GEOSKILL: BENCHMARKING AND EVOLVING DOMAIN SKILLS FOR REMOTE-SENSING MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ultra-high-resolution (UHR) remote-sensing interpretation requires coordinating local visual inspection and broader spatial analysis under task-specific constraints. Tool-augmented agents have advanced these capabilities, but effective execution does not itself make the underlying procedures explicitly reusable. Domain skills can preserve this expertise, yet their benefits and limitations in UHR tasks remain unclear. We introduce GeoSkillBench, comprising 1,385 evaluation questions across 13 UHR tasks, fine-grained skill annotations, a structured skill library, and a shared geospatial tool environment. With models and base tools held fixed, we compare agents without skill documents, with expert-curated skills, and with self-generated plans. Expert skills improve accuracy across the evaluated models, while their advantage over self-generated plans varies by model. Paired trajectories reveal gaps in spatial verification and tool execution. Building on these findings, we introduce GeoSkillAgent, which consolidates procedural corrections from failed trajectories on a dedicated training set to iteratively refine an expert-curated SkillBank. Candidate updates and the merged SkillBank are screened for validation accuracy gains and overall stability before adoption. Model parameters remain fixed throughout, and the final SkillBank is frozen for testing. Our refinement experiments show that learning from failed trajectories yields larger gains than learning from successful trajectories alone. Our dataset and code are available at GeoSkillBench Data and GeoSkillBench Code, respectively.

1 INTRODUCTION

Ultra-high-resolution (UHR) remote-sensing imagery covers vast scenes containing numerous small objects and complex spatial layouts (Sun et al., 2022). Interpreting these scenes requires both fine-grained visual inspection and domain knowledge (Blaschke et al., 2014), as in regional screening and urban change analysis. Recent remote-sensing agents have improved task performance through active image inspection, multi-step reasoning, and geospatial tool use (Liu et al., 2025; Shabbir et al., 2026). However, the procedures that organize these operations often recur across images and queries, and successful tool use does not by itself make those procedures explicitly available for reuse. Agent skills provide a complementary mechanism by packaging procedural knowledge into reusable modules that guide how to approach a class of tasks (Li et al., 2026; Chen et al., 2026). For UHR agents, this offers a way to preserve domain expertise across tasks and refine it without modifying the underlying model.

Fine-grained recognition in remote-sensing imagery can depend on broader spatial context (Li et al., 2023). Domain skills provide reusable guidance for UHR interpretation by coordinating local inspection and broader spatial analysis to jointly evaluate task constraints. In Figure 1, selecting an equipment storage yard requires ample undeveloped land, railway proximity, and separation from dense housing. Recognizing these features separately is insufficient: all conditions must hold at the same site, following the multi-criteria logic of GIS-based suitability analysis (Malczewski, 2004). The waterside-site query changes the criteria but shares the workflow of candidate localization, cross-scale inspection, and joint verification. A domain skill can encode this shared workflow, its applicability conditions, and verification rules for reuse across images and task requirements.

Making a procedure reusable, however, does not establish how much an agent benefits from it on UHR tasks. Recent studies of skill utility, composition, and multimodal adaptation motivate this question

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

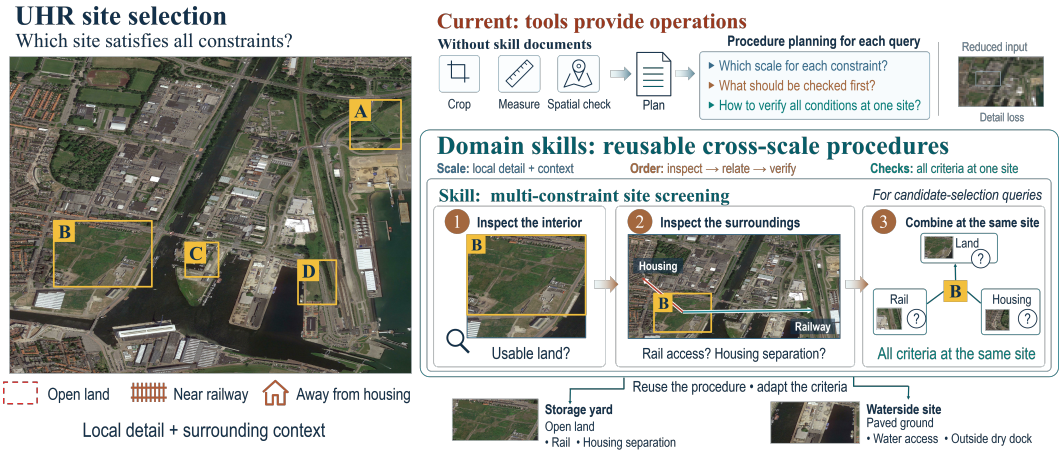

Figure 1: **Domain skills for UHR remote-sensing interpretation.** Tools provide individual operations (top right), whereas domain skills specify observation scales, operation order, and joint checks (bottom right). The bottom examples illustrate reuse of the same procedure under different site-selection criteria.

but do not directly evaluate it in UHR remote sensing (Li et al., 2026; Chen et al., 2026; Jiang et al., 2026). We therefore ask which tasks benefit from expert-curated skills and how these gains compare with those from self-generated plans. We also examine which failures persist when expert-curated skills are available. To answer these questions, we introduce **GeoSkillBench**, comprising 1,385 evaluation questions across 13 UHR remote-sensing tasks, fine-grained skill annotations, a structured skill library, and a shared geospatial tool environment. With the model and base tools held fixed, we compare agents without skill documents, with expert-curated skills, and with self-generated plans. By relating performance differences to task requirements and skill-use behavior, GeoSkillBench characterizes when procedural guidance helps and which challenges remain. These analyses provide an empirical basis for developing and evaluating skill-guided agents for UHR remote sensing.

Our evaluation shows that expert-curated skills improve accuracy across all three models, supporting the value of domain procedural knowledge for UHR agents. Self-generated plans also improve accuracy, but do not consistently outperform expert guidance. Skill benefits depend on both the task and the executing model, and the two forms of guidance solve partly different questions. Paired comparisons show that aggregate gains combine corrected answers with regressions, motivating skill refinement that preserves successful behavior. Trajectory analysis identifies remaining gaps in spatial verification, temporal correspondence, and tool execution. Together, these findings point to agents that combine reusable domain procedures with evidence checking and adaptation to tool feedback.

Building on these findings, we introduce GeoSkillAgent and a dedicated training set for iterative skill refinement. Starting from an expert-curated SkillBank, the agent uses domain procedures to guide tool-based execution. It extracts reusable procedural corrections from failed trajectories and consolidates them to revise existing skills or add new ones. Candidate updates are screened for validation accuracy gains and overall stability. The merged SkillBank must pass the same checks before adoption. Model parameters remain fixed, and the final SkillBank is frozen for testing. With GPT-5.6-Terra, paired refinement raises accuracy from 38.97% to 46.53%, a gain of 7.56 percentage points. Twelve of the 13 tasks improve, and one remains unchanged. Component ablations support learning from failed trajectories, stability screening, and merged-bank re-evaluation.

Our contributions are as follows:

- We introduce GeoSkillBench, comprising 1,385 evaluation questions across 13 UHR remote-sensing tasks, skill annotations, a structured skill library, and a controlled evaluation protocol.
- We provide systematic findings on skill utility across tasks, differences between expert guidance and self-generated plans, and the role of failure feedback in skill refinement.
- We introduce GeoSkillAgent and a dedicated training set for iterative skill refinement, combining failure analysis with validation and stability checks.

Table 1: **Comparison with multimodal and geospatial agent benchmarks.** Resolution is in pixels; RS denotes remote sensing. Size counts questions or agent tasks, including training splits where reported. GeoSkillBench reports its evaluation set. Human, Semi, and Auto denote manual, mixed, and automatic construction. Skill Eval compares skill libraries with the model and base tools fixed. – denotes unavailable data.

Benchmark	Resolution	RS	Size	Annot.	Tools	Skills	Skill Eval
General Benchmarks							
MMBench (Liu et al., 2024)	512×270	✗	3,217	Human	✗	✗	✗
MME-RealWorld (Zhang et al., 2025)	$\approx 2{,}000 \times 1{,}500$	✗	29,429	Human	✗	✗	✗
RS Benchmarks							
RSVQA-HR (Lobry et al., 2020)	512×512	✓	1,066,316	Auto	✗	✗	✗
VRSBench (Li et al., 2024)	512×512	✓	123,221	Semi	✗	✗	✗
XLRS-Bench (Wang et al., 2025c)	$8{,}500 \times 8{,}500$	✓	32,389	Human	✗	✗	✗
RSHR-Bench (Dang et al., 2025)	$8{,}700 \times 8{,}065$	✓	1,932	Semi	✗	✗	✗
LRS-VQA (Luo et al., 2025)	1,024–27,328	✓	7,333	Semi	✗	✗	✗
RS Agent Benchmarks							
ThinkGeo (Shabbir et al., 2025)	–	✓	486	Semi	✓	✗	✗
OpenEarth-Bench (Zhao et al., 2026)	–	✓	596	Semi	✓	✓	✗
GeoLLM-QA (Singh et al., 2024)	–	✓	1,000	Semi	✓	✗	✗
GeoMMBench (Xiao et al., 2026)	–	✓	1,053	Human	✓	✗	✗
GeoSkillBench (Ours)	$\approx 8{,}000 \times 8{,}000$	✓	**1,385**	Semi	✓	✓	✓

2 Related work

Remote sensing multimodal benchmarks. Remote-sensing benchmarks cover visual question answering, captioning, and grounding (Lobry et al., 2020; Hu et al., 2025; Muhtar et al., 2024; Li et al., 2024). GEOBench-VLM (Danish et al., 2025) extends evaluation to temporal and non-optical imagery. OmniEarth (Fu et al., 2026) covers multi-source geospatial tasks, while OmniEarth-Bench (Wang et al., 2025a) spans broader Earth-science domains. XLRS-Bench, RSHR-Bench, and LRS-VQA focus on interpreting large, high-resolution images (Wang et al., 2025c; Dang et al., 2025; Luo et al., 2025). Earth-Bench (Feng et al., 2026) evaluates geospatial tool use, including execution trajectories, and GeoMMBench (Xiao et al., 2026) targets expert-level multimodal geoscience reasoning. GeoSkillBench complements these resources by linking UHR questions to procedural skill requirements (Table 1).

Vision-language models and agents for remote sensing. Vision-language agents interleave reasoning with tool execution, building on reasoning-and-acting frameworks such as ReAct (Yao et al., 2023). Visual Sketchpad and DeepEyes extend this approach to visual manipulation and inspection (Hu et al., 2024; Zheng et al., 2026). For UHR remote sensing, GeoLLaVA-8K selects informative visual tokens (Wang et al., 2025b), while ZoomEarth learns to crop and zoom for active inspection (Liu et al., 2025). ThinkGeo, Earth-Agent, and GeoMMAgent coordinate geospatial tools for image interpretation and spatial analysis (Shabbir et al., 2025; Feng et al., 2026; Xiao et al., 2026). OpenEarthAgent uses validated tool trajectories for supervised training (Shabbir et al., 2026). OpenEarth-Agent retrieves expert workflow scripts to support dynamic tool creation (Zhao et al., 2026).

Skill-based agent augmentation. Reusable skills allow agents to accumulate procedures across tasks. Voyager stores executable skills (Wang et al., 2023), while ReasoningBank and ACE distill execution feedback into strategy memories and evolving context playbooks (Ouyang et al., 2026; Zhang et al., 2026a). SkillsBench evaluates skill utility (Li et al., 2026), SkillCraft studies reusable tool compositions (Chen et al., 2026), and SkillFlow benchmarks continual skill reuse and refinement (Zhang et al., 2026b). XSkill extracts skills and action-level experience from multimodal trajectories (Jiang et al., 2026), whereas SkillRL couples a skill library with policy training (Xia et al., 2026). Trace2Skill consolidates recurring trajectory lessons into reusable procedures (Ni et al., 2026). EvoSkill is closely related to our refinement setting: it analyzes failures, edits or adds skills, and uses held-out validation with a frozen backbone (Alzubi et al., 2026). Our focus is a UHR remote-sensing benchmark and empirical findings on when procedural guidance helps and which execution gaps remain.

