworks, which relies on unstructured natural language for error diagnosis.

Trajectory-Grounded Reflection Prompt Template

You are an advanced reasoning agent that improves from self-reflection (Reflexion-style).
You will be given:
The original question (includes data path and multiple-choice options),
Retrieved knowledge context (may be empty),
A compressed trajectory from the previous run (`tool_use`, `tool_result`, and key assistant text).
Your task:
In 3–5 sentences, diagnose the most likely reason for failure and devise a new, concise, high-level plan that avoids repeating the same failure. Focus on actionable, question-specific fixes.
Required JSON schema:

```
{
"question_id": "{question_id}",
"reflective_text": "..."
}
```

Figure 7: Trajectory-grounded reflection prompt template.

Reflection Prompt (Original Work)

You are an advanced reasoning agent that can improve based on self-reflection. You will be given a previous unsuccessful reasoning trial and asked to analyze the failure.
In a few sentences, diagnose a plausible reason for failure and propose a concise, high-level mitigation plan.
Previous trial:
Question: `{question}`
Reflection:

Figure 8: Baseline reflection prompt based on original Reflexion work.

A.1.5 Reflection Prompt (Modified)

To ensure a fair and rigorous comparison, we modified the baseline prompt to align with our task-specific requirements, as shown in Fig. 9. The Reflection Prompt (Modified) explicitly instructs the agent to diagnose failures related to tool usage, path integrity, and answer formatting, rather than relying on generalized error descriptions.

Reflection Prompt (Modified)

You are an advanced reasoning agent that can improve based on self-reflection. You will be given a previous reasoning trial in which you had access to a tool environment and a question to answer. You were unsuccessful either because you produced an invalid final answer format, used refusal/guessing language, had data/path access issues, or used up your set number of reasoning steps.
In a few sentences, diagnose a possible reason for failure and devise a new, concise, high-level plan that aims to mitigate the same failure. Use complete sentences.
Required JSON schema:

```
{
"question_id": "{question_id}",
"reflective_text": "..."
}
```

Previous trial:
{question_text}{trajectory_json}
Now output the JSON only:

Figure 9: The modified reflection prompt designed for structured comparison.

A.2 Construction of the Hierarchical Knowledge Base

The hierarchical knowledge base (KB) in RSMeM is curated through a rigorous, expert-driven process designed to ensure analytical validity and geoscientific professionalism. Unlike purely automated or data-driven approaches, our KB prioritizes the formal logic of Earth observation research.

Expert Selection and Sources The curation was performed by a team of three domain experts with advanced degrees in Remote Sensing and Geospatial Information Science. The taxonomy and task definitions are grounded in authoritative sources, including the *Remote Sensing Handbook* and recent state-of-the-art surveys in Earth observation.

Three-Tier Curation Process The experts followed a systematic top-down decomposition to ensure comprehensive coverage of the geoscience landscape:

- **Domain Identification**: Three high-level domains (Urban Planning Development, Disaster Management, and Environmental Monitoring) were established to represent primary application areas.
- **Sub-Domain Mapping**: These were further decomposed into 12 sub-domains, such as Land Cover Mapping and Water Resources Management, ensuring semantic coherence.
- **Atomic Task Distillation**: A total of 64 atomic tasks were defined. Each task was

mapped to a structured Task Card containing domain definitions (D_i) and expert-validated solutions (S_i), as exemplified in Tab. 6.

Lexicon and Validation To support the hybrid scoring in HKG, experts curated a deterministic keyword lexicon for each sub-domain, capturing both spectral properties and functional land-use categories. The final KB underwent a peer-review process within the expert group to eliminate semantic ambiguity and ensure that the retrieved tool-chains align with standard geoscientific practices. This expert-distilled expertise serves as the static "genetic" foundation for the subsequent online memory evolution.

