remote sensing

MDPI

Article

A Cross-Domain Tool-Augmented Vision–Language Framework for Remote Sensing Image Understanding

Xuan Zhou, Xuefeng Wei, Zhi Qu, Yusuke Sakai, Hidetaka Kamigaito and Taro Watanabe *

Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan; xuan.zhou.xc7@naist.ac.jp (X.Z.); xuefeng.wei.yb1@naist.ac.jp (X.W.); qu.zhi.pv5@is.naist.jp (Z.Q.); sakai.yusuke.sr9@is.naist.jp (Y.S.); kamigaito.h@is.naist.jp (H.K.)
* Correspondence: taro@is.naist.jp

Highlights

What are the main findings?

- We propose GeoPilot, a tool-augmented multimodal assistant for remote sensing that supports both optical and SAR imagery.
- GeoPilot shows strong tool-planning ability and competitive performance on representative remote sensing understanding tasks.

What is the implication of the main finding?

- Tool-augmented multimodal assistants are a promising direction for building more practical remote sensing systems.
- The proposed cross-domain dataset and benchmark provide reusable resources for future research on tool-aware remote sensing assistants.

Abstract

Vision–language models (VLMs) hold considerable potential for interpreting large-scale remote sensing (RS) archives, which are critical for applications such as environmental monitoring, disaster response, and urban planning. However, general-purpose VLMs primarily target optical imagery and often underperform on RS tasks, while existing RS-specific VLMs still struggle with fine-grained understanding. To address these limitations, we propose GeoPilot, a tool-augmented multimodal assistant tailored for RS scenarios. GeoPilot interprets user instructions, autonomously determines whether to invoke external tools, and synthesizes their outputs to generate precise responses. A key capability of our approach is its ability to process both optical and Synthetic Aperture Radar (SAR) imagery, supporting representative tasks such as visual grounding, object detection, segmentation, and cross-domain reasoning. To support this setting, we construct a novel large-scale RS instruction dataset that jointly supports optical and SAR imagery together with explicit tool use reasoning traces, addressing the critical challenge of task-specific data scarcity. We also introduce GeoPilotBench, a benchmark for cross-domain, multi-task dialogue and tool-aware evaluation in RS, and use it to assess GeoPilot across representative tasks. Experimental results show that GeoPilot achieves strong task planning accuracy (92.6% overall planning accuracy) and competitive performance on VQA, SAR understanding, and referring object detection. End-to-end evaluation further confirms that GeoPilot's learned tool policy introduces only limited overhead compared to standalone tool execution, demonstrating its practical value as a tool-augmented RS assistant.

Check for updates

Academic Editors: Yang Zhan, Xiuyuan Zhang, Ailong Ma, Qiqi Zhu and Xiaoyan Lu

Received: 10 April 2026
Revised: 11 May 2026
Accepted: 12 May 2026
Published: 17 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: remote sensing; vision–language model; multimodal assistant; tool augmentation; synthetic aperture radar

1. Introduction

Recent advancements in vision–language models (VLMs) have demonstrated notable success in the natural-image domain. These models enable unified visual understanding, capable of performing diverse tasks such as classification, localization, visual question answering, and dense captioning [1–4]. Their strong conversational and instruction-following capabilities have paved the way for general-purpose multimodal assistants [5–9]. This success has motivated efforts to bring such capabilities into the remote sensing (RS) domain. However, creating a truly practical, general-purpose RS assistant requires overcoming two critical frontiers that remain only partially addressed by current systems.

First, resilient Earth observation requires all-weather and day-and-night analytical capabilities, a setting in which optical-only models are fundamentally limited. Although most existing RS vision–language research focuses on optical imagery, Synthetic Aperture Radar (SAR) provides complementary sensing under cloud cover, poor illumination, and adverse weather. These properties are especially important for disaster assessment, maritime surveillance, flood mapping, and emergency response, where optical imagery may be unavailable or unreliable. A robust RS assistant should therefore support both optical and SAR imagery rather than treating SAR as a peripheral modality.

Second, real-world RS applications often require the execution of specialized analytical workflows rather than single-step question answering. A practical assistant should not only reason over image–text inputs, but also decide when specialized tools are required, invoke them in a structured way, and synthesize their outputs into grounded responses. Such a system must remain extensible as new tools, tasks, and sensing modalities emerge. In other words, remote sensing assistants should evolve from monolithic answer generators into modular tool-using agents.