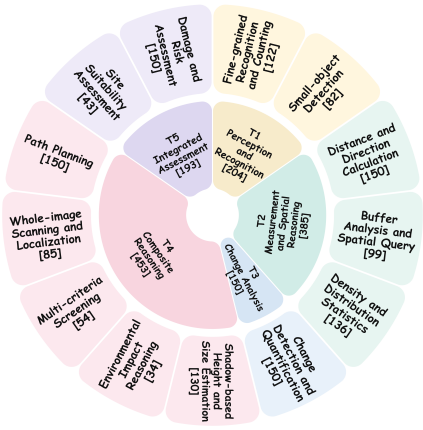

Figure 2: **Task coverage in GeoSkillBench.** Brackets show question counts; task sectors have equal width.

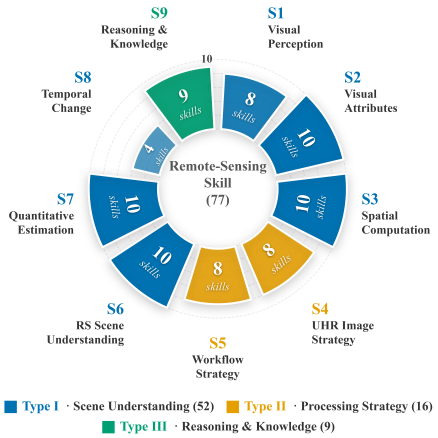

Figure 3: **Taxonomy of 77 atomic remote-sensing skills.**

3 GeoSkillBench

GeoSkillBench evaluates how multimodal agents access and use procedural knowledge in UHR remote sensing. It links questions to annotated skill requirements, readable skill documents, and executable tools. This supports comparisons of expert skill use and model-generated plans across different task demands.

Task design and coverage. The evaluation set contains 1,385 four-choice questions across 13 task types and 831 source scenes, referencing 1,058 distinct source-image paths. It includes 204 perception, 535 analysis, and 646 reasoning questions. These tasks are organized into five dimensions: perception and recognition (204 questions), measurement and spatial reasoning (385), change analysis (150), composite reasoning (453), and integrated assessment (193), as summarized in Figure 2. There are 1,235 single-image questions and 150 change-analysis questions using paired images from two time points.

The task design connects visual observations with the procedures needed to interpret them. Figure 4 provides illustrative remote-sensing questions. For example, regional screening can combine locating a specified area, inspecting candidate objects, and applying multiple spatial and visual criteria. These demands make image navigation, local inspection, and skill composition central to the evaluation.

Dataset construction and quality control. We draw UHR remote-sensing imagery and source questions from XLRS-Bench and OmniEarth-Bench (Wang et al., 2025c;a). Questions are constructed through source reuse, answer-preserving rephrasing, and visual generation. The first two routes retain reference answers from the source benchmarks. For visual generation, a vision-language model selects a region within an image tile and generates a question, options, and an answer. The question is then mapped to the source image, with regions specified in source-image pixel coordinates. Each record retains its source and spatial provenance. We construct a pool of 9,250 candidate questions and conduct three rounds of automated quality control. Structural checks cover answer formats, annotation references, and cross-field consistency. Text-only probes screen for language shortcuts, while a multimodel visual panel assesses difficulty and flags disputed answers. The final round evaluates visually generated questions on their original source images. Candidates are ranked by visual difficulty and text-only accuracy, with limits on repeated option sets and source scenes within each task. The evaluation set contains 923 questions with source-benchmark answers and 462 visually generated questions. A team of 10 experts validated the dataset, including its questions, reference answers, and annotations. All AI-assisted content and outputs underwent manual review. A separate training split of 1,697 questions supports skill refinement and shares no source scenes with the evaluation set. Appendix A provides the complete task statistics, construction rules, and data-split details.

Skill annotation and library. An atomic skill is the smallest unit used in our task annotations. We define 77 atomic skills spanning perception and computation, processing and workflow strategies, and reasoning and domain knowledge (Figure 3). Task specifications define the required skill set

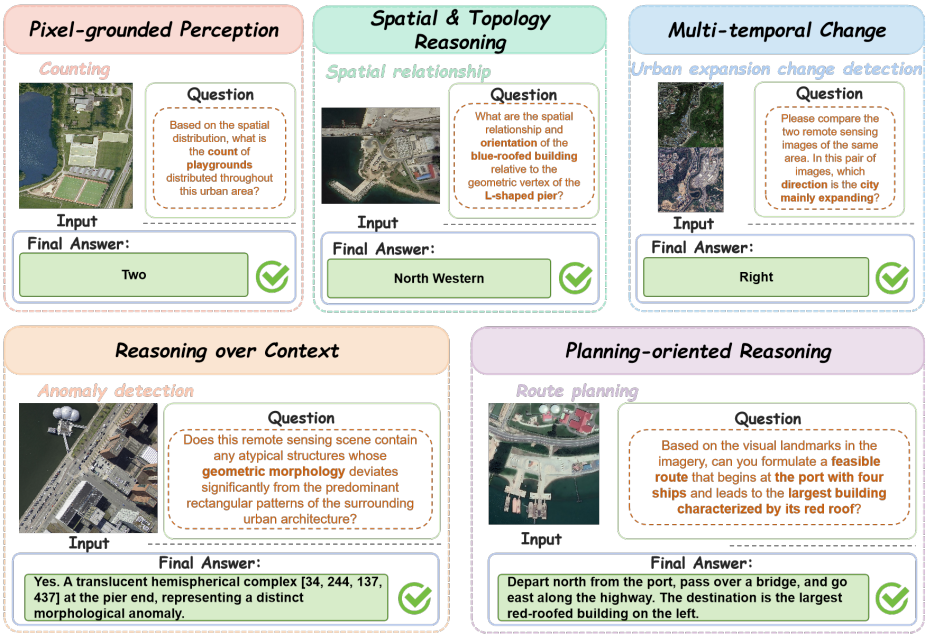

Figure 4: **Illustrative remote-sensing questions spanning perception, spatial analysis, change analysis, and reasoning.**

$\mathcal{A}_i$ stored with each question i. The skills are organized into 27 documents, with $m(s)$ mapping an atomic skill s to its document. The corresponding document set and skill count are

$$\mathcal{R}_i = \{m(s) : s \in \mathcal{A}_i\}, \qquad n_i = |\mathcal{A}_i|. \tag{1}$$

Here, $\mathcal{R}_i$ identifies relevant documents, while n_i summarizes the annotated skill requirements. Evaluation questions are annotated with 4–7 atomic skills each, with a mean of 5.29. The annotations cover 31 of the 77 skills across all nine dimensions. Several atomic skills can map to one document, and a document can combine skill types within a shared workflow, such as temporal comparison or consistency checking. Each skill document specifies when it applies and provides a procedural workflow. The workflow connects visual observations, intermediate computations, and decision criteria. Some documents also include parameter guidance, consistency checks, and reference materials. A team of 10 experts validated the skill taxonomy, document contents, and task-skill annotations. This organization makes procedural knowledge available as reusable guidance during inference. Ground-truth skill annotations remain hidden from the agent and are used to analyze which relevant documents it reads. Appendix B gives the full taxonomy, document mapping, and annotation examples.

Evaluation environment and protocol. The environment registers 38 tools covering perception, image manipulation, spatial computation, logic, statistics, aggregation, validation, and supporting operations. The agent receives an overview of the input image and uses regional tools for local inspection. Perceptual tools use the evaluated model, while coordinate calculations use deterministic operations. External retrieval is disabled in the evaluation configuration. We compare three conditions on the same questions. They share the model, base tools, image preprocessing, and a 12-round interaction budget that includes the final answer. When needed, the last round requests only the answer. **No Skills (NS)** provides tool access without skill documents. **Expert Skill Discovery (WS)** provides a directory of 27 domain documents and 87 general-domain distractor documents. The agent selects documents from their directory descriptions and reads procedures and references through dedicated tools. **Self-generated Plan (SG)** asks the model to produce a procedural plan for the current question in its first round and then execute it. Skill reads and the planning action consume the shared interaction budget. We score each answer by exact match. Accuracy A is the unweighted mean of the exact-match scores across the 13 tasks. We define three paired contrasts:

$$\Delta_{\text{expert}} = A_{\text{WS}} - A_{\text{NS}}, \quad \Delta_{\text{plan}} = A_{\text{SG}} - A_{\text{NS}}, \quad \Delta_{\text{gap}} = A_{\text{WS}} - A_{\text{SG}}. \tag{2}$$

These contrasts measure the gains from expert skill access and generated plans, and the difference between the two forms of guidance. Task-level results identify where each form of guidance helps. Paired answers reveal which questions improve or regress. We analyze skill-document access, tool calls, and interaction counts to characterize execution. Appendix C details the prompts, tools, execution settings, and scoring procedures.

4 GEOSKILLAGENT

GeoSkillAgent learns reusable procedural skills from execution failures in UHR remote-sensing tasks. Starting from an expert-authored SkillBank S_0, it alternates between skill-guided execution, failure-guided patch extraction, and validated skill updates (Figure 5). At round t, the currently accepted bank S_t is frozen as the execution library and the reference for evaluating updates. A dedicated training split provides evolution questions E_t and fixed validation and stability subsets V and R. These subsets are separated by scene and are disjoint from the final test set. Model parameters remain fixed throughout the process.

Skill-Augmented Execution. For an image I_i and question q_i, the agent accesses the skill directory and reads relevant procedures as needed. These procedures provide operation sequences, applicability conditions, and verification steps. The agent uses them to guide local image inspection, spatial computation, and result checking through the benchmark's tool environment (Figure 5(a)). Tool outputs, including generated images, become observations for subsequent reasoning. At round t, trajectories are collected on E_t using the frozen bank S_t:

$$\begin{aligned}(\tau_i, \hat{y}_i) &= \mathrm{Rollout}_\theta(I_i, q_i; S_t), \qquad i \in E_t, \\ \tau_i &= \big[(r_{i,j}, a_{i,j}, o_{i,j})\big]_{j=1}^{T_i}.\end{aligned} \tag{3}$$

Here, θ denotes the fixed model parameters and T_i is the number of recorded steps. Each step records the reasoning text $r_{i,j}$, tool call and arguments $a_{i,j}$, and observation $o_{i,j}$. Skill-reading actions are also retained, linking the retrieved guidance to subsequent execution.

Failure-Guided Patch Extraction. We identify failed executions by exact matching between the predicted option letter $\hat{y}_i$ and reference y_i. Missing or invalid answers receive a score of zero. Let $\mathcal{F}_t \subseteq E_t$ index failed trajectories with usable execution records. These trajectories provide the evidence for extracting procedural corrections (Figure 5(b)):

$$J(\hat{y}_i, y_i) = \mathbf{1}[\hat{y}_i = y_i], \qquad \mathcal{P}_t = \big\{\mathrm{Extract}(\tau_i, S_t) \mid i \in \mathcal{F}_t\big\}. \tag{4}$$

Patch extraction examines the retrieved guidance, tool arguments, operation order, and intermediate observations together. It identifies missing steps, incorrect operations, or underspecified execution conditions. Each patch describes the procedural failure, its triggering conditions, the target skill section, and a proposed correction. Source questions and supporting traces are retained separately as provenance.

To make corrections reusable, we remove the original question, options, answer, coordinates, image paths, and other instance-specific details from the procedural content. General categories and applicability conditions are retained. For example, overlapping crops may lead to duplicate counts (Figure 5(b)). The corresponding patch requires an overlap check before aggregation, expressing a reusable verification step.

Skill Consolidation and Validation. We group patches by procedural failure and target skill section, then deduplicate repeated corrections and resolve overlapping proposals using a deterministic rule. Each candidate specifies its applicability conditions, operations, and verification steps. Each update $p_{t,i}$ either adds a skill with a unique identifier or revises a declared section of an existing skill. It is applied to an independent, complete copy of S_t, producing $C_{t,i} = \mathrm{Apply}(\mathrm{Clone}(S_t), p_{t,i})$.

Candidate evaluation checks validation gains and stability, as shown in Figure 5(c). We cache the outcomes of S_t on V and R once per round. Each candidate is rerun under the same model, images, tools, prompt templates, interaction budget, decoding settings, and randomization protocol. Validation requires an accuracy gain of at least $\delta_V > 0$, fixed before evolution. Stability requires overall accuracy on R to remain unchanged or improve. A candidate is accepted only when both conditions hold. Per-task scores are recorded to characterize the distribution of changes.