Task ID	Distilled Instance-level Memory (Reflective Text)
Q1 (TVDI)	**Failure:** Inefficient manual sampling and missing linear trend computation. **Corrected:** Compute TVDI for all dates, aggregate to annual means (2019-2022), and compute regression slope.
Q7 (LST)	**Failure:** Incorrectly used raw brightness temperature (BT10) instead of LST. **Corrected:** Strictly apply single-channel LST algorithm (using NDVI emissivity) before threshold ratio analysis.
Q9 (Urban)	**Failure:** Failed to define 'urban area' and missed emissivity correction. **Corrected:** Calculate LST $\rightarrow$ Apply urban mask (NDVI < 0.2) $\rightarrow$ Count pixels $>$ 300K.
Q12 (Spatial)	**Failure:** Processed isolated time points instead of daily composites. **Corrected:** Compute daily LST maximums across all observations to capture peak heat stress (>30%).

Table 5: Examples of distilled Instance-level Memory (FAM).

A.3 Hierarchical Knowledge Base Detail

We provide a detailed decomposition of the **Environmental Monitoring** domain within our RSMeM, specifically focusing on the **Land Cover Mapping** sub-domain. This sample illustrates how expert-curated summaries are operationalized into discrete task cards $\{T_i, D_i, S_i\}$ to guide the agent.

A.4 Instance-level Memory (FAM) Samples

Instance-level memory, denoted as *Stream B* in our framework, consists of successful reasoning patterns and error-correction logs distilled from historical interactions. When an agent fails a task, a self-reflection module generates a `reflective_text` that summarizes the failure cause and the corrected strategy.

Tab. 5 provides representative samples of how these reflections are stored and subsequently retrieved to guide future similar tasks.

Level	Content / Metadata
Domain	**Environmental Monitoring**
Sub-Domain	**Land Cover Mapping**
Keywords	Residential, Commercial, Industrial, Recreational, Infrastructure, Land cover heterogeneity, Emissivity variation ...
Decomposed Atomic Tasks (Task Cards)	
Task 1 (T_1) Sol. (S_1)	**Pixel-Wise Semantic Segmentation** Target high-res thematic mapping by assigning semantic labels to pixels; leverages spectral signatures.
Task 2 (T_2) Sol. (S_2)	**Thermal-Based Heterogeneity** Analyze thermal infrared data to quantify spatial variations; separate temperature and emissivity.
Task 3 (T_3) Sol. (S_3)	**Visual Scene Classification** Automated recognition to identify and count specific scene types based on taxonomy.
Task 4 (T_4) Sol. (S_4)	**Sub-Pixel Spectral Unmixing** Resolve material abundances by decomposing composite spectral signatures into endmembers.
Domain Definitions	
D_1	**Land cover classification**: Process of assigning spatial units to discrete categories.
D_2	**Surface Emissivity**: Effectiveness of Earth's surface in emitting thermal radiation.

Table 6: Detailed hierarchical structure of the RSMeM for the Land Cover Mapping sub-domain. The expert-curated domain knowledge is decomposed into hierarchical levels, with four representative atomic tasks shown out of the 64 available in our knowledge base.

A.5 Evaluation Metrics Details

The proficiency of Large Language Models (LLMs) in utilizing external tools is assessed through a comprehensive set of metrics. We separately report tool calling accuracy and task-level outcome with efficiency, as improvements in execution correctness do not necessarily translate to final answer accuracy or computational cost.

Tool Calling Accuracy These metrics focus on the model's ability to correctly select and sequence the necessary tools based on the ground truth sequence of expected tools, $\mathrm{GT}_{\mathrm{Tools}} = [t_1, t_2, \ldots, t_n]$. Each metric is calculated as the proportion of questions where the model's tool trajectory satisfies the specific criteria. Let M_{Tools} denote the set or sequence of tools called by the model, and N be the total number of questions.

Any-Order (AO): Measures the success rate where the **set** of tools called by the model (M_{Tools}) exactly matches the set of expected tools, irrespec-

tive of execution order:

$$\mathrm{AO} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\{\mathcal{S}(M^i_{\mathrm{Tools}}) = \mathcal{S}(\mathrm{GT}^i_{\mathrm{Tools}})\}. \quad (5)$$

In-Order (IO): Measures the success rate where the **sequence** of tools called by the model precisely matches the chronological sequence of the expected tools:

$$\mathrm{IO} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\{\mathcal{Q}(M^i_{\mathrm{Tools}}) = \mathcal{Q}(\mathrm{GT}^i_{\mathrm{Tools}})\}. \quad (6)$$

Exact-Match (EM): Measures the success rate where the entire model-generated tool-calling trajectory is an identical sequence to the reference steps, including the total number of calls, the specific tools invoked, and their order:

$$\mathrm{EM} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\{M^i = \mathrm{GT}^i\}. \quad (7)$$

Agent Reasoning Trace (Stream A: HKG Guidance)

User Query: ...calculate how much cooler was the forested area (NDVI > 0.7) compared to the surrounding non-vegetated area (NDVI < 0.2)...