Despite recent progress, existing RS-VLMs still fall short of these goals. Efforts such as SkyEyeGPT [10] have focused on large-scale instruction tuning, while systems such as GeoChat [11] have improved region-level interaction. However, most prior methods remain centered on optical data, and more importantly, they typically lack a unified framework for dynamic tool orchestration. Existing tool-augmented RS systems often rely on fixed toolchains or manually designed templates, which limits flexibility when faced with novel user requests, cross-domain inputs, or more complex reasoning pipelines.

To overcome these challenges, we introduce GeoPilot, a tool-augmented multimodal assistant for RS. As illustrated in Figure 1, GeoPilot is designed to support diverse tasks on both SAR and optical imagery, which we group into two categories: (i) *General Abilities*, such as VQA, classification, and captioning, which rely primarily on the model's internal multimodal reasoning; and (ii) *Tool Abilities*, such as object detection, grounding, segmentation, and restoration, which are supported through external specialist tools. Rather than functioning as a single monolithic model, GeoPilot is designed as a modular end-to-end framework that integrates perception, instruction parsing, task decomposition, tool scheduling, and response generation, while remaining extensible to future tools and capabilities.

To support this system, we construct a novel large-scale RS instruction dataset that jointly supports both optical and SAR imagery together with explicit tool use reasoning traces, comprising 555,913 samples. We further introduce GeoPilotBench, a benchmark suite for evaluating multi-task, cross-domain dialogue and tool-augmented reasoning in RS.

Experiments on GeoPilotBench show that GeoPilot achieves strong task planning accuracy, with 92.6% overall planning accuracy (OPA), and competitive performance on representative downstream tasks, including VQA, referring object detection, SAR captioning, and tool-augmented reasoning.

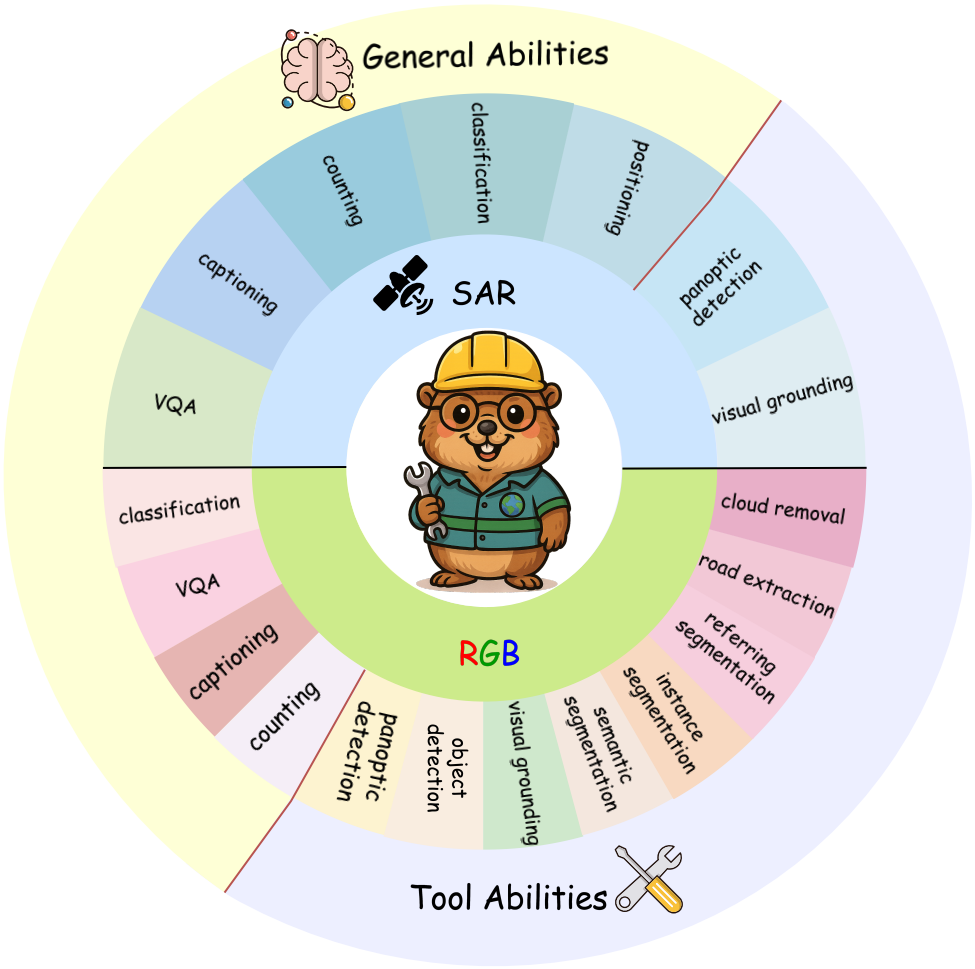

Figure 1. The diverse capabilities of GeoPilot across both SAR and optical imagery.

