

ThinkGeo : Evaluating Tool-Augmented Agents for Remote Sensing Tasks

Anonymous CVPR submission

Paper ID

Abstract

001 *Recent progress in large language models (LLMs) has en-*
 002 *abled tool-augmented agents capable of solving complex*
 003 *real-world tasks through step-by-step reasoning. However,*
 004 *existing evaluations often focus on general-purpose or mul-*
 005 *timodal scenarios, leaving a gap in domain-specific bench-*
 006 *marks that assess tool-use capabilities in complex remote*
 007 *sensing use cases. We present **ThinkGeo**, an agentic bench-*
 008 *mark designed to evaluate LLM-driven agents on remote*
 009 *sensing tasks via structured tool use and multi-step planning.*
 010 *Inspired by tool-interaction paradigms, ThinkGeo includes*
 011 *human-curated queries spanning a wide range of real-world*
 012 *applications such as urban planning, disaster assessment*
 013 *and change analysis, environmental monitoring, transporta-*
 014 *tion analysis, aviation monitoring, recreational infrastruc-*
 015 *ture, and industrial site analysis. Queries are grounded*
 016 *in satellite or aerial imagery, including both optical RGB*
 017 *and SAR data, and require agents to reason through a di-*
 018 *verse toolset. We implement a ReAct-style interaction loop*
 019 *and evaluate both open and closed-source LLMs (e.g., GPT-*
 020 *4o, Qwen2.5) on 486 structured agentic tasks with 1,778*
 021 *expert-verified reasoning steps. The benchmark reports both*
 022 *step-wise execution metrics and final answer correctness.*
 023 *Our analysis reveals notable disparities in tool accuracy and*
 024 *planning consistency across models. ThinkGeo provides the*
 025 *first extensive testbed for evaluating how tool-enabled LLMs*
 026 *handle spatial reasoning in remote sensing.*

027 1. Introduction

028 Recent advances in LLMs have enabled the emergence of
 029 tool-augmented agents, systems that can break down com-
 030 plex tasks into step-by-step plans, invoke external tools (e.g.,
 031 vision modules, calculators, and code interpreters), and rea-
 032 son across intermediate states [27, 36]. This paradigm,
 033 popularized via ReAct-style frameworks [36], has shown
 034 promise in general-purpose settings through benchmarks
 035 like ToolBench [24], GAIA [21], and GTA [29], which eval-
 036 uate agents on procedural correctness, tool use, and final
 037 task outcomes. However, these benchmarks largely focus on

Benchmark	Real queries	Deployed tools	MM inputs	Annotation chains	Execution eval.	RS images
API-Bench [23]	✗	✗	✗	✗	✗	✗
ToolBench [24]	✗	✓	✗	✗	✗	✗
GAIA [21]	✓	✗	✓	✗	✓	✗
APIBank [18]	✗	✓	✗	✓	✗	✗
m&m's [20]	✗	✓	✓	✓	✓	✗
GTA [29]	✓	✓	✓	✓	✓	✗
ThinkGeo (Ours)	✓	✓	✓	✓	✓	✓

Table 1. Comparison of agentic benchmarks across key dimensions. ThinkGeo is the only benchmark designed specifically for remote sensing (RS), incorporating real EO imagery alongside ReAct-style annotation chains and deployed tools. It uniquely supports spatial reasoning and remote sensing-specific tasks through geospatial grounded inputs and execution-level evaluation for studied models.

synthetic, open-domain, or web-grounded scenarios, leav- 038
 ing the question of agentic capability in precision-critical, 039
 spatially grounded domains, like remote sensing, largely un- 040
 explored. Remote sensing (RS) is critical to a wide range 041
 of applications, including environmental monitoring, urban 042
 infrastructure and transportation analysis, disaster response, 043
 and land-use mapping, with an ever-growing stream of high- 044
 resolution imagery from earth observation (EO) satellites 045
 and drones [15]. Despite advances in visual models for 046
 detection, segmentation, and change analysis, current pro- 047
 cessing pipelines remain brittle and manually engineered 048
 across tasks. Integrating these capabilities into LLM-driven 049
 agents demands reasoning over geodetic metadata, spatial 050
 resolutions, temporal dynamics, and unit-aware calculations. 051
 Existing agentic benchmarks (e.g., GTA [29], GAIA [21]) 052
 do not address these demands; they are built around general- 053
 purpose or web-grounded images, lacking the spatial fidelity 054
 and grounding required for geospatial workflows. Conse- 055
 quently, there is a pressing need for a benchmark that evalu- 056
 ates tool-augmented agents in remote sensing contexts, for 057
 reasoning over real EO imagery, coordination of general- 058
 purpose visual tools, and handling spatially grounded multi- 059
 step tasks. In this work, we introduce **ThinkGeo**, the first 060
 agentic benchmark specifically designed to evaluate tool- 061
 augmented LLM agents on realistic remote sensing tasks. 062
 As shown in Table 1, unlike existing agentic benchmarks 063