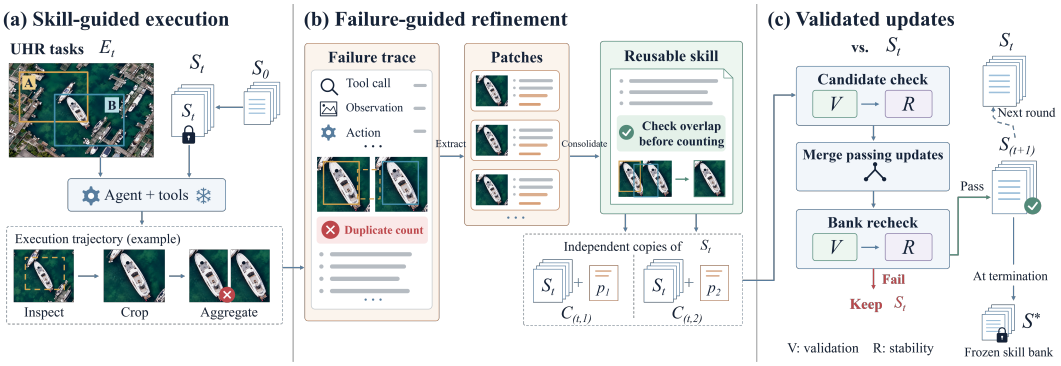

Figure 5: **Overview of GeoSkillAgent.** (a) The agent executes evolution tasks with the frozen SkillBank. (b) Failed trajectories yield reusable procedural patches, illustrated by an overlap check for regional counting. (c) Candidate banks are evaluated independently against the current bank. Passing updates are merged and rechecked for validation gains and stability before adoption. The final bank is frozen for testing.

Let $\mathrm{Acc}(C, D)$ denote exact-match accuracy with bank C on subset D, and define $\Delta_D(C; S_t) = \mathrm{Acc}(C, D) - \mathrm{Acc}(S_t, D)$. The acceptance rule is

$$G(C; S_t) = \mathbf{1}\big[\Delta_V(C; S_t) \geq \delta_V \wedge \Delta_R(C; S_t) \geq 0\big]. \tag{5}$$

The provisionally accepted updates form $A_t = \{p_{t,i} : G(C_{t,i}; S_t) = 1\}$. We merge them in a fixed order and evaluate the merged bank with the same rule:

$$\widetilde{S}_{t+1} = \mathrm{Merge}(S_t, A_t), \qquad S_{t+1} = \begin{cases} \widetilde{S}_{t+1}, & G(\widetilde{S}_{t+1}; S_t) = 1, \\ S_t, & \text{otherwise.} \end{cases} \tag{6}$$

The merged-bank check evaluates interactions among individually useful updates. If the combined bank fails, the current bank is retained for the next round. The update record links each patch to its source trajectories, target section, consolidation decision, evaluation scores, and acceptance or rejection. It tracks the SkillBank version associated with each decision. We also track cumulative changes relative to S_0.

At the end of evolution, the final bank S^* is frozen for testing (Figure 5(c)). We evaluate S^* and S_0 through the same skill-discovery protocol on the same test questions. The model, images, tools, prompt templates, interaction budget, decoding settings, and scoring rules are shared. The comparison varies the frozen SkillBank content. Appendix D provides the evolution algorithm and update-record specification.

5 Experiments

Experimental Setup. We evaluate GPT-5.6-Terra, Qwen3.5-9B, and Qwen2.5-VL-72B on 1,385 questions across 13 tasks. All conditions allow 12 interaction rounds. We report accuracy and audit paired outcomes and execution records. We additionally examine 16 question–model cases across all three conditions to illustrate workflow differences. Appendix C details the prompts, tools, and runtime settings. For NS/WS/SG, we compute 95% intervals using 20,000 paired bootstrap resamples within tasks. All overall contrast intervals include zero. Task comparisons and paired trajectories further characterize where performance differences arise and how the agent's execution changes under each form of guidance. Appendix F documents the paired trajectories used in the following analyses. Table 2 reports accuracy on all 13 tasks. Appendix E provides paired comparisons and execution diagnostics.

Finding 1. Expert skills yield positive mean gains across the evaluated models, while self-generated plans do not consistently outperform them. WS improves accuracy over NS by 1.17, 2.19, and 0.51 percentage points for GPT, Qwen3.5-9B, and Qwen2.5-VL-72B, respectively. SG also improves all three means, but its advantage over WS varies. WS leads SG by 0.20 points for GPT, whereas SG leads by 0.16 and 0.98 points for the two Qwen models. These results suggest that

Table 2: **Accuracy (%) on all 13 tasks.** NS, WS, and SG denote No Skills, Expert Skill Discovery, and Self-generated Plan. The final row averages task accuracies. Row-wise ranks: **1st** , 2nd , and 3rd . Ties share a rank. Task names appear in Appendix A.

Task	GPT-5.6-Terra			Qwen3.5-9B			Qwen2.5-VL-72B			**GeoSkillAgent (Ours)**
	NS	WS	SG	NS	WS	SG	NS	WS	SG	S_1
T1.2	37.70	36.89	36.07	31.97	36.07	32.79	34.43	33.61	37.70	**39.34**
T1.3	46.34	48.78	43.90	53.66	56.10	**58.54**	54.88	50.00	51.22	50.00
T2.1	28.00	34.00	**38.00**	19.33	17.33	24.00	16.67	18.67	17.33	36.67
T2.3	53.54	55.56	57.58	39.39	48.48	55.56	48.48	47.47	48.48	**58.59**
T2.4	**72.79**	58.82	66.91	36.03	38.97	36.76	38.24	42.65	33.82	66.91
T3.1	10.67	12.00	9.33	16.00	18.67	18.00	22.67	**24.67**	23.33	13.33
T4.1	54.62	53.85	**56.92**	38.46	52.31	39.23	32.31	33.08	36.15	53.85
T4.2	32.35	32.35	32.35	38.24	41.18	35.29	38.24	**44.12**	41.18	35.29
T4.3	72.22	**79.63**	74.07	53.70	48.15	48.15	31.48	40.74	42.59	**79.63**
T4.5	29.41	**35.29**	31.76	31.76	29.41	29.41	27.06	24.71	22.35	**35.29**
T4.6	22.00	20.00	17.33	24.00	21.33	29.33	28.67	26.00	**31.33**	26.00
T5.1	79.07	86.05	88.37	46.51	48.84	51.16	32.56	32.56	44.19	**90.70**
T5.2	19.33	20.00	18.00	23.33	24.00	24.67	**26.67**	20.67	22.00	19.33
Accuracy	42.93	44.09	43.89	34.80	36.99	37.15	33.26	33.76	34.75	**46.53**

expert procedural knowledge remains useful alongside model-generated planning. They motivate GeoSkillAgent's use of an expert library as a starting point for further refinement.

Finding 2. Skill benefits depend on the task and the executing model. On multi-condition filtering, WS changes accuracy by +7.41 points for GPT, −5.56 for Qwen3.5-9B, and +9.26 for Qwen2.5-VL-72B. Task names alone therefore do not determine whether skills help. Paired traces reveal more specific requirements. In a boundary-restricted counting case, GPT under WS localizes the marked region and inspects a crop before answering correctly. Its SG plan states the same containment rule, but successive crops lose the intended spatial scope.

In a site-selection case, Qwen3.5-9B under WS checks each candidate against open ground, railway proximity, and residential separation, then answers correctly. These cases motivate evaluating whether agents maintain the analysis region and verify all constraints for the same candidate.

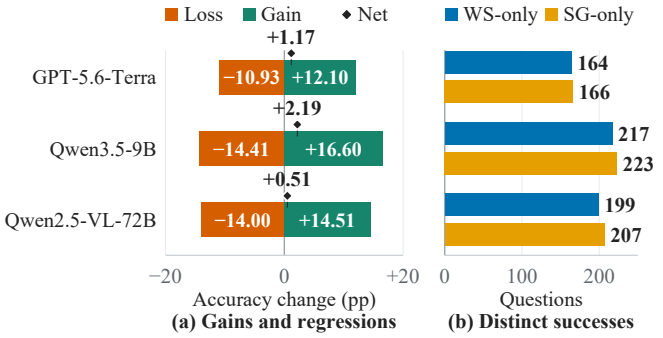

Figure 6: **Paired skill outcomes.** (a) NS-to-WS improvements and regressions in accuracy percentage points. Diamonds mark net gains. (b) Questions correct under exactly one of WS and SG.

Finding 3. Expert guidance and self-generated plans succeed on different questions and follow different execution paths. GPT answers 164 questions correctly only under WS relative to SG, and 166 only under SG. The corresponding counts are 217/223 for Qwen3.5-9B and 199/207 for Qwen2.5-VL-72B (Figure 6(b)). The traces also expose distinct strengths and gaps. In a red-roof counting case, GPT under SG switches from an empty detection result to a description request and answers correctly, while NS and WS exhaust the budget. Conversely, a Qwen2.5-VL-72B damage-assessment plan assumes that scene-change outputs provide building-level damage categories, which the tool does not return. The practical requirement is to combine reusable domain rules with adaptation to actual observations and tool outputs.

Finding 4. Net gains conceal substantial improvements and regressions. Decomposing the WS–NS accuracy difference gives improvement and regression contributions of 12.10/10.93 points for GPT, 16.60/14.41 for Qwen3.5-9B, and 14.51/14.00 for Qwen2.5-VL-72B (Figure 6(a)). Most improvements are thus offset by losses on other questions. Regression can occur after apparently

appropriate steps. In a quayside selection case, GPT under WS requests all three constraints but follows a visual-tool recommendation that disagrees with the reference answer, whereas NS and SG answer correctly. Another direction case reaches similar object locations under WS and SG, yet WS maps them to a different answer. Skill refinement therefore needs to preserve previously successful behavior while improving failed cases.

Finding 5. Skill access leaves gaps in spatial evidence, temporal correspondence, and execution. Successful skill-document retrieval covers 91.12% of GPT questions, 0.14% for Qwen3.5-9B, and 4.91% for Qwen2.5-VL-72B. All WS questions receive directory descriptions, so access and document retrieval represent different levels of guidance. Retrieval also leaves execution gaps. A GPT temporal-comparison trace maps identical normalized crop coordinates to different source regions across time points. Other traces show correct options without a complete object-level verification record. Across all nine model–condition combinations, 229 questions remain incorrect. Change detection scores range from 9.33% to 24.67%, and damage assessment from 18.00% to 26.67%. These results motivate evaluating region tracking, object correspondence, evidence checks, and recovery alongside final accuracy. Error rate alone is insufficient: Qwen3.5-9B's WS tool-error rate falls from 30.21% to 20.97%, while budget-exhausted questions rise from 661 to 750.

GeoSkillAgent Evaluation. We evaluate GeoSkillAgent with GPT-5.6-Terra as the execution model and Claude Opus 4.6 as the patch writer. Tables 2 and 3 report the evolved bank and component ablations, respectively.

In the paired refinement experiment, accuracy rises from 38.97% with S_0 to 46.53% with S_1, a gain of 7.56 percentage points. Twelve tasks improve and one remains unchanged. Gains span all five task dimensions, with the largest increases in integrated assessment and measurement and spatial reasoning. Of the paired answers, 185 change from incorrect to correct and 97 change in the reverse direction. These results show how the evolved bank improves aggregate accuracy while individual answers can still regress.

Table 3: **GeoSkillAgent ablations.** Δ: accuracy difference from Full, in percentage points.

Method	Accuracy (%)↑	Δ
GeoSkillAgent (Full)	**46.28**	—
Static expert library S_0	44.33	-1.95
w/o Stability Gate	45.82	-0.46
w/o Merged Re-evaluation	45.32	-0.96
Random-trace Induction	44.84	-1.44
Success-trace Induction	44.46	-1.82
Size-matched Sham Updates	44.33	-1.95

Base-tool calls decrease from 2,809 to 2,688, tool errors from 100 to 94, and budget-exhausted questions from 16 to 11. These execution measures improve alongside accuracy.

In the ablation study, Full improves accuracy from 44.33% with static S_0 to 46.28%, a gain of 1.95 percentage points. Removing the overall stability gate lowers accuracy by 0.46 points and increases regressed test tasks relative to S_0 from two to seven. Stability screening thus improves accuracy while reducing task regressions.