In summary, our main contributions are as follows:

- We construct a novel large-scale RS instruction dataset that jointly supports optical and SAR imagery together with explicit tool use reasoning traces. It contains 555,913 instruction pairs and enhances the model's cross-domain understanding and tool-application capabilities.
- We develop GeoPilot, a unified, tool-augmented assistant that dynamically orchestrates specialized visual tools based on user instructions. It supports end-to-end execution for representative tasks on both optical and SAR imagery while maintaining a modular and extensible workflow design.
- We introduce GeoPilotBench, a benchmark suite covering both optical and SAR domains, and use it to evaluate GeoPilot. Experimental results show that GeoPilot achieves strong task planning accuracy and competitive downstream performance. End-to-end evaluation on tool-augmented tasks confirms that GeoPilot's learned orchestration policy introduces limited overhead compared to direct tool invocation.

2. Materials and Methods

2.1. Related Work and Positioning

2.1.1. General Vision–Language Models and Multimodal Assistants

General-purpose VLMs have rapidly evolved from contrastive representation learning to instruction-following multimodal assistants. Early milestones such as CLIP [12] established strong image–text alignment, while BLIP and BLIP-2 [13,14] advanced multimodal understanding and generation through bootstrapped pretraining and efficient language model adaptation. Instruction-tuned models such as LLaVA [1], MiniGPT-v2 [6], InstructBLIP [15], and InternVL [8] further strengthened multimodal dialogue, reasoning, and task generalization. More recently, frontier VLMs such as GPT-4V-style systems [5], Qwen2.5-VL [9], and stronger open-source suites [3] have demonstrated increasingly broad visual reasoning abilities.

However, most of these systems are developed and evaluated primarily on natural-image benchmarks. Their assumptions do not directly transfer to RS imagery, where objects can be extremely small, spatial relations are often critical, semantic content may be sparse, and domain-specific sensing modalities, such as SAR, exhibit substantially different visual statistics. These domain gaps motivate the development of RS-specific multimodal models rather than direct reuse of natural-image VLMs.

2.1.2. Tool-Augmented Language and Vision–Language Agents

The broader line of work on intelligent agents has long aimed to integrate perception, reasoning, and action [16]. Earlier paradigms include symbolic planning agents [17] and reactive agents [18], while later machine learning approaches, especially deep reinforcement learning, enabled more adaptive and data-driven decision processes [19–21]. With the emergence of large language models, the focus shifted toward instruction-following agents that can decompose tasks, call external tools, and interact with environments [22].

Several influential frameworks have defined the modern tool-augmented agent paradigm. ReAct [23] interleaves reasoning and acting, Toolformer [24] teaches language models to invoke APIs, HuggingGPT [25] organizes expert models under LLM control, and AutoGPT-style systems explore autonomous multi-step execution. In the multimodal setting, LLaVA-Plus [2] demonstrates that image-grounded tool use can be learned through unified instruction data. These studies provide the conceptual foundation for our work. Nevertheless, they are still largely grounded in natural-image scenarios and generic APIs rather than geospatial reasoning, RS-specific tools, and cross-modal Earth observation inputs.

2.1.3. Remote Sensing Vision–Language Models

A growing body of work has sought to adapt multimodal models to RS imagery. Early RS-VLM systems such as RSGPT [26] and RS-ChatGPT [27] established the feasibility of instruction-following dialogue for aerial and satellite images. GeoChat [11] further advanced grounded remote sensing dialogue, especially for referring and region-level tasks. SkyEyeGPT [10] emphasized the scale of instruction data and the unification of multi-task learning. Other recent methods, such as LHRS-Bot [28], EarthGPT [29], EarthMarker [30], LRSCLIP [31], RS-MoE [32], and Falcon [33], explore various forms of alignment, adaptation, and expert routing for RS understanding. More recently, VHM [34] and Co-LLaVA [35] have demonstrated competitive VQA performance through specialized training strategies, while RS-LLaVA [36] showed the effectiveness of LoRA-based adaptation for joint captioning and VQA.