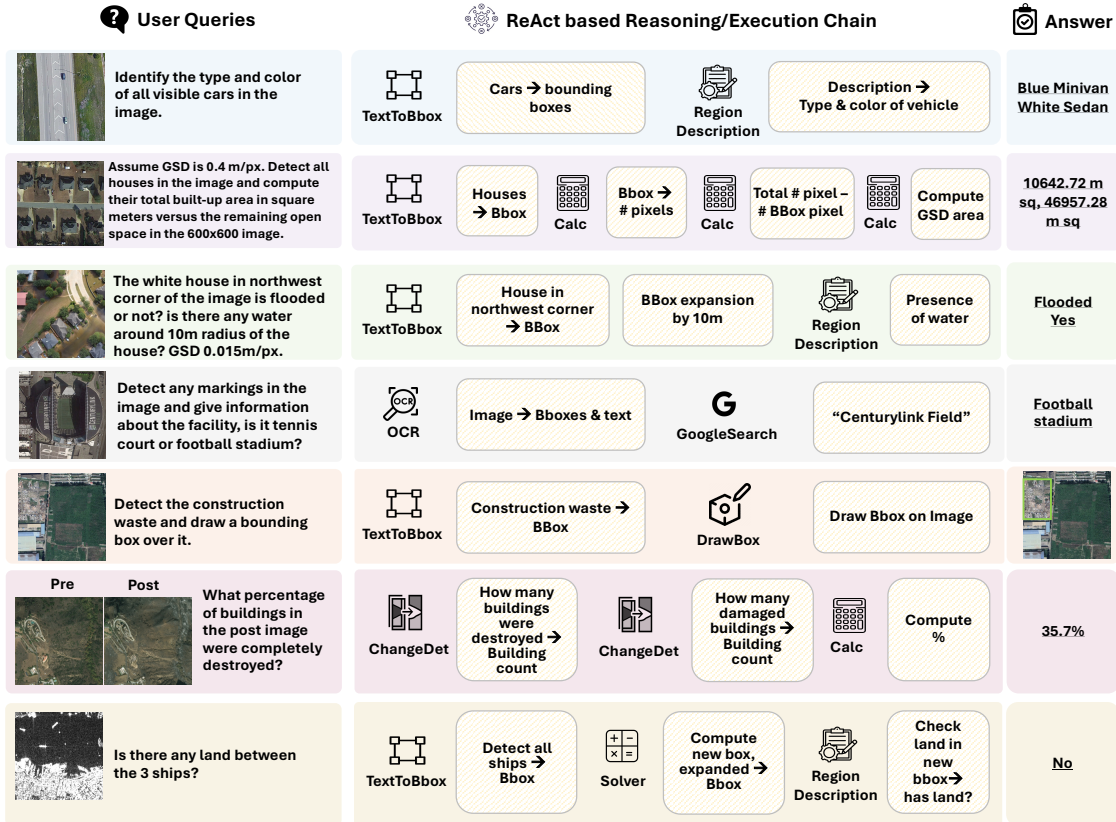


Figure 1. Representative samples from the ThinkGeo benchmark. Each row illustrates a user query grounded in real RS imagery, followed by a ReAct-based execution chain comprising tool calls and reasoning steps, and concludes with the resulting answer. The examples span diverse use cases, including transportation analysis, urban planning, disaster assessment and change analysis, recreational infrastructure, and environmental monitoring, highlighting multi-tool reasoning and spatial task complexity.

064 built on general or web-grounded images, ThinkGeo focuses
 065 on spatially grounded reasoning, requiring agents to plan
 066 and execute multi-step workflows using satellite and aerial
 067 imagery. Each query is coupled with an executable tool en-
 068 vironment and annotated with structured evaluation signals,
 069 enabling rigorous assessment of perception, planning, and
 070 geospatial reasoning under tool-based execution constraints.
 071 Our main contributions are as follows:

072 • **Task Suite & Dataset:** A curated set of 486 agentic tasks
 073 with 1,778 expert-verified reasoning steps over medium
 074 to high-resolution optical RGB (436 tasks) and SAR (50
 075 tasks) images, spanning urban, environmental, transporta-
 076 tion, aviation, industrial, change detection, and disaster-
 077 related scenarios. Examples are shown in Fig 1.
 078 • **Executable Tool Set:** An extended suite of 14 tools
 079 designed to simulate real-world RS workflows. This
 080 includes perception modules (e.g., ObjectDetection,
 081 SegmentObjectPixels, ChangeDetection),
 082 logic and numeric tools (e.g., Calculator, Solver,
 083 Plot), and visualization aids (e.g., DrawBox,
 084 AddText).

085 • **Evaluation Protocol:** We propose two evaluation modes,
 086 step-by-step and end-to-end, paired with fine-grained met-
 087 rics to assess instruction adherence, tool use correctness,
 088 argument formatting, multi-step reasoning, and final an-
 089 swer accuracy.

090 • **Benchmarking Study:** A comparative evaluation of
 091 state-of-the-art LLM agents, including GPT-4o, Claude-
 092 3, Qwen-2.5, and LLaMA-3, revealing persistent gaps in
 093 multimodal tool reasoning and execution trace alignment,
 094 even among top-performing models.

095 By grounding agentic evaluation in real EO imagery
 096 and requiring interpretable, tool-based interaction tracking,
 097 ThinkGeo provides a new foundation for benchmarking and
 098 ultimately providing insights to improving spatially-aware,
 099 tool-augmented LLM agents for geospatial analysis.

2. Related Work 100

101 **Tool-augmented LLM Agents and Benchmarks:** Integrat-
 102 ing large language models (LLMs) with executable tools has
 103 recently become a central focus in agent research. Early
 104 work presented tool use as an alternating planning and ex-