Full exceeds random-trace and success-trace induction by 1.44 and 1.82 points, respectively. Removing merged-bank re-evaluation lowers accuracy by 0.96 points, with two regressed tasks in both settings. Size-matched sham updates match S_0 at 44.33%. These results support learning procedural corrections from failures and validating merged updates. Appendix E.4 provides per-task results, paired diagnostics, ablation definitions, and the task-regression measure.

6 Conclusion

We introduced **GeoSkillBench** to evaluate procedural guidance in UHR remote-sensing agents. Expert skills improve accuracy across the evaluated models, while their advantage over self-generated plans varies by model and task. Paired trajectories expose gaps in spatial inspection, constraint checking, and tool execution, linking final answers to the procedures that produce them. These findings motivate agents that preserve spatial context, verify task constraints, and adapt to tool feedback. **GeoSkillAgent** refines an expert SkillBank from failed trajectories with fixed model parameters. Validation gains, overall stability checks, and merged-bank re-evaluation guide update selection. The evolved bank improves accuracy, and component ablations support learning reusable corrections from failures.

AI USE STATEMENT

We used generative AI to assist with manuscript drafting and language editing. We used generative AI models for benchmark data construction, automated quality screening, and the agent experiments reported in this work. A vision-language model generated a subset of questions, answer options, and candidate answers for GeoSkillBench. Language and vision-language models were also used to probe linguistic shortcuts, assess visual difficulty, and flag disputed answers, as described in Section 3. Generative models performed agent reasoning and tool use and supported the skill refinement procedures studied in GeoSkillAgent.

The authors manually reviewed all AI-assisted content and outputs, including generated benchmark questions and answers, quality screening outputs, and proposed skill updates. A team of 10 experts validated the dataset and the skill library, including the skill taxonomy, procedural documents, and task-skill annotations. This review was complemented by structural and consistency checks, reevaluation of generated questions on their original source images, and the validation and stability checks for SkillBank updates described in Section 4. The authors take full responsibility for the manuscript, dataset, methodology, reported results, and conclusions, including all content produced with assistance from generative AI.

REPRODUCIBILITY STATEMENT

The dataset sources, question construction and quality control procedures, and scene splits are described in Appendix A. Appendix B provides the atomic skill taxonomy, document mapping, and question annotation scheme. The shared evaluation conditions, visual inputs, tool interfaces, prompt excerpts, interaction budget, and scoring rules are specified in Appendix C. Section 4 and Appendix D detail the SkillBank evolution procedure, candidate and merged-bank acceptance criteria, and update-record structure. Model parameters remain fixed during evolution, and the final SkillBank is frozen before evaluation on the disjoint test set.

REFERENCES

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. EvoSkill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*, 2026. URL https://arxiv.org/abs/2603.02766.

Thomas Blaschke, Geoffrey J. Hay, Maggi Kelly, Stefan Lang, Peter Hofmann, Elisabeth Addink, Raul Queiroz Feitosa, Freek van der Meer, Harald van der Werff, Frieke van Coillie, and Dirk Tiede. Geographic object-based image analysis – towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:180–191, 2014. doi: 10.1016/j.isprsjprs.2013.09.014. URL https://doi.org/10.1016/j.isprsjprs.2013.09.014.

Shiqi Chen, Jingze Gai, Ruochen Zhou, Jinghan Zhang, Tongyao Zhu, Junlong Li, Kangrui Wang, Zihan Wang, Zhengyu Chen, Klara Kaleb, Ning Miao, Siyang Gao, Cong Lu, Manling Li, Junxian He, and Yee Whye Teh. SkillCraft: Can LLM agents learn to use tools skillfully? *arXiv preprint arXiv:2603.00718*, 2026. URL https://arxiv.org/abs/2603.00718.

Yunkai Dang, Meiyi Zhu, Donghao Wang, Yizhuo Zhang, Jiacheng Yang, Qi Fan, Yuekun Yang, Wenbin Li, Feng Miao, and Yang Gao. A benchmark for ultra-high-resolution remote sensing MLLMs. *arXiv preprint arXiv:2512.17319*, 2025. URL https://arxiv.org/abs/2512.17319.

Muhammad Sohail Danish, Muhammad Akhtar Munir, Syed Roshaan Ali Shah, Kartik Kuckreja, Fahad Shahbaz Khan, Paolo Fraccaro, Alexandre Lacoste, and Salman Khan. GEOBench-VLM: Benchmarking vision-language models for geospatial tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7132–7142, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Danish_GEOBench-VLM_Benchmarking_Vision-Language_Models_for_Geospatial_Tasks_ICCV_2025_paper.html.

Peilin Feng, Zhutao Lv, Junyan Ye, Xiaolei Wang, Xinjie Huo, Jinhua Yu, Wanghan Xu, Wenlong Zhang, Lei Bai, Conghui He, and Weijia Li. Earth-Agent: Unlocking the full landscape of earth

observation with agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dkIXAbWuxO.

Ronghao Fu, Haoran Liu, Weijie Zhang, Zhiwen Lin, Xiao Yang, Peng Zhang, and Bo Yang. OmniEarth: A benchmark for evaluating vision-language models in geospatial tasks. *arXiv preprint arXiv:2603.09471*, 2026. URL https://arxiv.org/abs/2603.09471.

Yuan Hu, Jianlong Yuan, Congcong Wen, Xiaonan Lu, Yu Liu, and Xiang Li. RSGPT: A remote sensing vision language model and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 224:272–286, 2025. doi: 10.1016/j.isprsjprs.2025.03.028. URL https://doi.org/10.1016/j.isprsjprs.2025.03.028.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual Sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379, 2024. doi: 10.52202/079017-4423. URL https://papers.nips.cc/paper_files/paper/2024/hash/fb82011040977c7712409fbdb5456647-Abstract-Conference.html.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. XSkill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026. URL https://arxiv.org/abs/2603.12056.

Xiang Li, Jian Ding, and Mohamed Elhoseiny. VRSBench: A versatile vision-language benchmark dataset for remote sensing image understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 3229–3242, 2024. doi: 10.52202/079017-0106. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/05b7f821234f66b78f99e7803fffa78a-Abstract-Datasets_and_Benchmarks_Track.html.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Binxu Li, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, Yipeng Gao, Junwei He, Yizhuo He, Liqiang Jing, Luyang Kong, Xin Lan, Jiachen Li, Songlin Li, Yijiang Li, Yueqian Lin, Xinyi Liu, Xuanqing Liu, Haoran Lyu, Ze Ma, Bowei Wang, Runhui Wang, Tianyu Wang, Wengao Ye, Yue Zhang, Hanwen Xing, Yiqi Xue, Steven Dillmann, and Han-chung Lee. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026. URL https://arxiv.org/abs/2602.12670.

Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li. Large selective kernel network for remote sensing object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16794–16805, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_Large_Selective_Kernel_Network_for_Remote_Sensing_Object_Detection_ICCV_2023_paper.html.

Ruixun Liu, Bowen Fu, Jiayi Song, Kaiyu Li, Wanchen Li, Lanxuan Xue, Hui Qiao, Weizhan Zhang, Deyu Meng, and Xiangyong Cao. ZoomEarth: Active perception for ultra-high-resolution geospatial vision-language tasks. *arXiv preprint arXiv:2511.12267*, 2025. URL https://arxiv.org/abs/2511.12267.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/00959.pdf.

Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. RSVQA: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8555–8566, 2020. doi: 10.1109/TGRS.2020.2988782. URL https://doi.org/10.1109/TGRS.2020.2988782.

Junwei Luo, Yingying Zhang, Xue Yang, Kang Wu, Qi Zhu, Lei Liang, Jingdong Chen, and Yansheng Li. When large vision-language model meets large remote sensing imagery: Coarse-to-fine text-guided token pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9206–9217, 2025. doi: 10.1109/ICCV51701.2025.00860. URL https://openaccess.thecvf.com/content/ICCV2025/html/Luo_When_Large_Vision-Language_Model_Meets_Large_Remote_Sensing_Imagery_Coarse-to-Fine_ICCV_2025_paper.html.

Jacek Malczewski. GIS-based land-use suitability analysis: a critical overview. *Progress in Planning*, 62(1):3–65, 2004. doi: 10.1016/j.progress.2003.09.002. URL https://doi.org/10.1016/j.progress.2003.09.002.

Dilxat Muhtar, Zhenshi Li, Feng Gu, Xueliang Zhang, and Pengfeng Xiao. LHRS-Bot: Empowering remote sensing with VGI-enhanced large multimodal language model. In *Computer Vision – ECCV 2024*, volume 15132 of *Lecture Notes in Computer Science*, pp. 440–457. Springer, 2024. doi: 10.1007/978-3-031-72904-1_26. URL https://doi.org/10.1007/978-3-031-72904-1_26.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026. URL https://arxiv.org/abs/2603.25158.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

Akashah Shabbir, Muhammad Akhtar Munir, Akshay Dudhane, Muhammad Umer Sheikh, Muhammad Haris Khan, Paolo Fraccaro, Juan Bernabe Moreno, Fahad Shahbaz Khan, and Salman Khan. ThinkGeo: Evaluating tool-augmented agents for remote sensing tasks. *arXiv preprint arXiv:2505.23752*, 2025. URL https://arxiv.org/abs/2505.23752.

Akashah Shabbir, Muhammad Umer Sheikh, Muhammad Akhtar Munir, Hiyam Debary, Mustansar Fiaz, Muhammad Zaigham Zaheer, Paolo Fraccaro, Fahad Shahbaz Khan, Muhammad Haris Khan, Xiao Xiang Zhu, and Salman Khan. OpenEarthAgent: A unified framework for tool-augmented geospatial agents. *arXiv preprint arXiv:2602.17665*, 2026. URL https://arxiv.org/abs/2602.17665.

Simranjit Singh, Michael Fore, and Dimitrios Stamoulis. Evaluating tool-augmented agents in remote sensing platforms. In *ICLR 2024 Machine Learning for Remote Sensing Workshop*, 2024. URL https://ml-for-rs.github.io/iclr2024/camera_ready/papers/65.pdf.

Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, Martin Weinmann, Stefan Hinz, Cheng Wang, and Kun Fu. FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022. doi: 10.1016/j.isprsjprs.2021.12.004. URL https://doi.org/10.1016/j.isprsjprs.2021.12.004.

Fengxiang Wang, Mingshuo Chen, Xuming He, Yi-Fan Zhang, Yueying Li, Feng Liu, Zijie Guo, Zhenghao Hu, Jiong Wang, Jingyi Xu, Zhangrui Li, Junchao Gong, Di Wang, Fenghua Ling, Ben Fei, Weijia Li, Long Lan, and Wenjing Yang. OmniEarth-Bench: Towards holistic evaluation of Earth's six spheres and cross-spheres interactions with multimodal observational earth data. *arXiv preprint arXiv:2505.23522*, 2025a. URL https://arxiv.org/abs/2505.23522.

Fengxiang Wang, Mingshuo Chen, Yueying Li, Di Wang, Haotian Wang, Zonghao Guo, Zefan Wang, Boqi Shan, Long Lan, Yulin Wang, Hongzhen Wang, Wenjing Yang, Bo Du, and Jing Zhang. GeoLLaVA-8K: Scaling remote-sensing multimodal large language models to 8k resolution. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-5317. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e95ef5c6ed4096c609e0b8b47ffaeb9b-Abstract-Conference.html.

Fengxiang Wang, Hongzhen Wang, Zonghao Guo, Di Wang, Yulin Wang, Mingshuo Chen, Qiang Ma, Long Lan, Wenjing Yang, Jing Zhang, Zhiyuan Liu, and Maosong Sun. XLRS-Bench: Could your multimodal LLMs understand extremely large ultra-high-resolution remote sensing imagery? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14325–14336, 2025c. doi: 10.1109/CVPR52734.2025.01336. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_XLRS-Bench_Could_Your_Multimodal_LLMs_Understand_Extremely_Large_Ultra-High-Resolution_Remote_CVPR_2025_paper.html.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026. URL https://arxiv.org/abs/2602.08234.

Aoran Xiao, Shihao Cheng, Yonghao Xu, Yexian Ren, Hongruixuan Chen, and Naoto Yokoya. GeoMMBench and GeoMMAgent: Toward expert-level multimodal intelligence in geoscience and remote sensing. *arXiv preprint arXiv:2604.08896*, 2026. URL https://arxiv.org/abs/2604.08896.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2510.04618.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k5VHHgsRbi.