— HKG Context: Task-Level Guidance (Stream A) —
Domain: Env. Monitoring **Sub-domain:** Land Cover
Task: Thermal-Based Heterogeneity Detection
Suggested: Analyze thermal infrared data to quantify spatial variations; use single-channel LST retrieval with NDVI-based emissivity.

[leftmargin=1.2em, noitemsep, topsep=4pt, label=•]

- **Step 1 (Tool: `get_filelist`)**
  Obs: [`BT10.tif`, `b4.tif`, `b5.tif`]
- **Step 2 (HKG-Guided):** Apply single-channel method.
  Tool: `lst_single_channel(bt_path=..., ...)`
  Obs: Saved at `/path/to/lst_result.tif`
- **Step 3 (Thinking):** Mean LST for $NDVI > 0.7$ and $NDVI < 0.2$.
  Tool: `calc_mean_lst(..., mode='above')` → *290.94K*
  Tool: `calc_mean_lst(..., mode='below')` → *281.97K*

Final Answer:
Thinking: Forested (290.94K) vs. Non-veg (281.97K). Diff ≈ 8.97K, closest to **Option D**.
Output: `<Answer>D<Answer>`

Figure 10: Full execution trace of the agent leveraging HKG (Stream A).

Wise-Score (WS): Wise-Score jointly evaluates trajectory correctness and parameter-level accuracy in a prefix-sensitive and importance-aware manner. It measures whether a model follows a correct sequence of tool invocations, where the contribution of each step is conditioned on all preceding steps being correct. Moreover, each tool call is weighted to reflect its relative importance in the overall task.

Formally, the score is defined as:

$$\mathrm{WS} = \frac{1}{N}\sum_{i=1}^{M}\left(\mathbb{I}(\mathrm{Tool}^i_{\mathrm{act}} = \mathrm{Tool}^i_{\mathrm{gt}})\cdot\gamma_i\right), \quad (8)$$

where N denotes the total number of tool calls in the ground-truth trajectory, and M is the length of the longest continuous matching prefix such that the predicted tool matches the ground truth at every step $j \le M$. The term γ_i represents the attribute-level correctness score of the i-th tool call, which serves as an importance-aware weight capturing the correctness and relevance of the associated parameters for that tool.

End-to-End Accuracy This metric confirms successful synthesis of tool results to produce the correct overall output. ***Accuracy (Acc):*** The ultimate end-to-end performance measure, computed as the ratio of questions where the model's **Final Answer** (y) exactly matches the **Ground Truth** final answer (y^*):

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\{y^i = y^{*i}\} \quad (9)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function that returns 1 if the condition is met, and 0 otherwise.

Efficiency Metrics These metrics evaluate the trade-off between the performance gains achieved through iterative reflection and the associated computational overhead.

Trajectory Redundancy Index(TRI): Evaluates computational overhead by comparing the total number of steps in the model's trajectory to the minimum number of reference steps required. A value closer to 1.0 indicates higher efficiency:

$$\mathrm{TRI} = \frac{\sum_{i=1}^{N}\mathrm{Steps}(M^i)}{\sum_{i=1}^{N}\mathrm{Steps}(\mathrm{GT}^i)}. \quad (10)$$

Token Efficiency (Tok.): To assess the cost-effectiveness of the autonomous evolution, we normalize the final success rate against the logarithmic total token expenditure. This represents the "accuracy yield" per unit of computational resource:

$$\eta = \frac{\mathrm{Acc}^{(T)}}{\log_{10}(\mathcal{C})}, \quad (11)$$

Agent Reasoning Trace (Stream B: Hybrid Memory Adaptation)

User Query: ...calculate the proportion of the area with LST greater than 305 K in the Okavango Delta...