105	ecution. ReAct, for instance, interleaves "thought" tokens	158
106	with structured tool calls, enabling a single LLM to both	159
107	reason and act [36]. Subsequent systems generalized this	160
108	idea to larger tool repertoires. HuggingGPT employs a GPT	161
109	controller to select from hundreds of vision, speech, and lan-	162
110	guage models exposed as functions [27], while Visual Chat-	163
111	GPT and MM-ReAct demonstrate analogous pipelines for	164
112	multimodal perception tasks [31, 35]. To measure tool-use	165
113	proficiency, several benchmarks have been proposed. Tool-	166
114	Bench, APiBench, and API-Bank evaluate single-step API	167
115	invocation within synthetic prompts [18, 23, 24]; m&m's	
116	extends this to multi-step multimodal settings [20]. More	
117	recently, GAIA [21] and GTA [29] introduced human-written,	
118	step-implicit tasks paired with executable tool chains, reveal-	
119	ing substantial performance gaps: GPT-4 completes fewer	
120	than half of GTA queries once real tools and intermediate	
121	checks are enforced. MLGym casts the agent problem into	
122	a Gym environment for open-ended AI-research workflows,	
123	highlighting long-horizon planning and code execution and	
124	also without geospatial imagery [22].	
125	Remote Sensing Agents: Recent efforts to extend LLM	
126	agents into EO have produced diverse tool-augmented	
127	pipelines, yet planning transparency, and step-level reason-	
128	ing fidelity remain limited. Remote Sensing ChatGPT [9]	
129	and RS-Agent [34] represent early vision-language pipelines	
130	that chain pretrained detectors, segmenters, and geospatial	
131	utilities under GPT-based planners. However, they typically	
132	report only final answer accuracy, omitting structured ReAct-	
133	style trace evaluation or step-wise error attribution. TreeGPT	
134	and GeoMap-Agent [7, 11] introduce domain-specific agents	
135	for forestry and geological mapping, respectively. While	
136	these systems operate over visual maps and structured vi-	
137	sual inputs, they rely on template-grounded or qualitative	
138	responses and do not implement formal multi-step evalu-	
139	ation. UnivEARTH [15], by contrast, employs a purely	
140	language-based framework that requires LLMs to generate	
141	valid Google Earth Engine (GEE) code, revealing that over	
142	58% of completions fail to execute and that even the best	
143	agents answer only around 33% of geospatial queries cor-	
144	rectly. Together, these works suggest that while EO agents	
145	can interface with rich toolsets, failures in tool selection,	
146	argument grounding, and spatial unit reasoning persist, un-	
147	derscoring the need for benchmarks that explicitly evaluate	
148	tool-level correctness alongside geospatial task outcomes.	
149	Evaluation Protocols: Early benchmarks for tool-	
150	augmented LLMs, such as ToolBench [24], APiBench [23],	
151	and API-Bank [18], primarily evaluate single-step tool usage	
152	in synthetic or isolated API call settings. While useful for	
153	measuring basic tool and argument prediction, these setups	
154	lack support for multi-tool planning, intermediate tracking,	
155	or long-horizon reasoning. To address these limitations, GTA	
156	[29] presents a tightly scoped yet richly instrumented bench-	
157	mark requiring sequential tool usage across perception, logic,	
	operation, and generation modules. GTA adopts a ReAct-	
	style interface and introduces fine-grained supervision for	
	each agent step, reporting metrics like ToolAcc, ArgAcc,	
	StepAcc, and final answer correctness, thereby uncovering	
	latent failure modes in tool selection and planning. Com-	
	plementing this, MLGym [22] reframes agent evaluation	
	as multi-task episodic learning within a Gym-style environ-	
	ment, simulating end-to-end ML workflows (e.g., training,	
	evaluation, reporting) that demand persistent memory and	
	adaptive behavior.	
	3. ThinkGeo Benchmark	168
	In this section, we present the design of ThinkGeo , a bench-	169
	mark designed to evaluate tool-augmented LLM agents in the	170
	context of remote sensing. ThinkGeo focuses on spatially	171
	grounded reasoning tasks that require agents to interpret op-	172
	tical EO imagery, plan multi-step tool usage, and produce	173
	geospatially coherent outputs. We describe our core <i>design</i>	174
	<i>goals</i> , define the <i>use case categories</i> that span both primary	175
	and supporting remote sensing tasks, detail the <i>query con-</i>	176
	<i>struction pipeline</i> , and provide a summary of the <i>integrated</i>	177
	<i>datasets</i> and task coverage.	178
	3.1. Design Goals	179
	Geospatial Reasoning: Tasks are modeled after real-world	180
	use cases in environmental monitoring and disaster response.	181
	Queries reflect practical challenges such as measuring metre-	182
	scale distances, counting structures within spatial buffers,	183
	and identifying features of damaged buildings. These tasks	184
	require fine spatial fidelity, unit-based reasoning, and visual	185
	attribute grounding, capabilities often overlooked in existing	186
	benchmarks such as GAIA [21] or ToolBench [24].	187
	Step-Implicit Tool Use: Unlike benchmarks where tool use	188
	is predefined or explicitly mentioned (e.g., APiBench [23]),	189
	ThinkGeo presents <i>step-implicit</i> , <i>tool-implicit</i> queries.	190
	Prompts do not reference tools by name; agents must infer	191
	which modules (e.g., perception, logic, operation) are	192
	needed and in what order. This design promotes true agentic	193
	planning and aligns with ReAct-style decision traces as used	194
	in GTA [29].	195
	3.2. Use Case Categories	196
	ThinkGeo is organized into seven primary categories, each	197
	reflecting critical application domains within the remote	198
	sensing ecosystem. These include <i>Urban Planning</i> , <i>Disaster</i>	199
	<i>Assessment & Change Analysis</i> , <i>Environmental Monitoring</i> ,	200
	<i>Transportation Analysis</i> , <i>Aviation Monitoring</i> , <i>Recreational</i>	201
	<i>Infrastructure</i> , and <i>Industrial Sites</i> . Fig. 2 shows the use case	202
	taxonomy. Each category encapsulates a range of spatially	203
	grounded, tool-invoking subtasks inspired by operational	204
	workflows in urban analytics, environmental science, and	205
	infrastructure planning:	206



Figure 2. **Use case taxonomy.** The benchmark spans seven major domains: Urban Planning, Disaster Assessment & Change Analysis, Environmental Monitoring, Transportation Analysis, Aviation Monitoring, Recreational Infrastructure, and Industrial Sites. Each domain includes representative task types requiring multimodal reasoning, spatial analysis, and tool-augmented execution.

- *Urban Planning* tasks involve residential layout analysis, swimming pool and parking area detection, road and bridge mapping, accessibility assessment, and proximity-based zoning.
- *Disaster Assessment & Change Analysis* includes multi-temporal damage comparison across disaster events like floods, hurricanes, wildfires, and volcanoes, featuring change detection, categorical damage classification (e.g., no-damage, minor, major, destroyed), area-based summaries, and quadrant-level spatial reports.
- *Environmental Monitoring* spans water-body adjacency, garbage and construction waste detection, contamination risk evaluation, and agricultural land-use impact assessments.
- *Transportation Analysis* covers vehicle detection and counting, heading direction estimation, traffic flow characterization, and distance-based safety checks across roads and intersections.
- *Aviation Monitoring* includes aircraft identification and categorization, runway occupancy, gate-terminal proximity analysis, and airfield layout planning.
- *Recreational Infrastructure* tasks address playground identification (e.g., basketball, baseball, tennis, and football fields), orientation detection, and coverage estimation.
- *Industrial Sites* focus on storage tank localization, diame-

ter and area measurement, and spatial relation mapping to adjacent operational zones.

These categories serve as testbeds for evaluating diverse capabilities such as multimodal reasoning, fine-grained spatial understanding, tool composition, and temporal change detection. By covering both canonical and underexplored use cases, ThinkGeo supports a systematic, application-driven evaluation of agentic LLM pipelines for real-world geospatial intelligence.

3.3. Query Construction Pipeline

To evaluate the capabilities of agentic systems in solving realistic remote sensing problems, we curate a diverse set of *complex queries*, defined as prompts that are concise and natural for humans but require agents to perform multi-step reasoning across multiple tools. These queries cannot be answered by the invocation of a single tool in isolation and instead test the agent’s ability to plan and compose a coherent sequence of actions. We implement a semi-automated query generation pipeline.

Step 1: Data Sampling & Guidelines. We begin by curating high-quality samples from diverse RS datasets. Domain experts provide task-specific guidelines and generate initial reference queries to bootstrap the selection process.