Ziao Zhang, Kou Shi, Shiting Huang, Avery Nie, Yu Zeng, Yiming Zhao, Zhen Fang, Qishen Su, Haibo Qiu, Wei Yang, Qingnan Ren, Shun Zou, Wenxuan Huang, Lin Chen, Zehui Chen, and Feng Zhao. SkillFlow: Benchmarking lifelong skill discovery and evolution for autonomous agents. *arXiv preprint arXiv:2604.17308*, 2026b. URL https://arxiv.org/abs/2604.17308.

Sijie Zhao, Feng Liu, Xueliang Zhang, Hao Chen, Xinyu Gu, Zhe Jiang, Fenghua Ling, Ben Fei, Wenlong Zhang, Junjue Wang, Weihao Xuan, Pengfeng Xiao, Naoto Yokoya, and Lei Bai. OpenEarth-Agent: From tool calling to tool creation for open-environment earth observation. *arXiv preprint arXiv:2603.22148*, 2026. URL https://arxiv.org/abs/2603.22148.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xUyMXkI958.

A Dataset Details

A.1 Task coverage and composition

Table A.1 lists the 13 task types across five task dimensions in the 1,385-question GeoSkillBench evaluation set. Perception comprises T1, analysis comprises T2–T3, and reasoning comprises T4–T5. The evaluation set contains 204, 535, and 646 questions in these cognitive layers, respectively. Its domain labels cover urban scenes (646), transportation (439), and disasters (300). All questions have four options and one reference answer. The answer counts are 323, 328, 359, and 375 for A–D. There are 1,235 single-image questions and 150 questions with paired temporal images. The difficulty annotations comprise 218 L2 questions and 1,167 L3 questions.

Table A.1: **Task coverage and evaluation sizes.** Task identifiers follow the final taxonomy. T3.1 uses paired temporal images. Other task types use a single image per question.

ID	Task	Test
T1: Perception and recognition (204)		
T1.2	Fine-grained recognition and counting	122
T1.3	Small object detection	82
T2: Measurement and spatial reasoning (385)		
T2.1	Distance and direction calculation	150
T2.3	Buffer and spatial query	99
T2.4	Density and distribution statistics	136
T3: Change analysis (150)		
T3.1	Change detection and quantification	150
T4: Composite reasoning (453)		
T4.1	Shadow-based height and size estimation	130
T4.2	Environmental impact reasoning	34
T4.3	Multi-condition filtering	54
T4.5	Whole-image scanning and localization	85
T4.6	Route planning	150
T5: Integrated assessment (193)		
T5.1	Site suitability	43
T5.2	Damage and risk assessment	150
Total		**1,385**

Original question identifiers remain unchanged for provenance. The `task_type` field determines the final task assignment.

A.2 Construction and source-image coordinates

We construct 9,250 candidates from XLRS-Bench and OmniEarth-Bench (Wang et al., 2025c;a). Pool A directly reuses source questions, options, and answers (2,000 candidates). Pool B rewrites questions while preserving source options and answers (2,250 candidates). When rewrites introduced visual answer cues, we restored the source wording. Together, these routes contribute 923 evaluation questions with source-benchmark answers.

Pool C generates 5,000 candidates from image tiles to extend task coverage. A vision-language model selects a target region and produces a question, options, an answer, and a rationale. Accepted questions are mapped back to the source image. Tile offsets transform local regions into source-image pixel coordinates, and the question specifies the scene region to inspect. Relative terms such as upper-left refer to that region. The evaluation set contains 462 questions from this route.

Grouping source labels into dataset families gives DOTA-v2 (569 questions), XView under the OmniEarth source label (300), MiniFrance (223), HRSCD (165), Potsdam (51), Toronto (51), and ITCVD (26).

Each generated record distinguishes the scene region (`tile_bbox`) from the target annotation (`bbox_used`). Agent evaluation provides the source images referenced by the record's `image` field:

Table A.2: **Final selection settings.** Blind and panel accuracy denote per-question averages over the text-only and visual predictions. Scene and option-set limits apply within each task.

Criterion	Setting
Text-only shortcut	Exclude if blind accuracy > 0.6.
Visual difficulty	Exclude if panel accuracy > 0.8.
Region coverage	Exclude if annotated target area / scene-region area > 0.8, when recorded.
Answer disagreement	Exclude if all four panel models choose the same non-reference answer.
Question validity	Exclude duplicate option texts and references to absent visual markers.
Task quota	At most 150 questions.
Source-scene diversity	At most three questions from one scene.
Option-set diversity	At most three questions sharing an option set. This cap is relaxed for T2.4, T3.1, and T5.2, which use fixed category sets.
Randomization	Seed 42 for tie breaking and review sampling.

one source image for single-image questions and two source images for paired temporal questions. An intermediate composite with a highlighted region and enlarged view is retained as `review_image` for inspection. This composite is excluded from the agent input. Source dataset, parent question, construction route, and coordinate metadata preserve the link between each question and its evidence.

A.3 Quality control and selection

We conduct three rounds of quality control. Structural checks validate fields, answer formats, references to task and skill definitions, and cross-field consistency. Text-only probes identify questions that can be answered from wording or option patterns. We revise leaked target descriptions and weak distractors before repeating these probes. A visual panel assesses question difficulty and agreement with the reference answer. The final round evaluates visually generated questions on their original source images.

The final candidate records contain six text-only predictions and four visual predictions per question. Text-only probes use two models with three samples each at temperature 0.7. The visual panel uses GPT-5.6-Terra, GPT-6-Luna, Gemini-3.8-Flash, and Gemini-3.7-Flash. The first two also serve as text-only probes. During visual QC, models receive a source-image overview and, for coordinate-based questions, a regional crop. For Pool C, displayed regions follow coordinates present in the question. These QC views serve dataset screening. The agent evaluation follows the observation and tool protocol in Section 3.

After filtering (Table A.2), candidates are ranked by $(1 - \text{panel_acc}) - 0.5\,\text{blind_acc}$ with a uniform tie-breaking term in $[0, 10^{-3})$. The selector adds 0.15 when a source scene already appears in the evaluation set, preserving more unused scenes for training.

An item is flagged for review when at least three visual models agree on a non-reference answer and none selects the reference. Unanimous disagreement triggers exclusion. A team of 10 experts completed the dataset review and validation. They reviewed questions, options, reference answers, and annotations, including AI-assisted content and items flagged by visual-panel disagreement. The final evaluation set contains 1,385 expert-validated questions.

A.4 Scene splits and record format

The evaluation set contains 1,385 questions from 831 source-scene keys. Its image fields reference 1,058 unique source-image paths, counting the two images in each temporal pair separately. The training pool contains 1,697 questions for skill evolution. Evaluation and training source-scene keys do not overlap. The keys group questions from the same original image across construction routes, including corresponding temporal images and source-image variants. Training excludes unanimous answer disagreements, duplicate options, and absent-marker references.

Table A.3 separates inference inputs from annotations. Reference procedural descriptions are available for 132 evaluation questions: 99 in T2.3 and 33 in T5.2. They provide textual steps for inspecting the intended reasoning procedure.

Table A.3: **Question record fields and their roles.** Hidden annotations support analysis and provenance. Regions stated explicitly in the question remain part of the public input.

Role	Fields
Question input	`question`, `options`, `image`. The record identifier `id` links predictions to the question.
Hidden annotations	Reference answer, task, cognitive layer, domain, skill count, required atomic skills, and relevant skill documents.
Hidden provenance	Source dataset, parent identifier, construction route, target coordinates, scene-region coordinates, and review-image path where available.

B Skill Taxonomy, Annotation, and Library

B.1 Atomic skill taxonomy

The taxonomy connects task requirements to reusable procedures. Its 77 atomic skills span nine dimensions and three types. Type I contains 52 perception and computation skills. Type II contains 16 strategies for image inspection and workflow organization. Type III contains nine reasoning and knowledge skills. Table B.1 gives the complete catalog. Each entry has a stable identifier, a definition, a type, a dimension, indicative tools, and a document assignment. The tool associations describe possible implementations of a skill. The question annotations use the skill identifiers independently of the particular tool sequence taken by an agent. A team of 10 experts reviewed and validated the 77-skill taxonomy, all 27 procedural documents, and task-skill annotations, including their AI-assisted content.

Table B.1: **Complete catalog of 77 atomic skills.** Dimension headings specify the skill type. GSD denotes ground sampling distance.

ID	Operation or judgment
S1: Visual Perception and Localization (Type I)	
S1.1	Identify the overall scene type of an image or region.
S1.2	Detect and localize objects of a specific class.
S1.3	Count instances of a target class.
S1.4	Determine fine-grained subclass of an object.
S1.5	Segment land-cover classes at the pixel level.
S1.6	Segment individual object instances.
S1.7	Localize target objects/regions via natural language descriptions.
S1.8	Detect very small targets, typically fewer than 20 pixels across.
S2: Fine-grained Visual Attributes (Type I)	
S2.1	Recognize color and texture attributes.
S2.2	Analyze target shape, aspect ratio and regularity.
S2.3	Analyze target orientation or rotation angle.
S2.4	Estimate real-world size of an object using GSD.
S2.5	Infer building height from shadow length and sun angle.
S2.6	Assess physical condition, such as intact, damaged, abandoned, or under construction.
S2.7	Infer surface material of a target.
S2.8	Infer functional use of a building or facility from visual cues.
S2.9	Determine whether an object is in use, idle, abandoned, or under construction.
S2.10	Determine whether vehicles are parked, moving, or queued.
S3: Measurement and Spatial Computation (Type I)	
S3.1	Compute real-world distance between two targets using GSD.
S3.2	Determine compass direction of A relative to B.
S3.3	Compute real-world area from segmentation mask and GSD.
S3.4	Generate a buffer zone around a reference feature and query within it.
S3.5	Intersect, unite, or subtract spatial masks.
S3.6	Aggregate counts, areas, or densities within specified zones.
S3.7	Determine containment, contact, overlap, or disjointness.
S3.8	Find the nearest same-class or different-class neighbor and its distance.
S3.9	Analyze road-network based accessibility and connectivity.
S3.10	Perform spatial clustering and identify cluster centers.

Table B.1 continued.

ID	Operation or judgment
S4: UHR Image Processing Strategy (Type II)	
S4.1	Understand the macro-scale layout and functional zoning.
S4.2	Systematically scan a UHR image via grid splitting.
S4.3	Localize a specific region in a UHR image given description or direction.
S4.4	Zoom in on candidate regions for detailed verification.
S4.5	Compare differences across multiple regions of the same image.
S4.6	Recognize spatial patterns visible only at macro scale.
S4.7	Refine target identification based on surrounding context.
S4.8	Switch between bi-temporal images to locate changed regions.
S5: Multi-step Workflow Strategy (Type II)	
S5.1	Establish the relation between GSD and real-world scale.
S5.2	Joint filtering across multiple conditions.
S5.3	Run the same analysis on multiple regions and aggregate results.
S5.4	Locate first, then analyze attributes within the localized region.
S5.5	Coarse-to-fine analysis across global and local scales.
S5.6	Systematic bi-temporal change comparison workflow.
S5.7	Check intermediate results for consistency and plausibility.
S5.8	Integrate evidence across multiple analysis steps for a final conclusion.
S6: Remote Sensing Scene Understanding (Type I)	
S6.1	Infer land-use type of a region.
S6.2	Recognize facilities such as schools, hospitals, ports, airports, and substations.
S6.3	Analyze structure, hierarchy and connectivity of the road network.
S6.4	Recognize grid, scattered, enclosed, or row-aligned building layouts.
S6.5	Classify urban, suburban, peri-urban, or rural settings.
S6.6	Recognize residential, commercial, industrial, or educational zones.
S6.7	Distinguish rivers, lakes, reservoirs, ponds, canals, and seas.
S6.8	Distinguish forest, grassland, cropland, and urban vegetation.
S6.9	Recognize terraces, fish ponds, greenhouses, and center-pivot fields.
S6.10	Recognize flood, fire, collapse, and landslide traces.
S7: Quantitative Estimation and Statistics (Type I)	
S7.1	Compute area coverage ratio of a specific class.
S7.2	Estimate spatial density of target objects.
S7.3	Measure length of linear features (roads or rivers).
S7.4	Identify clustered, random, or uniform spatial distributions.
S7.5	Compute proportions of different classes by area or count.
S7.6	Analyze spatial density gradients, such as center-to-periphery trends.
S7.7	Analyze size distribution characteristics of a target population.
S7.8	Estimate floor area ratio from building footprints and inferred floor counts.
S7.9	Compute road-network density and structural indices.
S7.10	Compute the fraction covered by buildings, roads, and paved ground.
S8: Temporal Change Analysis (Type I)	
S8.1	Detect regions that have changed between two time stamps.
S8.2	Distinguish new construction, demolition, renovation, and recovery.
S8.3	Quantify added area, removed objects, or the fraction of change.
S8.4	Localize main changed regions and assess severity.
S9: Reasoning, Knowledge and Judgment (Type III)	
S9.1	Causal reasoning from observed phenomena to causes or outcomes.
S9.2	Threshold/rule-based conditional judgment.
S9.3	Apply remote-sensing, geography, or urban-planning knowledge.
S9.4	Systematic comparative evaluation across multiple regions or options.
S9.5	Multi-criteria suitability assessment.
S9.6	Assess disaster or environmental risk level.
S9.7	Predict future trends from observed change patterns.
S9.8	Explain causes of observed spatial anomalies.
S9.9	Use commonsense priors to verify the plausibility of analysis results.