— HKG Context: Urban Analysis (Stream A - Mismatch) —
Domain: Urban Planning **Sub-task:** Urban Trend Analysis
Solution: Characterize long-term urbanization; use temporal aggregation for densification patterns.

[leftmargin=1.1em, rightmargin=0.2em, noitemsep, topsep=4pt, label=•]

- **Step 1 (Tool: `get_filelist`)**
  Obs: [`BT10.tif`, `b4.tif`, `b5.tif`]
- **Step 2 (Adaptive Thinking):** *Analysis:* HKG suggests urban analysis; however, the query requires LST retrieval. Relying on Stream B, I will use the `lst_single_channel` method.
  Tool: `lst_single_channel(bt_path=..., ...)`
  Obs: Result saved at:
  `/path/to/question53/lst_result.tif`
- **Step 3 (Refining Action):** Calculate ratio with `calculate_threshold_ratio` tool.
  Tool: `calc_threshold_ratio(..., threshold=305)` → *6.3377*

Final Calculation & Answer:
Thinking: The ratio is ≈ 6.34%. Matches **Option C**.
Output: `<Answer>C<Answer>`

Figure 11: Full execution trace illustrating adaptive reasoning (Stream B).

where $\mathcal{C}$ is the total cumulative token count consumed across all attempts, including inference tokens for $a_i^{(t)}$ and generation tokens for reflections $\mathcal{M}_i^{(t)}$.

Experience Density (ED): To further calibrate the effectiveness of the generated reflections under fixed hyperparameter constraints, we introduce Experience Density. Unlike the raw ratio, ED serves as a penalized utility metric that measures the "information purity" of the reflection trace:

$$\mathrm{ED} = \mathrm{Acc} \times \left(1 - \frac{\mathcal{C}_{refl}}{\mathcal{C}_{total}}\right) \tag{12}$$

where $\mathcal{C}_{refl}$ and $\mathcal{C}_{total}$ are predefined hyperparameters. ED quantifies the model's ability to compress corrective knowledge into a limited token window; a higher ED signifies that the hierarchical grounding effectively filters redundant linguistic fillers, thereby increasing the density of actionable experience.

A.6 Qualitative Analysis and Case Studies

Validation of HKG Guidance. The trace in Fig. 10 exemplifies a successful execution driven solely by **Stream A (HKG)**. Even without prior instance-level experience from FAM (Stream B), the agent correctly identifies the professional tool-chain (`lst_single_channel` → `calculate_mean_lst_by_ndvi`) by grounding the user query into our hierarchical knowledge base. Specifically, the "Key Principle" retrieved from the Task Card prevents the agent from attempting generic image-to-temperature mappings, ensuring that geoscientific physical constraints are maintained throughout the multi-step reasoning process. **Scenario.** The agent is tasked with calculating the Land Surface Temperature (LST) difference between forested (NDVI > 0.7) and non-vegetated (NDVI < 0.2) areas in the Black Forest region using Landsat 8 imagery, the analysis as shown in Fig. 11.

Resilience under Mismatched Guidance The trace in Fig. 11 demonstrates the robustness of RSMeM when **Stream A (HKG)** retrieves a suboptimal Task Card. Despite the mismatch, the agent leverages its **Instance-level Memory (Stream B)** to maintain analytical rigor.

Scenario. The agent is tasked with calculating the proportion of the area with Land Surface Temperature (LST) greater than 305 K in the Okavango Delta using Landsat 8 data (Band 10, 4, and 5), the analysis as shown in Fig. 11.

A.7 Comparison with Reflexion

We further compare RSMeM with Reflexion, Fig.12 summarizes the Round@3 results across four LLM backbones.

A.7.1 Implementation of the Reflexion Baseline

To ensure Reflexion operates effectively within the EO environment, we specialized its feedback mechanism as follows:

Redefined Failure Signals: We expanded the original termination criteria to include environment-specific errors such as invalid answer formats, tool/path access failures, and maximum step exhaustion.

Lightweight Taxonomy: To minimize redundancy and token overhead, we replaced verbose textual reflections with a JSON-only schema based on a predefined failure taxonomy.