Step 2: Authoring ReAct Format. Using these guidelines, annotators manually inspect imagery, identify key objects and spatial relationships, and construct natural language queries following the ReAct format [36]. This involves composing a user query that implies multi-step reasoning, manually annotating missing elements, and generating a semi-structured dialog trace (thoughts, tool calls, observations, answers). The process is supported by a script built on the OpenAI GPT API. This script leverages per-image metadata (e.g., object types, GSD, bounding boxes) and tool definitions to generate diverse, tool-requiring prompts.

Step 3: Validation. All generated samples are verified through a two-stage validation protocol. First, expert reviewers assess the semantic correctness, relevance, and alignment with the toolset. Second, we apply script-based checks to validate tool argument consistency, dialog structure, and completeness. Invalid samples are manually refined and corrected before inclusion in the final dataset. Fig. 3 shows the end-to-end dataflow for constructing our ThinkGeo benchmark.

Additional Details. Beyond the core steps described above, the query construction pipeline incorporates several design elements to ensure scale, diversity, and inference robustness:

- **Query Diversity:** For each image, we generate 1-5 queries varying in spatial relationships, counting logic, or temporal comparisons, ensuring broad coverage of tool use compositions and reasoning patterns within the same scene. These queries span a range of agentic reasoning complex-

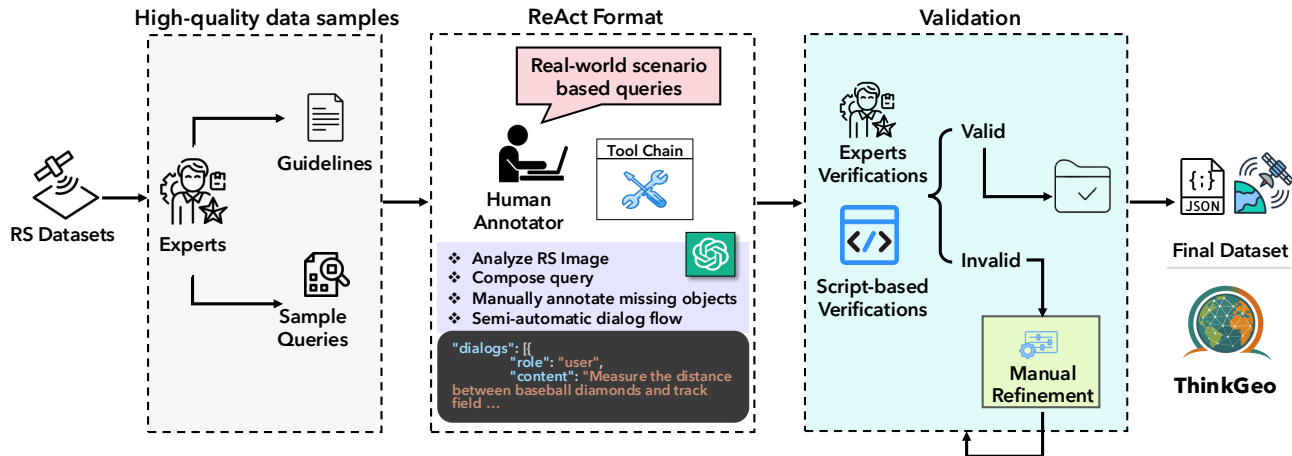


Figure 3. End-to-end dataflow for constructing the ThinkGeo benchmark. We begin with expert-curated samples from remote sensing datasets, guided by scenario-specific query design templates. Human annotators inspect images and generate ReAct-style multi-step queries using a semi-automated GPT-powered interface. Each query is validated via expert review and script-based consistency checks. Invalid cases are manually refined. The final dataset consists of JSON-formatted ReAct traces grounded in satellite or aerial imagery.

Table 2. Remote sensing datasets used as image sources in the construction of the ThinkGeo benchmark. These datasets span a wide range of applications, sensor resolutions, and annotation types. Notably, the agentic tasks defined on these images are newly annotated bottom-up.

Name	Tasks	Annotation Type	Sensor (Res)	Year
Optical RGB Datasets				
DOTA [33]	Monitoring Transport, Aviation, Infrastructure	GSD, B-Box, Category	(0.1–1)m/px	2021
NWPU-VHR-10 [6]	Monitoring Transport, Aviation, Infrastructure	B-Box, Category	(0.5–2)m/px	2023
UCAS-AOD [39]	Monitoring Transport, Aviation	B-Box, Category	(0.5–2)m/px	2015
AID [32]	Urban Planning, Monitoring Transport, Industr. Sites	B-Box, Category	(0.2–2)m/px	2017
iSAID [30]	Monitoring Transport	GSD, B-Box, Seg. Map, Pixel Count	(0.1–1)m/px	2019
xBD [10]	Disaster Assessment & Change Analysis	GSD, B-Box, Category, Pixel Count	(1–3.5)m/px	2019
FloodNet [26]	Urban Planning, Disaster, Transport Analysis	GSD, B-Box, Category, Seg. Map, Pix. Count	(0.015–0.02)m/px	2020
Global-Dumpsite [28]	Environmental Monitoring	B-Box, Category	(0.3–0.8)m/px	2023
SAR Datasets				
SSDD [38]	Monitoring Transport	B-Box, Category	(1–15)m/px	2021
SADD [37]	Aviation	B-Box, Category	(0.5–3)m/px	2022
SIVED [19]	Monitoring Transport	B-Box, Category	(0.1–0.3)m/px	2023

284 ity and tool invocation diversity, capturing variations in
 285 interaction depth and logical structure across tasks.
 286 • **Difficulty Annotation:** We categorized the queries into
 287 easy and hard based on two criteria: the presence of com-
 288 plex keywords and the reasoning steps required. Queries
 289 containing terms such as “estimate,” “compare,” “distri-
 290 bution,” “count,” “area,” “how many,” “orientation,” and
 291 “proximity,” along with more reasoning steps, were con-
 292 sidered harder. To organize the queries, we sorted them
 293 based on the count of complex keywords and the number
 294 of steps. Queries appearing earlier in the sorted list, with
 295 fewer complex keywords and shorter reasoning steps, were
 296 labeled as easy, while the rest were classified as hard.
 297 • **Inference-Aligned Prompting:** Prompts are designed
 298 such that the agent must recover the reasoning chain with-

299 out relying on field names or explicit tool indicators, pro-
 300 moting alignment with real-world, instruction-following
 301 behavior. This adheres to the tool-implicit design philoso-
 302 phy established in agentic benchmarks like GTA [29].