B.2 From atomic skills to documents

The library groups related atomic skills into 27 procedural documents. Every atomic skill belongs to exactly one document, as listed in Table B.2. A document can combine different skill types when they

support the same procedure. For example, temporal comparison includes both temporal inspection strategies and change-analysis operations. Consistency checking combines workflow checks with commonsense verification. The three skill types describe the atomic entries, while the directory categories organize reusable documents.

Each document starts with metadata containing its name, description, and covered skill identifiers. The body specifies when the procedure applies. Three documents, overview-first inspection, detection and counting, and buffer-overlay analysis, provide expanded workflows with parameters and sanity checks. The other 24 specify core reasoning steps and links to related skills. Eight documents also provide supplementary reference files. The discovery directory combines these 27 domain documents with 87 documents from a general-domain distractor collection. Document access and reference access are described in Appendix C.

Table B.2: **Mapping from the 27 domain documents to atomic skills.** Document identifiers omit the shared `skill_` prefix. Every atomic identifier occurs once.

Document	Covered atomic skills
`scene_classification`	S1.1
`detect_count`	S1.2, S1.3, S1.4, S1.8
`visual_grounding`	S1.7
`segment_coverage`	S1.5, S1.6, S7.1, S7.5
`attribute_judgment`	S2.1, S2.2, S2.3, S2.4, S2.5, S2.7
`condition_assessment`	S2.6, S2.9, S2.10, S6.10
`land_use_inference`	S2.8, S6.1, S6.5, S6.6
`facility_recognition`	S6.2, S6.4
`vegetation_water`	S6.7, S6.8, S6.9
`road_analysis`	S6.3, S7.3, S7.9, S3.9
`distance_direction`	S3.1, S3.2
`buffer_overlay`	S3.4, S3.5, S3.6
`area_measurement`	S3.3, S7.8, S7.10
`topology_neighbor`	S3.7, S3.8, S3.10
`density_distribution`	S7.2, S7.4, S7.6, S7.7
`overview_first`	S4.1, S4.6
`grid_scanning`	S4.2, S5.5
`locate_then_analyze`	S4.3, S4.4, S4.7, S5.4
`cross_region_compare`	S4.5, S5.3
`temporal_comparison`	S4.8, S5.6, S8.1, S8.2, S8.3, S8.4
`multi_condition`	S5.1, S5.2
`consistency_check`	S5.7, S5.8, S9.9
`threshold_judgment`	S9.2
`causal_reasoning`	S9.1, S9.8
`suitability_risk`	S9.5, S9.6
`comparative_eval`	S9.4, S9.7
`domain_knowledge`	S9.3

B.3 Question annotations and composition sizes

Task specifications assign the required atomic set $\mathcal{A}_i$ to each question i. Let $m(s)$ map atomic skill s to its document. The relevant document set and annotated skill count are

$$\mathcal{R}_i = \{m(s) : s \in \mathcal{A}_i\}, \qquad n_i = |\mathcal{A}_i|. \tag{B.1}$$

The record stores these sets, n_i, and the counts of the three skill types. Multiple atomic skills can map to one document, so $|\mathcal{R}_i|$ can be smaller than n_i. All 1,385 evaluation records satisfy these mapping and count relations. These fields remain hidden during inference and provide reference labels for analyzing skill discovery. The document labels specify a task-assigned reference subset of relevant procedures. Additional unannotated documents may also support a valid solution. Actual execution length is measured from tool calls and interaction rounds.

The evaluation set contains 218 questions with four annotated skills, 606 with five, 507 with six, and 54 with seven. The mean is 5.29. Across questions, Type I, II, and III contribute 5,678, 1,363, and

281 atomic-skill occurrences, respectively. A total of 281 questions combine all three types. The annotations cover 31 distinct atomic skills across all nine dimensions and map to 18 expert documents. These coverage counts describe the evaluation questions; the complete library contains 77 atomic skills in 27 domain documents.

Region-restricted counting. Question `Q_PL_T1_2_MM1_0091` asks for the number of house groups to the right of a small wooded patch in the upper-right part of the image. Its annotation contains S1.3, S1.4, S2.1, S2.2, and S5.4. These cover counting, fine-grained recognition, color and texture, shape, and localization before analysis. The five entries map to three documents: detection and counting, attribute judgment, and locate-then-analyze. The example connects a spatially restricted question to both visual judgments and an inspection strategy.

Site suitability. Question `Q_PL_T5_1_MV_1187` asks which site meets three requirements for a temporary public event hub: direct vehicular access to the main north–south road, an existing paved and marked parking surface, and immediate adjacency to an outdoor sports court. Its task-assigned annotation contains S5.1, S5.2, S3.4, S3.9, S3.3, and S9.2. The six entries map to five documents: multi-condition analysis, buffer-overlay analysis, road analysis, area measurement, and threshold judgment. The annotation combines computation, workflow strategy, and conditional reasoning.

B.4 Representative procedural content

The following summaries retain the operations and checks in two library documents. They illustrate how a document turns annotated capabilities into a reusable procedure.

Detection and counting: an expanded workflow. The procedure first identifies the target class and the spatial scope. A textual region description is localized before counting. For common objects with a clear appearance, it calls `count_objects`. For dense, overlapping, or small targets, it detects individual instances and filters them by region. UHR counting uses image tiles and aggregates their results. The checks compare the resulting count with the answer ranges and revisit crops when the count suggests missed objects or duplicates. The document specifies class-name conventions and requires counting within the requested region.

Cross-region comparison: a core procedure. The procedure defines comparable regions from a geometric partition or a dividing feature. It applies the same analysis to every region, such as vegetation coverage, building density, or parking occupancy. It then tabulates the regional values and ranks or contrasts them according to the question. A final check compares regional aggregates with plausible image-level totals. Links to locate-then-analyze and consistency checking connect this procedure to more focused procedures.

C Evaluation Protocol and Implementation

C.1 Shared execution and skill conditions

Within each model configuration, the three benchmark conditions use the same questions, model, base tools, image encoding, and interaction limit. Table C.1 specifies their differences. Question text, options, and images enter the model context. Reference answers, task labels, required skills, and relevant-document annotations are reserved for evaluation. The WS directory presents document identifiers, descriptions, and available reference filenames in alphabetical order. The agent selects documents from these metadata descriptions.

Each interaction requests one JSON object containing reasoning text and one action. A terminal action returns an option letter. Skill reads, the SG planning action, invalid-action responses, and termination all consume interaction rounds. Each rollout has a budget of 12 rounds, including the final answer and forced-answer termination when needed. Perception tools can also make model calls within an action, so interaction counts and total model calls measure different costs. SG generates an independent plan for each question.

Table C.1: Skill access under the three evaluation conditions. All conditions retain the 38 base tools.

Condition	Procedural information	Additional actions
NS	Shared instructions and tool descriptions	None
WS	Directory of 27 domain and 87 distractor documents	`read_skill`, `read_skill_reference`
SG	A plan generated for the current question	First-turn `NoOp` for the plan

C.2 Visual inputs and tool interfaces

The initial request uses the source images referenced by each question's `image` field, in their recorded order. Single-time questions provide one source image, and the 150 temporal questions provide both time points together. The input-image limits are listed in Table C.3. Original pixels remain available to image tools. The crop interface specifies normalized coordinates in $[0, 1]$ relative to the current view, while regions specified in question text use source-image pixel coordinates. Crop observations report the corresponding source region to support subsequent navigation.

Tool-generated views support subsequent model requests. Tool observations and errors enter the text conversation, and the traces retain the returned text and generated-image paths for behavioral analysis.

Table C.2: Base tools and their input/output types. Skill-reading actions are listed separately from the base-tool count.

Category	Tools	Inputs and outputs
Perception (6)	`detect`, `segment`, `describe`, `region_attribute`, `text_to_bbox`, `change_detect`	Images and queries to descriptions, boxes, attributes, or change summaries
Image operations (9)	`overview`, `crop_region`, `grid_split`, `tile_extract`, `pan_view`, `draw_box`, `diff_images`, `compare_images`, `enhance_view`	Images and regions to views, comparisons, or difference maps
Spatial computation (9)	`compute_distance`, `compute_area`, `compute_direction`, `buffer_zone`, `spatial_overlay`, `topological_relation`, `nearest_neighbor`, `count_in_region`, `measure_length`	Coordinates and optional scale to measurements, relations, or regions
Logic (2)	`calculator`, `count_objects`	Expressions or image queries to numerical outputs
Statistics (5)	`coverage_ratio`, `density_map`, `size_distribution`, `classify_pattern`, `zonal_stats`	Images, points, or boxes to ratios, distributions, or maps
Aggregation (3)	`aggregate_stats`, `loop_over_regions`, `rank_regions`	Values or regional operations to summaries or rankings
Validation (3)	`validate_result`, `cross_check`, `check_gsd_conversion`	Values and constraints to consistency checks
External retrieval (1)	`web_search`	Query to an empty offline response
Skill access (2)	`read_skill`, `read_skill_reference`	Document identifier and optional reference name to procedural text

Perception tools and `count_objects` produce model-based observations. In particular, `segment` returns an estimated coverage ratio and region boxes. Spatial calculations operate on the supplied coordinates, and `coverage_ratio` applies color thresholds. Questions do not provide GSD. Distance and area tools report pixel measurements unless the agent supplies a scale for conversion. Aggregation and validation use deterministic operations, although a regional loop can invoke a perceptual tool. External retrieval returns an empty offline response in the evaluation configuration.

C.3 PROMPTS AND RUNTIME SETTINGS

The shared prompt describes the action format, tool signatures, sequential evidence collection, and final option selection. It also includes scale-conversion and buffer reminders. NS retains these common instructions. WS appends the directory and selection instructions. SG appends the first-turn planning requirement. The following excerpts show the action format and condition-specific instructions:

Shared action format

```
{"thought": "your reasoning for this step (1-3 short sentences)",
 "actions": [{"name": "<tool_name>", "arguments": {"...": "..."}}]}
```

WS selection instruction

Skim only the `description` fields and pick the smallest set of remote-sensing skills that actually apply. Call `read_skill` on those, and only those.

SG first-turn format

```
{"thought": "PLAN: <numbered procedure tailored to this question>",
 "actions": [{"name": "NoOp", "arguments": {}}]}
```

For the 1,385-question evaluation set, we analyze GPT-5.6-Terra, Qwen3.5-9B, and Qwen2.5-VL-72B-Instruct (abbreviated Qwen2.5-VL-72B). Each model has one recorded rollout per question under each condition, totaling 4,155 rollouts per model and 12,465 overall. Table C.3 lists the recorded settings. Each question-condition pair contributes one answer per model. The models use different serving and pipeline configurations, so cross-model comparisons describe these recorded configurations. Appendix E reports the corresponding results and execution statistics.

Table C.3: Recorded execution and analysis settings. Model-specific values are identified in the table.

Setting	Value
Evaluation set	1,385 questions across 13 task types
Rollouts per question	NS: 1, WS: 1, SG: 1
Interaction budget	12 rounds, including termination
Temperature	GPT: unset (`null`), effective value unrecorded. Both Qwen models: 0.7
Configured generation limit	4,096 tokens
Input-image long-side limit	2,048 pixels
Image-size budget	4 MiB per image and 4 MiB in total
GPT request timeout	120 seconds
GPT configured retry attempts	3
Bootstrap analysis	20,000 resamples; random seed 20260926

C.4 SCORING AND BEHAVIORAL MEASURES

Answer parsing accepts a single option letter or the first standalone A–D letter in the terminal answer, ignoring case. Missing or unparseable answers score zero. Accuracy is the unweighted mean of exact-match scores across the 13 task types. Each dimension averages its constituent tasks. Task-level scores are reported separately.