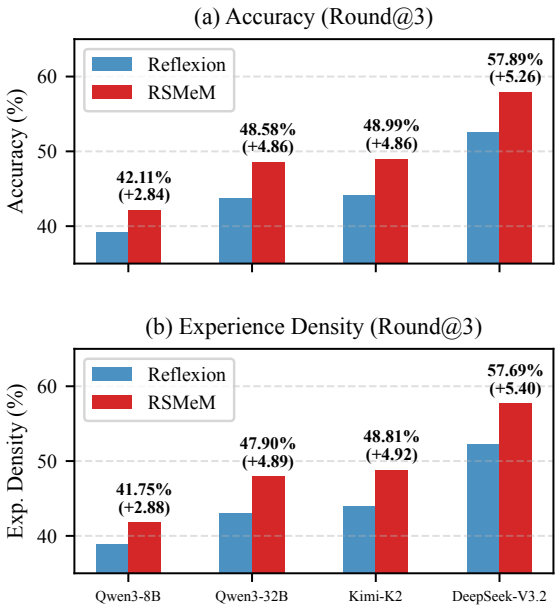

Figure 12: **Comparison with Reflexion under Round@3.** RSMeM consistently outperforms the reflection-only baseline Reflexion across four LLM backbones. (a) Outcome accuracy (%) and (b) Experience Density (ED, %) on Earth-Bench. Numbers above RSMeM bars report absolute scores, while values in parentheses denote the improvement over Reflexion.

Prompt Adaptation: The original templates were restructured to prioritize tool-calling logic while maintaining the core reflection-loop integrity, as shown in Appendix A.1.

A.7.2 Detailed Analysis

Overall, RSMeM consistently outperforms Reflexion on both outcome accuracy and Experience Density (ED) across all evaluated backbones, as shown in Fig. 12. The improvements are stable rather than model-specific, indicating that RSMeM provides benefits beyond generic linguistic reflection by grounding decisions with structured domain knowledge and distilling failure-aware, reusable constraints. Notably, ED improves alongside accuracy, suggesting that the gains are achieved through higher-utility experience utilization rather than increased trial-and-error or redundant computation.

A.8 Reflection-Token Overhead of RSMeM

As shown in Table 7, the reflection stage introduces negligible token overhead across all tested backbones. The fraction of reflection tokens remains below 1.5% of the total inference budget in every case, confirming that RSMeM's post-hoc reflection mechanism is computationally lightweight. Notably, the smaller-scale backbones (Qwen3-8B and Qwen3-32B) incur slightly higher relative costs (0.85% and 1.40%, respectively), likely because their reflective outputs tend to be more verbose. In contrast, the stronger backbones, Kimi-K2 and DeepSeek-V3.2, consume only 0.37% and 0.35% of total tokens for reflection, suggesting that higher-capacity models generate more concise and focused reflective responses.

Table 7: Reflection-token overhead of RSMeM. We report the fraction of tokens consumed by the reflection stage in the entire inference budget, i.e., $\mathcal{C}_{\text{refl}}/\mathcal{C}_{\text{total}}$, where $\mathcal{C}_{\text{refl}}$ counts tokens generated during the post-hoc reflection call and $\mathcal{C}_{\text{total}}$ counts all tokens generated in an episode. Results correspond to the same RSMeM evaluation setup as Table 1 (R@3).

Backbone	**Refl./Total (%)**
Qwen3-8B	0.85
Qwen3-32B	1.40
Kimi-K2	0.37
DeepSeek-V3.2	0.35

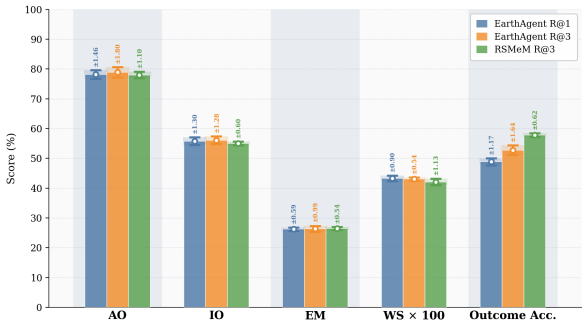

Figure 13: DeepSeek-V3.2 results (mean ± std, n=3) across all tool-calling accuracy metrics and outcome accuracy. Error bars and shaded regions indicate ±1 standard deviation. All ±std values are annotated above each bar. WS scores are scaled by 100 for uniform axis range.