303 This modular structure supports robust and scalable gener-
 304 ation of diverse queries, enabling ThinkGeo to serve as a
 305 high-coverage benchmark for multimodal, tool-augmented
 306 RS agentic systems.

3.4. Source RS Datasets

307 To construct the ThinkGeo benchmark, we leverage a di-
 308 verse set of publicly available RS datasets (Table 2) span-
 309 ning various domains: DOTA [33], NWPU-VHR-10 [6],
 310 UCAS-AOD [39], and iSAID [30] support transportation and
 311 aviation tasks; FloodNet [26] and xBD [10] contribute flood-
 312

Table 3. Evaluation results across models on the ThinkGeo benchmark. The table reports step-by-step (left) and end-to-end evaluation results (right), including tool-type accuracy (P: Perception, O: Operation, L: Logic), Ans. (final answer), and answer accuracy under image grounding (Ans_I). Overall, GPT4 family performs the best.

Model	Step-by-Step Metrics				End-to-End Metrics				
	Inst.	Tool.	Arg.	Summ.	P.	O.	L.	Ans.	Ans_I
GPT-4o	82.33	67.73	34.75	84.00	89.78	74.74	67.84	9.78	20.40
GPT-4-1106	86.49	74.05	36.96	77.76	84.65	70.16	65.58	5.16	14.69
Claude-Sonnet	22.31	27.31	0.00	76.15	66.67	71.54	76.80	8.97	7.57
Qwen1.5-7b-chat	26.92	11.47	3.06	76.72	8.81	58.72	19.80	5.43	4.59
Qwen2.5-7b-Instruct	64.88	51.04	20.08	76.40	29.29	34.00	35.98	7.34	6.40
InternLM3-8b-Instruct	50.53	45.37	21.38	48.33	38.43	36.23	30.72	8.15	9.15
LLaMA3-1-8b-Instruct	47.27	37.29	13.69	70.14	58.84	46.67	56.70	3.80	3.42
Phi-3-mini-4k-Instruct	38.28	31.49	13.30	64.76	29.32	34.44	23.97	6.25	5.28
Mistral-7B-Instruct-v0.2	21.35	20.40	0.00	71.84	33.44	44.25	38.54	3.80	3.21
Yi-1.5-6B-Chat	22.29	23.27	0.20	46.25	7.03	2.27	8.04	3.26	2.75
Qwen3-8B (w/reasoning)	26.03	14.73	3.19	65.70	57.99	61.27	34.09	5.16	5.63

313 specific and temporal disaster imagery; AID [32] covers ur- 345
 314 ban/industrial scenes; and the Global Dumpsite Dataset [28] 346
 315 addresses environmental monitoring. To expand transport 347
 316 and object-specific coverage, we additionally incorporate 348
 317 SSDD [38] for maritime ship monitoring, SADD [37] for 349
 318 aviation/aircraft monitoring, and SIVED [19] for ground 350
 319 vehicle detection. Original images are reused, with task- 351
 320 specific annotations added where labels are missing. 352

321 4. Tool Suite and Evaluation 353

322 **Task Format:** Each task is posed as a step and tool-implicit 354
 323 query, requiring the agent to reason and respond in a ReAct- 355
 324 style format [36]. Agents autonomously generate thought 356
 325 steps, select tools from a predefined set, format arguments, 357
 326 and produce final answers, evaluating spatial reasoning, plan- 358
 327 ning, and multi-step execution grounded in RS imagery. 359

328 **Tool Categories:** ThinkGeo extends the AgentLego 360
 329 framework [3] with two additional tools: 361
 330 ChangeDetection [13] for multi-temporal remote 362
 331 sensing analysis and SegmentObjectPixels [16, 17] 363
 332 for segmentation and pixels counting. The toolset is 364
 333 organized into three functional categories: *Perception* 365
 334 (e.g., TextToBbox, ChangeDetection), *Logic*, and 366
 335 *Operation*, supporting object localization, spatial reasoning, 367
 336 and interactive annotation. Logic tools (e.g., Calculator, 368
 337 Solver) support numerical reasoning, distance calcula- 369
 338 tions, and spatial comparisons. Operation tools (e.g., 370
 339 DrawBox, GoogleSearch) facilitate visual annotation 371
 340 and output formatting. This categorization supports 372
 341 fine-grained evaluation (e.g., tool-category performance) 373
 342 and structured analysis of planning behavior across spatial, 374
 343 logical, and domain-specific subtasks. 375

344 **Evaluation Methodology:** We adopt the evaluation frame- 376

work of GTA [29] for step-by-step metrics, including 345
 instruction-following (InstAcc), tool selection (ToolAcc), 346
 argument correctness (ArgAcc), and summary generation 347
 (SummAcc), to assess the agent. While GTA computes an- 348
 swer accuracy (Ans.) using deterministic string matching, 349
 this can misclassify predictions due to variations in phrases. 350
 To mitigate this, we introduce LLM-as-a-judge: curated eval- 351
 uation questions per query and use 4o-mini to verify the 352
 correctness of the prediction. This offers a more reliable 353
 measure of task success, especially for multi-fact answers. 354

355 5. Experiments & Discussion 356

To assess the reasoning and tool-use capabilities of language 356
 models under real-world remote sensing scenarios, we con- 357
 duct comprehensive evaluations on the ThinkGeo bench- 358
 mark. Our benchmark poses multimodal and tool-implicit 359
 challenges that require agentic models to invoke tools across 360
 perception, operation, and logic categories. Unlike prior 361
 evaluations that rely on synthetic queries or shallow tool 362
 interactions, our benchmark emphasizes realistic queries 363
 grounded in satellite or aerial imagery and demands multi- 364
 step reasoning with spatial and numerical precision. 365

Quantitative Analysis: We evaluate a wide range of models, 366
 including GPT-4o [12], GPT-4-1106 [2], and several open- 367
 source variants (e.g., Qwen [4, 25], InternLM [5], LLaMA3 368
 [8], Phi [1], and Mistral [14]), in both step-by-step and 369
 end-to-end settings. The step-by-step mode evaluates in- 370
 termediate stages such as instruction following (Inst.), tool 371
 selection (Tool.), argument formatting (Arg.), and summary 372
 generation (Summ.). The end-to-end mode measures perfor- 373
 mance on tool categories (P: Perception, O: Operation, L: 374
 Logic), final answer correctness (Ans.), and answer correct- 375
 ness under visual grounding (Ans_I). As reported in Table 3, 376

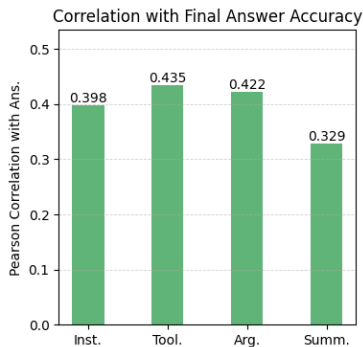


Figure 4. Plot of Pearson correlation between step-by-step execution metrics and final answer accuracy on ThinkGeo.