For the NS/WS/SG comparisons, paired contrasts match the same questions across conditions. Writing accuracy as A, we define

$$\Delta_{\text{expert}} = A_{\text{WS}} - A_{\text{NS}}, \quad \Delta_{\text{plan}} = A_{\text{SG}} - A_{\text{NS}}, \quad \Delta_{\text{gap}} = A_{\text{WS}} - A_{\text{SG}}. \tag{C.1}$$

These contrasts measure gains from expert skill access and generated plans, and the difference between them. We use 20,000 bootstrap resamples with random seed 20260926 and take the 2.5th and 97.5th percentiles of the resulting accuracy differences as the 95% confidence interval. The bootstrap resamples paired questions with replacement within each task, preserves task sizes, and averages the 13 task-level differences. These intervals quantify question-sampling variability for the recorded predictions.

Skill-body access measures whether a WS rollout successfully retrieves a document through `read_skill`. Question coverage is the proportion of the 1,385 questions with at least one successful response containing a skill body. We report attempted and successful retrievals separately at the question and call levels. A question counts once in each applicable category, while call counts retain repeated requests. Reference-document retrieval through `read_skill_reference` is counted separately. Subsequent actions provide evidence of how retrieved procedures are used.

Traces record actions, arguments, observations, errors, and final answers. The recorded action count includes skill reads, `NoOp`, and parse-error entries, and excludes terminal actions. Calls nested inside a regional loop are represented by its outer action. Base-tool invocation counts exclude skill-access calls, `NoOp`, parse-error entries, and the valid terminal action `Terminate`. Failed attempts with unregistered names remain in the count, including lowercase `terminate`. The base-tool error rate is the proportion of these attempts that return an error. Budget exhaustion counts questions requiring forced finalization at the 12-round limit, and invalid answers count missing or unparseable final options. These question counts can overlap. We also report the mean number of interaction rounds per question.

A subset of 132 evaluation questions provides reference procedural descriptions as textual steps for qualitative inspection: 99 buffer and spatial query questions (T2.3) and 33 damage and risk assessment questions (T5.2).

D GeoSkillAgent: Evolution Details

GeoSkillAgent refines a reusable SkillBank through failure-guided updates while keeping model parameters fixed. This section details the update protocol in Section 4.

D.1 Training partitions and execution records

The 1,697 training questions are partitioned into 1,141 evolution questions, 230 validation questions, and 326 stability questions. Questions that share a source scene or an image are assigned to the same partition. Paired images are grouped together. All three subsets are disjoint from the final test set. At round t, E_t supplies questions for collecting execution trajectories, while V and R are fixed throughout evolution. Validation measures the benefit of an update. The stability subset checks overall accuracy, with per-task scores retained for analysis.

The state of a round consists of the accepted bank S_t, the evolution questions E_t, and the shared execution configuration. The bank is read-only during trajectory collection and candidate evaluation. Before a run, the configuration specifies the positive validation threshold δ_V, candidate ordering, and the stopping rule. Baseline and candidate runs share the model, prompt templates, tools, decoding settings, interaction budget, randomization protocol, and scoring procedure.

For image I_i and question q_i, execution with the frozen bank produces a trajectory τ_i and answer $\hat{y}_i$:

$$\begin{aligned}(\tau_i, \hat{y}_i) &= \mathrm{Rollout}_\theta(I_i, q_i; S_t), \qquad i \in E_t, \\ \tau_i &= \big[(r_{i,j}, a_{i,j}, o_{i,j})\big]_{j=1}^{T_i}.\end{aligned} \tag{D.1}$$

Here, θ denotes the fixed model parameters and T_i is the number of recorded steps. Each step stores the reasoning text $r_{i,j}$, tool call and arguments $a_{i,j}$, and observation $o_{i,j}$. Tool outputs, including generated images, become observations for subsequent reasoning. Skill-reading actions are retained to link retrieved procedures to subsequent execution.

D.2 Procedural patches and candidate banks

We score the predicted option letter $\hat{y}_i$ by exact matching against the reference y_i, assigning zero to missing or invalid answers. Let $\mathcal{F}_t \subseteq E_t$ index incorrect executions with usable records. Patch extraction uses these trajectories and the current bank to produce procedural corrections:

$$J(\hat{y}_i, y_i) = \mathbf{1}[\hat{y}_i = y_i], \qquad \mathcal{P}_t = \big\{\mathrm{Extract}(\tau_i, S_t) \mid i \in \mathcal{F}_t\big\}. \tag{D.2}$$

Extraction examines the retrieved skill guidance, tool arguments, operation order, and intermediate observations. It identifies missing steps, incorrect operations, or underspecified execution conditions.

Algorithm: failure-guided SkillBank evolution

1. Initialize $S_t \leftarrow S_0$. Fix V, R, $\delta_V > 0$, execution settings, and the stopping rule.
2. Freeze S_t and collect trajectories on E_t. Select failed runs with usable records.
3. Extract reusable patches. Group, deduplicate, and resolve updates to the same skill section.
4. Cache baseline outcomes on V and R. For each update $p_{t,i}$, construct $C_{t,i} = \mathrm{Apply}(\mathrm{Clone}(S_t), p_{t,i})$.
5. Evaluate each candidate on V and R. Retain updates satisfying $G(C_{t,i}; S_t) = 1$.
6. Merge retained updates in the fixed order. Commit the merged bank only if $G(\widetilde{S}_{t+1}; S_t) = 1$. Otherwise retain S_t.
7. Record all decisions. Repeat until the stopping rule is met, then freeze S^* for final testing.

Figure D.1: The evolution procedure. Individual candidates and their merged bank are evaluated against the same current SkillBank.

A patch contains four elements: the failure being corrected, the conditions under which it occurs, the target skill or section, and the revised procedure. Its source question and trajectory are stored separately. The reusable content removes question text, answer options, reference answers, image paths, and instance-specific coordinates. It retains general conditions, such as overlapping observation regions, and the operations needed to handle them. For example, the counting correction illustrated in Figure 5 requires checking overlap before combining regional counts.

Patches are grouped by failure mechanism and target section. A deterministic rule removes repeated corrections and resolves overlapping proposals. Each candidate specifies its applicability conditions, operations, and verification steps. Each update $p_{t,i}$ has one explicit operation: add a skill with a unique identifier, or revise a declared section of an existing skill. Applying the operation to an independent, complete copy of S_t preserves the remaining library and produces $C_{t,i} = \mathrm{Apply}(\mathrm{Clone}(S_t), p_{t,i})$. All individual candidates are evaluated against the same frozen baseline S_t.

D.3 Candidate and merged-bank validation

The outcomes of S_t on V and R are cached once per round. Each candidate is evaluated on both subsets. Acceptance requires a validation gain of at least $\delta_V > 0$ and non-decreasing overall accuracy on R. Per-task scores describe the distribution of changes. Equation 5 defines this acceptance rule on the held-out subsets.

Individually passing updates are provisional. They are merged in a fixed order, and the combined bank is evaluated again against S_t on the same subsets. The merge becomes S_{t+1} only if it passes the complete acceptance rule. Otherwise, S_t is retained. This second check tests interactions between updates that were useful in isolation. The initial bank S_0 provides an anchor for cumulative changes, while the current bank S_t determines whether an update is accepted.

D.4 Update records and final evaluation

An append-only record links each proposal to its source trajectories, target section, and candidate content. It stores grouping and conflict decisions, validation scores, overall and per-task stability scores, and the acceptance or rejection reason. Bank content hashes identify the versions before and after each update, together with the execution configuration used for comparison. Rejected proposals and failed merges remain in the record, allowing the sequence of accepted versions to be reconstructed.

After evolution, the final SkillBank S^* is frozen. The final comparison runs S_0 and S^* through the same skill-discovery protocol on the isolated test set. Model parameters, images, tools, prompt templates, decoding settings, interaction budget, attempt count, and scoring rules are held fixed. Patch generation and update selection use the training partitions. The final test comparison measures the effect of the evolved library.

E Additional Results and Analyses

This section reports the complete task results, paired outcomes, and execution diagnostics for the fixed 1,385-question evaluation set. Question counts and process rates retain their stated denominators. The evaluation conditions and statistical protocol are defined in Appendix C.

E.1 Per-task performance

Table E.1 expands the main comparison to every task. Task identifiers and names are listed in Table A.1. The direction and size of the WS change vary across models. For example, multi-condition filtering (T4.3) improves for GPT and Qwen2.5-VL-72B but declines for Qwen3.5-9B. Route planning (T4.6) declines under WS for all three models. Change detection (T3.1) improves in each model, while its accuracy remains between 9.33% and 24.67% across all nine settings. These task-level results motivate examining the operations required by individual questions.

Table E.1: **Accuracy on all 13 tasks (%).** NS, WS, and SG denote No Skills, Expert Skill Discovery, and Self-generated Plan. The last row is the unweighted task mean.

		GPT-5.6-Terra			Qwen3.5-9B			Qwen2.5-VL-72B		
Task	n	NS	WS	SG	NS	WS	SG	NS	WS	SG
T1.2	122	37.70	36.89	36.07	31.97	36.07	32.79	34.43	33.61	37.70
T1.3	82	46.34	48.78	43.90	53.66	56.10	58.54	54.88	50.00	51.22
T2.1	150	28.00	34.00	38.00	19.33	17.33	24.00	16.67	18.67	17.33
T2.3	99	53.54	55.56	57.58	39.39	48.48	55.56	48.48	47.47	48.48
T2.4	136	72.79	58.82	66.91	36.03	38.97	36.76	38.24	42.65	33.82
T3.1	150	10.67	12.00	9.33	16.00	18.67	18.00	22.67	24.67	23.33
T4.1	130	54.62	53.85	56.92	38.46	52.31	39.23	32.31	33.08	36.15
T4.2	34	32.35	32.35	32.35	38.24	41.18	35.29	38.24	44.12	41.18
T4.3	54	72.22	79.63	74.07	53.70	48.15	48.15	31.48	40.74	42.59
T4.5	85	29.41	35.29	31.76	31.76	29.41	29.41	27.06	24.71	22.35
T4.6	150	22.00	20.00	17.33	24.00	21.33	29.33	28.67	26.00	31.33
T5.1	43	79.07	86.05	88.37	46.51	48.84	51.16	32.56	32.56	44.19
T5.2	150	19.33	20.00	18.00	23.33	24.00	24.67	26.67	20.67	22.00
Accuracy	1,385	42.93	44.09	43.89	34.80	36.99	37.15	33.26	33.76	34.75

E.2 Paired improvements, regressions, and uncertainty

A small net change can combine many corrected and regressed answers. For task t with n_t questions, let $n_{01,t}$ count answers that change from incorrect under NS to correct under WS, and let $n_{10,t}$ count the reverse. Their contributions to the accuracy difference, in percentage points, are

$$G = \frac{100}{13}\sum_{t=1}^{13}\frac{n_{01,t}}{n_t}, \qquad L = \frac{100}{13}\sum_{t=1}^{13}\frac{n_{10,t}}{n_t}, \qquad \Delta_{\mathrm{WS-NS}} = G - L. \tag{E.1}$$

Figure 6(a) visualizes these contributions. Table E.2 also reports the underlying question counts. For Qwen2.5-VL-72B, WS has one fewer correct answer overall but a positive accuracy difference. This reflects the unequal task sizes and the equal weighting of tasks in our primary metric.

Table E.2: **Paired NS-to-WS changes.** Corrected and regressed columns count questions. G, L, and the net difference are accuracy percentage points.

Model	Corrected	Regressed	G	L	Net
GPT-5.6-Terra	168	164	12.10	10.93	+1.17
Qwen3.5-9B	228	195	16.60	14.41	+2.19
Qwen2.5-VL-72B	191	192	14.51	14.00	+0.51

Figure 6(b) compares WS-only and SG-only successes. Table E.3 adds their shared outcomes. These are differences in observed outcomes from one rollout per condition. Across all three models

and all three conditions, 229 questions (16.53%) remain incorrect. This shared set includes 62 change-detection questions, 33 damage-assessment questions, and 33 route-planning questions.

Table E.3: **Distinct and shared success sets.** WS-only and SG-only compare those two conditions. Both correct means both WS and SG are correct. All three wrong additionally includes NS. Every entry counts questions.