Despite this progress, most current RS-VLMs remain strongest on image-level optical tasks and provide limited support for richer region- or pixel-level interactions. Their

support for SAR is often weak or absent, and they rarely learn explicit tool use policies. As a result, they can answer descriptive questions but are less capable of executing specialized workflows requiring detection, segmentation, grounding, or restoration.

2.1.4. Tool Use and Cross-Domain Learning in Remote Sensing

A smaller but important line of work explores explicit tool integration for RS. RS-Agent [37] is a representative attempt to combine an LLM controller with domain tools. However, its reliance on fixed pipelines and predefined templates constrains flexibility when user requests or task compositions change. More broadly, recent RS research has highlighted the need for cross-domain capability, especially the joint handling of optical and SAR imagery [38,39]. Yet the field still lacks a unified framework that simultaneously offers instruction-following dialogue, dynamic tool orchestration, and cross-domain training over both optical and SAR data.

Our work addresses this gap. GeoPilot combines an instruction-tuned multimodal backbone with explicit, learnable tool invocation and cross-domain data construction. The resulting system is designed not only to answer questions about RS imagery, but also to plan and execute specialized operations in a flexible manner.

To provide a concise overview of the current landscape and position our work, Table 1 compares representative RS VLMs across image-level, region-level, pixel-level, and tool-related capabilities. As the table shows, many existing models handle image-level optical tasks, but few offer a complete suite of region- and pixel-level abilities. Support for SAR data and dedicated training for tool invocation are even rarer. GeoChat mainly focuses on grounded RS dialogue, while EarthGPT and SkyEyeGPT emphasize large-scale RS instruction tuning and general vision–language understanding. RS-Agent is closer to tool-augmented RS analysis, but it uses a different predefined tool repository and task schema. In contrast, GeoPilot jointly combines optical/SAR instruction data, structured tool use reasoning traces, learned tool/no-tool decision making, and unified evaluation of both general understanding and tool-augmented execution.

Table 1. A comprehensive comparison of capabilities for representative RS-oriented vision–language models. CL: classification, IC: image captioning, VQA: visual question answering, OP: object positioning, PD: panoptic detection, OD: object detection, VG: visual grounding, SS: semantic segmentation, IS: instance segmentation, RS: referring segmentation. A checkmark indicates that the capability is supported or explicitly reported by the corresponding model; a blank cell indicates that the capability is unsupported or not explicitly reported.

	Models	Image Level							Region Level					Pixel Level			External Skills	
		CL	IC	VQA	CL_{SAR}	IC_{SAR}	VQA_{SAR}	OP_{SAR}	PD	OD	VG	PD_{SAR}	VG_{SAR}	SS	IS	RS	Tool	Learning
RS VLMs	GeoChat	✓	✓	✓							✓							
	LHRS-Bot	✓	✓	✓						✓	✓							
	RSGPT	✓	✓	✓														
	EarthGPT	✓	✓	✓							✓							
	RS-ChatGPT	✓	✓	✓											✓			
	SkyEyeGPT	✓	✓	✓							✓							
	EarthMarker	✓	✓	✓														
	Falcon	✓	✓	✓						✓	✓				✓			
	RS-Agent	✓	✓	✓					✓			✓			✓		✓	
	GeoPilot (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

2.2. GeoPilot Framework

GeoPilot is designed to address a spectrum of multimodal RS tasks, ranging from holistic image comprehension to fine-grained, spatially aware interactions. Its framework strategically combines intrinsic cross-domain understanding with selective tool augmentation, thereby improving both robustness and precision. Figure 2 presents the overall architecture. The tasks executable by GeoPilot can be grouped into the following two operational modes: ***General Image Understanding and Dialogue*** and ***Tool-Augmented Spatial Reasoning***.