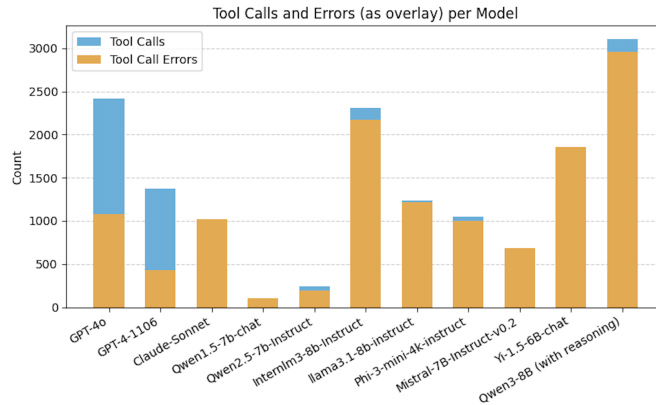


Figure 5. The plot of total number of tool calls made by each model and the corresponding number of tool call errors. The large discrepancy in open source models indicates a high rate of tool misuse. In contrast, models like GPT-4o show better tool invocation reliability.

Table 4. Evaluation results across models on the ThinkGeo SAR benchmark. The table reports step-by-step execution metrics (left) and end-to-end evaluation results (right), including tool-type accuracy (P: Perception, O: Operation, L: Logic), Ans. (final answer), and answer accuracy under image grounding (Ans.I).

Model	Step-by-Step Metrics				End-to-End Metrics				
	Inst.	Tool.	Arg.	Summ.	P.	O.	L.	Ans.	Ans.I
GPT-4o	88.21	77.87	55.33	84.84	82.16	81.82	49.69	5.56	16.83
GPT-4-1106	88.57	75.82	57.38	83.03	59.31	70.00	28.76	2.78	15.11
Claude-Sonnet	19.86	14.75	0.00	87.55	75.41	55.56	80.19	5.56	15.32
Qwen1.5-7b-chat	20.36	4.92	2.87	75.15	4.21	47.06	14.08	2.78	2.00
Qwen2.5-7b-Instruct	65.71	50.82	31.56	84.87	19.61	37.50	24.49	2.78	2.00
InternLM3-8b-Instruct	56.07	46.72	33.61	44.02	40.68	32.00	27.88	2.78	9.22
LLaMA3-1-8b-Instruct	41.79	31.97	20.08	72.36	44.44	28.57	32.89	2.78	2.00
Phi-3-mini-4k-Instruct	38.21	28.69	18.03	65.90	24.19	34.48	11.35	2.78	4.00

377 GPT-4o and GPT-4-1106 achieve the strongest overall accuracy, reflecting superior planning and execution across
 378 multi-step tool chains. Most open-source models struggle
 379 with tool-call formatting and argument-prediction, resulting
 380 in significantly lower answer accuracy. To demonstrate the
 381 extension of our framework to additional modalities, we
 382 present SAR-based analysis in the Table 4. Among all step-
 383 by-step metrics, tool selection has the highest correlation
 384 with final answer accuracy, underscoring its importance in
 385 agentic performance (Fig. 4).
 386

387 **Tool Call & Error:** Tool calls and error rates highlight key
 388 gaps in agentic reliability. Proprietary models (GPT-4o, GPT-
 389 4-1106) show frequent tool use with relatively low error rates
 390 (44.46% and 30.86%), indicating strong tool-handling capa-
 391 bilities (Fig. 5). In contrast, open-source models (Qwen3-
 392 8B, InternLM3-8B, LLaMA3-8B) invoke tools aggressively
 393 but incur high error rates, reflecting poor execution control.
 394 Meanwhile, smaller models (Qwen1.5-7B, Phi-3) often fail

despite limited tool use, underscoring issues with format-
 395 ting and context alignment. Thus, an effective agent hinges
 396 not just on tool access but on precise invocation and robust
 397 reasoning.
 398

Failure Analysis: The qualitative examples in Fig. 6 il-
 399 lustrate several recurring failure cases in multimodal agen-
 400 tic reasoning. GPT-4o, despite its high tool usage, often
 401 struggles with incorrect argument formatting, poorly aligned
 402 bounding boxes, and redundant or unnecessary tool calls,
 403 which together lead to inaccurate counting. Qwen3, on the
 404 other hand, tends to invoke unnecessary tools (e.g., NoAc-
 405 tion), exhibits fragmented or poorly connected reasoning
 406 steps, and fails to adequately justify its numerical outputs
 407 with spatial context. These observations highlight critical
 408 challenges in agent planning and coordination, including ar-
 409 gument misalignment, repeated tool misuse, and the lack of
 410 unit-aware spatially grounded calculations, underscoring the
 411 need for precise reasoning across perception and reasoning
 412

Message: You are an assistant, provided with an image <file> and a text query <query>, who can utilize external tools. {tool_description} To use a tool, please use the following format: {thought} Think what you need to solve, do you need to use tools? {action} the tool name, should be one of [{action_names}] {action_input} the input to the action. The response after utilizing tools should use the following format: {response} the results after call the tool. If you already know the answer, or you do not need to use tools, please use the following format to reply: {thought} the thought process to get the final answer {finish} final answer.

 GPT 4o

Query: Inside the area of all flooded houses, how many cars are parked?

Error Response:

Step 1:
{"arguments": ("flooded houses", top1: true),
"tool": "TextToBbox",
"tool_output": "(821, 446, 1308, 924), score 63"}
Step 2:
{"arguments": {"text": "car", "bbox": "(821, 446, 1308, 924)"},
"tool": "CountGivenObject", "tool_output": "ARGS_ERROR"}
Step 3:
{"arguments": ("car", "bbox": "821, 446, 1308, 924"), "tool": "CountGivenObject",
"tool_output": "2"}
Step 4:
{"arguments": ("car", "bbox": "821, 446, 1308, 924"), "tool": "CountGivenObject",
"tool_output": "2"}
Final answer:
There are 2 cars parked inside the flooded houses.