Model	WS-only	SG-only	Both correct	All three wrong
GPT-5.6-Terra	164	166	376	583
Qwen3.5-9B	217	223	250	585
Qwen2.5-VL-72B	199	207	238	620

Table E.4 gives the task-stratified paired bootstrap intervals described in Appendix C.4. We use 20,000 resamples with seed 20260926, resampling matched questions within each task and averaging the resulting task differences. All nine intervals include zero.

Table E.4: **Accuracy differences and 95% paired bootstrap intervals.** All values are percentage points. The last contrast is WS minus SG.

Model	WS−NS	SG−NS	WS−SG
GPT-5.6-Terra	$+1.17\ [-1.60, 3.96]$	$+0.97\ [-1.73, 3.76]$	$+0.20\ [-2.50, 2.89]$
Qwen3.5-9B	$+2.19\ [-1.09, 5.46]$	$+2.35\ [-0.92, 5.58]$	$-0.16\ [-3.52, 3.17]$
Qwen2.5-VL-72B	$+0.51\ [-2.68, 3.66]$	$+1.49\ [-1.73, 4.77]$	$-0.98\ [-4.26, 2.34]$

E.3 Skill access and execution diagnostics

The WS directory provides all models with skill descriptions. We distinguish attempted retrieval from a successful tool response containing a skill body (Table E.5). Successful body access ranges from 0.14% to 91.12% of questions. Qwen2.5-VL-72B attempts retrieval on 71 questions and succeeds on 68. Its three failed calls request nonexistent skill identifiers. It also attempts reference-document retrieval on four questions and succeeds on two. The other models do not request reference documents. A successful response measures access to a document, while adoption of its procedure requires examining subsequent actions. Questions without body retrieval still receive the directory descriptions.

Table E.5: **Skill-body access under WS.** Question coverage uses 1,385 as its denominator. A question is successful if at least one body request succeeds. Calls count retrieval attempts and successful responses separately.

	Questions			Calls	
Model	Attempted	Successful	Coverage (%)	Attempted	Successful
GPT-5.6-Terra	1,262	1,262	91.12	1,315	1,315
Qwen3.5-9B	2	2	0.14	4	4
Qwen2.5-VL-72B	71	68	4.91	77	74

Table E.6 separates tool failures, budget exhaustion, and invalid answers. Base-tool attempts exclude skill and reference reads, `NoOp`, parse-error placeholders, and valid `Terminate` calls. Calls to unregistered names remain in the denominator and count as errors. Error rates use base-tool calls as the denominator. Budget exhaustion and invalid answers count questions, and the two counts can overlap. An empty but valid detection result is not a tool error.

The process measures capture different failure modes. Qwen3.5-9B has fewer base-tool errors under WS than NS, yet reaches the round limit more often. Qwen2.5-VL-72B has 31 invalid WS answers. All 31 terminate before the budget limit without reading a skill body or reference document, and 23 have no base-tool error. Thus, fewer invocation errors do not by themselves establish completed reasoning or a valid answer. These records motivate measuring access, execution, and answer validity separately.

Table E.6: **Execution diagnostics for all conditions.** Rounds are means per question. Calls and errors refer to base tools. Budget counts forced finalization at the 12-round limit. Invalid counts missing or unparseable final options.

Model	Setting	Rounds	Calls	Errors	Error (%)	Budget	Invalid
GPT-5.6-Terra	NS	3.38	3,272	141	4.31	30	0
	WS	4.09	2,938	288	9.80	18	0
	SG	4.43	3,352	207	6.18	36	0
Qwen3.5-9B	NS	9.03	10,894	3,291	30.21	661	18
	WS	9.57	11,595	2,431	20.97	750	13
	SG	8.68	9,595	2,938	30.62	596	12
Qwen2.5-VL-72B	NS	5.56	6,312	1,411	22.35	102	5
	WS	5.65	6,322	1,865	29.50	132	31
	SG	6.72	6,506	1,504	23.12	179	9

E.4 GeoSkillAgent refinement and ablations

GPT-5.6-Terra executes the agent, and Claude Opus 4.6 generates patches. The paired refinement experiment compares S_0 and S_1 on the same 1,385 questions. Table E.7 gives their task results. The component ablations in Table 3 use their own S_0 run as the baseline.

Paired refinement. Accuracy increases from 38.97% to 46.53%. Of the paired answers, 185 change from incorrect to correct and 97 change in the reverse direction. Twelve tasks improve and one is unchanged. Base-tool calls decrease from 2,809 to 2,688, tool errors from 100 to 94, and budget-exhausted questions from 16 to 11.

Table E.7: **Paired refinement accuracy (%).** S_0 is the initial expert library and S_1 the evolved library.

Task	S_0	S_1	Task	S_0	S_1
T1.2	36.07	39.34	T4.2	35.29	35.29
T1.3	43.90	50.00	T4.3	66.67	79.63
T2.1	28.00	36.67	T4.5	31.76	35.29
T2.3	42.42	58.59	T4.6	18.00	26.00
T2.4	60.29	66.91	T5.1	62.79	90.70
T3.1	12.67	13.33	T5.2	18.00	19.33
T4.1	50.77	53.85	Accuracy	38.97	46.53

Ablation definitions. The stability ablation removes the overall stability check while retaining the validation-gain requirement. The merged re-evaluation ablation omits the combined-bank check after candidate screening. Random- and success-trace induction change the trajectory source. Sham updates match update size. Test-task regressions count tasks whose accuracy falls below the corresponding S_0 run. The gate operates on the held-out stability set.

Component effects. Full reaches 46.28% accuracy, 1.95 points above its 44.33% initialization. Removing stability screening lowers accuracy by 0.46 points and increases regressed tasks from two to seven. Removing merged re-evaluation lowers accuracy by 0.96 points, with two regressed tasks in both settings. Random- and success-trace induction score 1.44 and 1.82 points below Full. Sham updates match S_0 at 44.33%, with one regressed task. Equal aggregate scores can coexist with different task-level outcomes.

F Qualitative Examples and Failure Analysis

We inspect paired executions to connect outcome differences with four UHR requirements: maintaining the observation region across scales, evaluating spatial relations, checking joint constraints, and establishing temporal object correspondence. We manually review 16 question–model cases under all

three conditions, giving 48 trajectories. These cases were selected purposively from 54 candidates to cover improvements, regressions, SG-only successes, and shared failures. The sample supports qualitative comparison of execution patterns. A separate audit checks outcomes and process counts across all 12,465 trajectories.

The analysis uses questions, tool arguments, returned observations, and final answers. Correctness means agreement with the expert-validated benchmark answer. Some stored observations are truncated. Within WS, we distinguish directory access, successful document retrieval, and subsequent procedural actions. Table F.1 identifies every reviewed case.

F.1 Region tracking in boundary-restricted counting

The tennis-court question `T1_2_MM1_0008` asks for court surfaces fully inside a red circle. GPT under WS reads the counting document, localizes the circle in round 2, and requests a crop. After correcting the crop argument format, it obtains a local view and selects A, the reference answer of one court. NS selects C, corresponding to two courts. SG also selects C after successive crops map to increasingly small source regions, followed by a description returning two courts. The traces contrast WS's boundary localization with SG's loss of spatial scope across crops. WS returns the correct option without a separate court-by-court boundary checklist in the trace.

F.2 Spatial grounding and answer mapping

For `T2_1_MM1_0002`, GPT must locate an orange-roofed building relative to a circular plaza. WS retrieves the visual-grounding document and obtains a bounding box for each object. Their returned centers differ in both the rightward and downward directions, yet WS selects B, to the right. SG obtains similar boxes and selects the reference option D, bottom-right. NS also selects D after a description request. The discrepancy occurs after object localization and concerns the relation or its mapping to an option. Counting localization calls alone would miss this difference.

F.3 Joint constraints and conflicting candidate evidence

The quayside question `T5_1_MV_1052` requires paved ground, direct adjacency to navigable water, and exclusion from a dry dock. GPT under WS reads the multi-condition document and explicitly requests all three checks for each candidate. The returned table marks both C and D as eligible and recommends D. WS follows this recommendation, while NS and SG select the reference answer C. All conditions are present in the WS request, so omission of a constraint does not explain this regression. The unresolved step is distinguishing candidates using their visual attributes and boundary relations.

A complementary example is Qwen3.5-9B on `T5_1_MV_0415`. Its WS trajectory queries each of four candidates against open ground, railway proximity, and residential separation in rounds 5–8, then selects the reference answer C. NS and SG reach the round limit and answer incorrectly. WS completes the candidate-wise checks with directory descriptions available, without retrieving a skill body.

F.4 Recovery after an empty detection result

For red-roofed houses in woodland, `T1_2_MM1_0412`, GPT under SG requests a crop and then detects red-roofed houses. The detector returns zero objects with no tool error. SG next asks for a description with counts and location cues. The response identifies three houses, and SG selects the reference answer B in round 5. NS and WS select D at the 12-round limit, despite WS having read the counting document. This case illustrates an alternative observation route after an empty result. A valid zero detection and a failed tool invocation require different recovery actions.

F.5 Temporal comparison and view-state consistency

Question `T3_1_MEbi_0727` asks for minor and major building damage, excluding intact and destroyed buildings. GPT under NS selects the reference answer B. WS first reads the condition-assessment document, then calls `change_detect` with `image` instead of the required `image1`.

It corrects this argument in the next round but ultimately selects D. SG uses identical normalized crop coordinates for the two dates, yet the crop observations report different source regions. It later encounters an invalid argument encoding while iterating over regions and selects C at the round limit. These records expose interface and view-state continuity problems.

F.6 SCENE CHANGE VERSUS OBJECT-LEVEL DAMAGE

All three Qwen2.5-VL-72B conditions select A for `T3_1_MEbi_0528`, whose reference answer is B. WS reads the temporal-comparison document, computes an image difference, and calls change detection. The latter reports flooding. Two crop requests fail because of argument formatting, and subsequent detection returns three buildings before the event and none afterward. NS and SG also obtain an empty post-event detection. The trajectories provide scene-level changes but no building correspondences with individual damage labels. This task requires matching buildings across dates and assigning object-level damage categories.

Table F.1: Index of all 16 manually reviewed question–model cases. IDs omit the common `Q_PL_` prefix. GPT, Q9, and Q72 denote GPT-5.6-Terra, Qwen3.5-9B, and Qwen2.5-VL-72B. Answers follow NS/WS/SG order. Ref. is the benchmark option, and ∅ denotes an invalid answer.

Model	Question ID	Ref.	Answers	Observed comparison
GPT	`T2_1_MM1_0001`	B	D/B/D	WS and SG both localize objects, but receive different boxes.
GPT	`T2_1_MM1_0002`	D	D/B/D	Similar WS/SG locations lead to different direction options.
GPT	`T5_1_MV_0650`	C	A/C/C	All conditions request candidate comparisons, with different visual returns.
GPT	`T5_1_MV_1097`	C	D/C/D	WS answers after retrieval with no subsequent base-tool call.
GPT	`T5_1_MV_1052`	C	C/D/C	WS checks every constraint but follows an incorrect recommendation.
GPT	`T5_1_MV_0542`	D	B/B/D	SG adds a targeted comparison between two candidate areas.
GPT	`T1_2_MM1_0008`	A	C/A/C	WS localizes the boundary. SG crops lose the intended scope.
GPT	`T1_2_MM1_0010`	D	D/C/C	WS detection reaches its requested cap. Counts differ across tools.
GPT	`T1_2_MM1_0412`	B	D/D/B	SG switches to description after a valid empty detection.
GPT	`T3_1_MEbi_0226`	B	D/B/D	The correct WS option lacks a complete destroyed-building list.
GPT	`T3_1_MEbi_0727`	B	B/D/C	WS repairs an interface error. SG crops map to different regions.
Q72	`T2_1_MM1_0099`	D	B/D/A	Correct WS output follows a file error and coordinates without a successful localization record.
Q72	`T2_1_MM1_0056`	C	C/∅/D	WS returns no valid option in round 5, without a tool error.
Q72	`T3_1_MEbi_0566`	C	B/C/A	WS attempts the correct option before its first document read.
Q72	`T3_1_MEbi_0528`	B	A/A/A	Scene-change returns never establish object-level damage counts.
Q9	`T5_1_MV_0415`	C	B/C/A	WS checks all candidates without retrieving a document.