A.9 DeepSeek-V3.2 Results with Variance

Fig. 13 reports DeepSeek-V3.2 performance (n=3 runs) for all five metrics across the three compared configurations. Error bars and shaded regions denote ±1 standard deviation; annotated values above each bar show the exact ±std for every method–metric pair. WS scores are multiplied by 100 for a uniform axis scale.

A.10 Extended Evaluation on ThinkGeo with DeepSeek

To provide a broader performance context, we extended our evaluation to the ThinkGeo dataset using the DeepSeek-V3.2 model. For this benchmark, we randomly sampled 100 questions and reformulated them into a multiple-choice format. To ensure a rigorous evaluation and eliminate heuristic shortcuts, we implemented a hint-free distractor strategy:

- **Numerical:** Distractors are generated by applying a random scaling factor ($0.35\times$ to $2.2\times$) to the ground truth, ensuring they remain within a plausible magnitude.
- **Directional:** Substitutions are restricted to intra-category cardinal directions (e.g., East $\rightarrow$ South/North/West) to maintain natural phrasing.
- **Counting:** Distractors use neighboring integers (GT $\pm 1 \sim 3$) to force precise quantification.
- **Descriptive:** Key adjectives are replaced with direct antonyms (e.g., *closer* $\leftrightarrow$ *farther*).
- **Binary:** All four options provide substantive, content-driven answers rather than suggestive fillers like "unknown."

The comparative results are summarized in Table 8. The data indicates that our proposed RSMeM model achieves a significant accuracy improvement of 17.0% over the Baseline, reaching an overall accuracy of 49.0%.

Model	Acc. (%)
Baseline (DeepSeek-V3.2)	32.0
RSMeM (Ours)	**49.0**
Net Gain	**+17.0**

Table 8: Accuracy comparison on the ThinkGeo subset ($n = 100$) using DeepSeek-V3.2 as the backbone.

A.11 Comparison with Retrieval Baselines

To empirically isolate the contribution of RSMeM's structured grounding from generic retrieval augmentation, we evaluate two additional baselines on 50 EarthBench tasks using DeepSeek-V3.2 (Tab. 9). **Simple RAG** retrieves the top-k passages via embedding similarity from a flat corpus, while **Web Retrieval** queries Bing for task-relevant context at inference time.

Method	EM	WS	Acc. (%)
ReAct + Simple RAG	0.0494	0.2603	36.73
ReAct + Web Retrieval	0.0437	0.2603	48.98
RSMeM (HKG + FAR)	**0.3584**	**0.6126**	**53.06**

Table 9: Comparison with retrieval baselines on 50 EarthBench tasks (DeepSeek-V3.2).

RSMeM outperforms Simple RAG by +16.33 pp in accuracy and +0.3092 in EM, confirming that taxonomy-aware structured retrieval provides fundamentally different value than flat semantic matching. Compared to Web Retrieval, RSMeM still achieves +4.08 pp in accuracy despite Bing's access to broader information, demonstrating that curated domain knowledge with failure-aware refinement is more effective than unconstrained web context for specialized geoscientific tasks.

A.12 Per-Domain Performance Breakdown

To understand where RSMeM provides the most benefit, we report per-domain accuracy changes using DeepSeek-V3.2 on EarthBench (Tab. 10).

Domain	Baseline	RSMeM	Δ (pp)
Disaster Management	48.72%	58.97%	**+10.25**
Environmental Monitoring	51.77%	58.45%	+6.68
Urban Planning & Dev.	53.73%	55.22%	+1.49

Table 10: Per-domain accuracy breakdown (DeepSeek-V3.2). RSMeM provides the largest gains in domains with standardized multi-step workflows.

RSMeM benefits most in Disaster Management (+10.25 pp), where tasks involve the most standardized multi-step workflows; HKG constraints reduce step omissions and parameter violations, while FAR guardrails curb repeated failures. Environmental Monitoring gains are moderate (+6.68 pp) due to longer pipelines with cross-modal dependencies. Urban Planning shows the smallest gain (+1.49 pp), as its shorter pipelines yield a stronger baseline and residual errors are more often dominated by upstream perception limits.