Issues:

- Incorrect argument to TextToBbox
- Incorrect bounding box returned
- Calling the tool CountGivenObject counted twice
- The number of cars is incorrect

The correct Response:

Step 1:
{"arguments": ("flooded house", top1: false),
"tool": "TextToBbox",
"tool_output": "(129, 654, 689, 1343), score 90"}
Step 2:
{"arguments": ("expression": "min(129, 818), min(288, 654), max(689, 1550), max(1343, 919)",
"tool": "Calculator",
"tool_output": "129, 288, 1550, 1343")
Step 3:
{"arguments": ("car", "bbox": "129, 288, 1550, 1343"), "tool": "CountGivenObject",
"tool_output": "6"}
Final answer:
There are 6 cars parked inside the area of the flooded houses.

 Qwen3

Query: How much area is majorly damaged, while counting the buildings in the image, GSD 2.1m/px?

Error Response:

Step 1:
{"arguments": ("",), "tool": "NoAction",
"tool_output": "API_ERROR: Please follow the format"}
Step 2:
{"arguments": ("majorly damaged buildings", top1: false), "tool": "TextToBbox",
"tool_output": "(3, 21, 1021, 1024), score 60"}
Step 3:
{"arguments": ("expression" = "(1021-3) * (1024-21) * (2.1**2)", "tool": "Calculator",
"tool_output": "4502848.14")

Final answer:

The area that is majorly damaged in the image is approximately 4,502,848.14 square meters, based on the GSD of 2.1 meters per pixel.

Issues:

- Redundant Step 1 with unnecessary NoAction tool
- Empty or invalid arguments in Step 1
- Single low-confidence box used for area
- Final answer lacks justification for area estimate

The correct Response:

Step 1:
{"arguments": ("major-damage building", top1: false), "tool": "TextToBbox",
"tool_output": "(917, 288, 951, 322), score 91"}
Step 2:
{"arguments": ("expression" = "(951-917)*(322-288) + (775-737)...",
"tool": "Calculator",
"tool_output": "27685.98")

Final answer:

There are 10 major-damage buildings, with a total damaged area of 27,685.98 square meters.

Figure 6. Examples of typical failure cases in GPT-4o and Qwen3 during evaluation on ThinkGeo benchmark queries. On the left, GPT-4o struggles with incorrect argument formatting, misidentifies bounding boxes, redundantly invokes tools, and produces an incorrect final count. On the right, Qwen3 misuses tools (e.g., invoking NoAction), introduces redundant reasoning steps, and fails to provide spatial justification in its area estimate. In contrast, the correct responses illustrate structured reasoning with accurate spatial computation and coherent tool invocation.

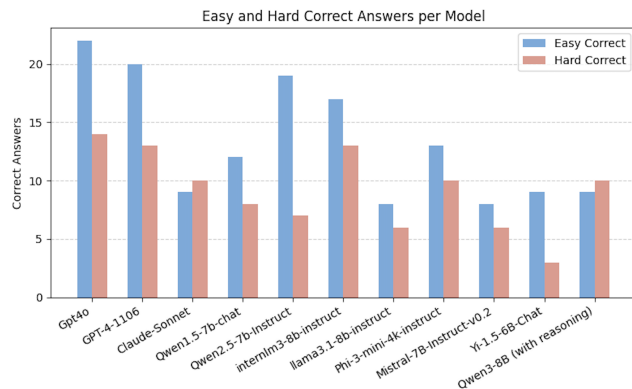


Figure 7. Number of correctly answered queries by each model across varying levels of difficulty. It highlights how LLMs increasingly struggle with complex, multi-step reasoning tasks compared to simpler ones.

modules, especially for complex remote sensing tasks that require both geometric precision and logical consistency.

Easy vs Hard Queries: We analyze the performance of LLM agents on queries of varying difficulty levels, as defined in Sec. 3.3. Fig. 7 presents a bar chart illustrating the number of correct responses for each model, separated by difficulty level. The x-axis lists the evaluated models, while

the y-axis indicates the total count of correctly answered queries. Blue bars represent easy queries, and red bars correspond to hard queries. This analysis highlights a consistent performance gap across difficulty levels, emphasizing the increased challenges LLMs face when dealing with complex, multi-step reasoning tasks.

6. Conclusion

We propose ThinkGeo, the first benchmark tailored specifically to evaluate tool-augmented LLM agents on real-world remote sensing tasks. By grounding evaluation in high-resolution EO imagery, structured tool-use pipelines, and fine-grained reasoning annotations, ThinkGeo reveals key gaps in current agent capabilities. In particular, our analysis shows room for improvement in spatial planning, temporal consistency, and domain-specific tool integration. Our extensive study across 486 tasks and multiple SoTA LLMs demonstrates that, while tools like segmentation and change-detection improve raw perception, true geospatial reasoning remains an open challenge. ThinkGeo aims to attract further efforts towards the development of next-generation multi-modal agents that can seamlessly blend perception, planning, and execution in complex, spatially grounded RS and EO environments.

443

References

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

- [1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 6
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6
- [3] AgentLego Contributors. Agentlego: Open-source tool api library to extend and enhance llm agents. <https://github.com/InternLM/agentlego>, 2023. Apache-2.0 License. Accessed: 2024-05-16. 6
- [4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 6
- [5] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 6
- [6] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016. 5
- [7] Siqi Du, Shengjun Tang, Weixi Wang, Xiaoming Li, and Renzhong Guo. Tree-gpt: Modular large language model expert system for forest remote sensing image understanding and interactive analysis. *arXiv preprint arXiv:2310.04698*, 2023. 3
- [8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6
- [9] Haonan Guo, Xin Su, Chen Wu, Bo Du, Liangpei Zhang, and Deren Li. Remote sensing chatgpt: Solving remote sensing tasks with chatgpt and visual models, 2024. 3
- [10] Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019. 5
- [11] Yangyu Huang, Tianyi Gao, Haoran Xu, Qihao Zhao, Yang Song, Zhipeng Gui, Tengchao Lv, Hao Chen, Lei Cui, Scarlett Li, and Furu Wei. PEACE: Empowering Geologic Map Holistic Understanding with MLLMs. *arXiv e-prints*, art. arXiv:2501.06184, 2025. 3
- [12] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6
- [13] Jeremy Andrew Irvin, Emily Ruoyu Liu, Joyce Chuyi Chen, Ines Dormoy, Jinyoung Kim, Samar Khanna, Zhuo Zheng, and Stefano Ermon. TeoChat: A large vision-language assistant for temporal earth observation data. *arXiv preprint arXiv:2410.06234*, 2024. 6
- [14] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L el io Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth ee Lacroix, and William El Sayed. Mistral 7B. *arXiv e-prints*, art. arXiv:2310.06825, 2023. 6
- [15] Chia Hsiang Kao, Wenting Zhao, Utkarsh Mall, Carl Vondrick, and Bharath Hariharan. Towards llm agents for earth observation. *arXiv preprint arXiv:2504.12099*, 2025. 1, 3
- [16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 6
- [17] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10975, 2022. 6
- [18] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3116, 2023. 1, 3
- [19] Xin Lin, Bo Zhang, Fan Wu, Chao Wang, Yali Yang, and Huiqin Chen. SIVED: A SAR image dataset for vehicle detection based on rotatable bounding box. *Remote Sensing*, 15(11):2825, 2023. 5, 6
- [20] Zixian Ma, Weikai Huang, Jieyu Zhang, Tanmay Gupta, and Ranjay Krishna. m & m’s: A benchmark to evaluate tool-use for m multi-step m multi-modal tasks. In *European Conference on Computer Vision*, pages 18–34. Springer, 2024. 1, 3
- [21] Gr egoire Mialon, Cl ementine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general ai assistants. In *ICLR*, 2024. 1, 3
- [22] Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A New Framework and Benchmark for Advancing AI Research Agents. *arXiv e-prints*, art. arXiv:2502.14499, 2025. 3
- [23] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis, 2023. *arXiv preprint arXiv:2305.15334*, 2023. 1, 3
- [24] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023. 1, 3

- 556 [25] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan
557 Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu,
558 Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu,
559 Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Jun-
560 yang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le
561 Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men,
562 Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang
563 Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,
564 Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zi-
565 han Qiu. Qwen2.5 Technical Report. *arXiv e-prints*, art.
566 arXiv:2412.15115, 2024. 6
- 567 [26] Maryam Rahnmoomfar, Tashnim Chowdhury, Argho Sarkar,
568 Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy.
569 Floodnet: A high resolution aerial imagery dataset for post
570 flood scene understanding. *IEEE Access*, 9:89644–89654,
571 2021. 5
- 572 [27] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weim-
573 ing Lu, and Yueting Zhuang. HuggingGPT: Solving AI Tasks
574 with ChatGPT and its Friends in Hugging Face. *arXiv e-prints*,
575 art. arXiv:2303.17580, 2023. 1, 3
- 576 [28] Xian Sun, Dongshuo Yin, Fei Qin, Hongfeng Yu, Wanxuan
577 Lu, Fanglong Yao, Qibin He, Xingliang Huang, Zhiyuan Yan,
578 Peijin Wang, Chubo Deng, Nayu Liu, Yiran Yang, Wei Liang,
579 Ruiping Wang, Cheng Wang, Naoto Yokoya, Ronny Hänsch,
580 and Kun Fu. Revealing influencing factors on global waste
581 distribution via deep-learning based dumpsite detection from
582 satellite imagery. *Nature Communications*, 14:1444, 2023. 5,
583 6
- 584 [29] Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian
585 Chen, Kai Chen, and Xinyi Le. GTA: A benchmark for
586 general tool agents. In *NeurIPS Datasets and Benchmarks*
587 *Track*, 2024. 1, 3, 5, 6
- 588 [30] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman
589 Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao,
590 Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset
591 for instance segmentation in aerial images. In *Proceedings*
592 *of the IEEE/CVF conference on computer vision and pattern*
593 *recognition workshops*, pages 28–37, 2019. 5
- 594 [31] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang,
595 Zecheng Tang, and Nan Duan. Visual ChatGPT: Talking,
596 Drawing and Editing with Visual Foundation Models. *arXiv*
597 *e-prints*, art. arXiv:2303.04671, 2023. 3
- 598 [32] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang
599 Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. AID:
600 A Benchmark Data Set for Performance Evaluation of Aerial
601 Scene Classification. *IEEE Transactions on Geoscience and*
602 *Remote Sensing*, 55(7):3965–3981, 2017. 5, 6
- 603 [33] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Be-
604 longie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liang-
605 pei Zhang. Dota: A large-scale dataset for object detection in
606 aerial images. In *The IEEE Conference on Computer Vision*
607 *and Pattern Recognition (CVPR)*, 2018. 5
- 608 [34] Wenjia Xu, Zijian Yu, Yixu Wang, Jiuniu Wang, and Mugen
609 Peng. Rs-agent: Automating remote sensing tasks through
610 intelligent agents. *arXiv preprint arXiv:2406.07089*, 2024. 3
- 611 [35] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan
612 Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael
Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT
for Multimodal Reasoning and Action. *arXiv e-prints*, art.
arXiv:2303.11381, 2023. 3
- [36] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran,
Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing
Reasoning and Acting in Language Models. *arXiv e-prints*,
art. arXiv:2210.03629, 2022. 1, 3, 4, 6
- [37] Peng Zhang, Hao Xu, Tian Tian, Peng Gao, Linfeng Li, Tian-
ming Zhao, Nan Zhang, and Jinwen Tian. SEFEPNet: Scale
expansion and feature enhancement pyramid network for SAR
aircraft detection with small sample dataset. *IEEE Journal of*
Selected Topics in Applied Earth Observations and Remote
Sensing, 15:3365–3375, 2022. 5, 6
- [38] Tianwen Zhang, Xiaoling Zhang, Jia Li, Xiang Xu, Bo Wang,
Xiang Zhan, Yahui Xu, Xiaozhi Ke, Tao Zeng, Hao Su, Im-
ran Ahmad, Dong Pan, Chao Liu, Yanan Zhou, Jie Shi, and
Shaoyi Wei. Sar ship detection dataset (ssdd): Official release
and comprehensive data analysis. *Remote Sensing*, 13(18):
3690, 2021. 5, 6
- [39] Haigang Zhu, Xiaogang Chen, Weiqun Dai, Kun Fu, Qixiang
Ye, and Jianbin Jiao. Orientation robust object detection in
aerial images using deep convolutional neural network. In
2015 IEEE international conference on image processing
(ICIP), pages 3735–3739. IEEE, 2015. 5