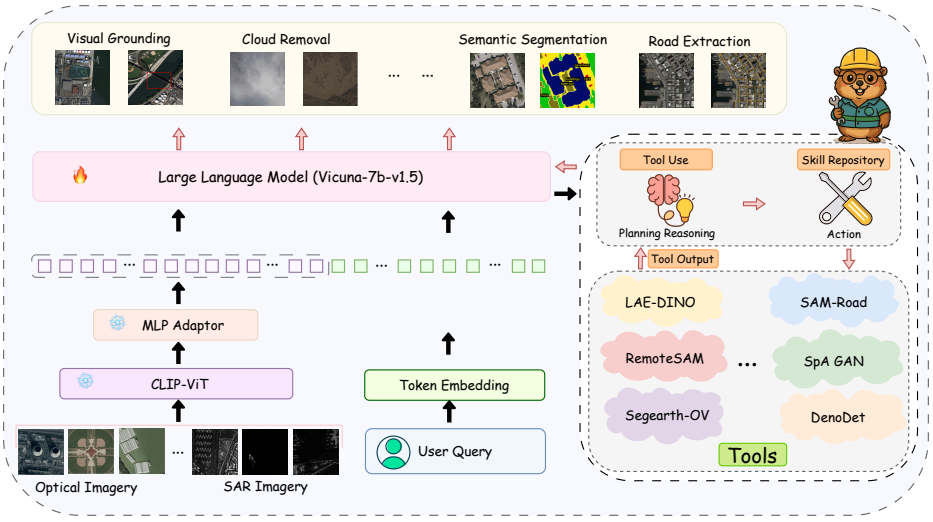

Figure 2. An overview of the GeoPilot architecture. Our model is centered around Vicuna 7B-v1.5, which processes multimodal inputs. The framework features a dual-pathway design: it can directly generate responses for general understanding tasks (top path), or invoke specialized external tools via a planning and reasoning module for precision-based tasks (right path).

General Image Understanding and Dialogue mode is dedicated to holistic, context-driven reasoning over RS imagery and user text. It supports image-level dialogue without requiring explicit spatial coordinates, including VQA, scene classification, and image description. Importantly, it also covers SAR-oriented understanding, enabling SAR-specific VQA and captioning.

Tool-Augmented Spatial Reasoning mode is activated for tasks that require fine-grained localization, grounded interaction, or precise output. These include referring object grounding, instance-level reasoning, semantic segmentation, road extraction, and restoration-related tasks such as cloud removal. In this mode, GeoPilot determines whether a specialist tool is needed, selects the appropriate tool, executes it, and integrates the result into a grounded response.

2.2.1. Architecture

GeoPilot is built upon the LLaVA-Plus framework [2], which provides multimodal reasoning and dynamic tool integration capabilities. The system consists of three core components:

(i) Visual Backbone. We adopt CLIP-ViT(L-14) as the visual encoder, with an input resolution of 336 × 336. Given an input image X_i, the visual encoder produces a sequence of visual features:

$$Z_v = \text{CLIP-ViT}(X_i) \in \mathbb{R}^{N \times D_v} \tag{1}$$

where N is the number of visual tokens and D_v is the visual feature dimension.

(ii) Cross-Modal Adapter. We use a lightweight two-layer Multi-Layer Perceptron (MLP) adapter following LLaVA [1]. The adapter projects visual features into the language model's embedding space:

$$H_v = W_2 \cdot \text{GELU}(W_1 \cdot Z_v + b_1) + b_2 \tag{2}$$

where $H_v \in \mathbb{R}^{N \times D_l}$ and D_l is the hidden dimension of the language model. The adapter is initialized from the LLaVA MLP projector pretrained on CC3M-595K, providing a stable starting point for domain-specific RS adaptation.

(iii) Large Language Model. The core controller is Vicuna-v1.5 (7B) [40]. It receives the concatenated multimodal input:

$$H_{\text{input}} = [H_v;\ H_q] \tag{3}$$

where H_q denotes the token embeddings of the user's textual instruction X_q. The LLM is responsible for instruction interpretation, multimodal reasoning, response generation, and structured tool invocation.

2.2.2. Workflow and Interaction Format

GeoPilot operates through a structured, multi-turn dialogue workflow that unifies direct reasoning and tool-augmented execution. As shown in Figure 3, the interaction is formalized as follows:

$$\text{User} : X_i,\ X_q\texttt{<STOP>}\ \text{Agent} : Y_{\text{think}}\texttt{<STOP>} \tag{4}$$

$$\text{User} : X_{\text{tool}}\texttt{<STOP>}\ \text{Agent} : Y_{\text{ans}}\texttt{<STOP>} \tag{5}$$

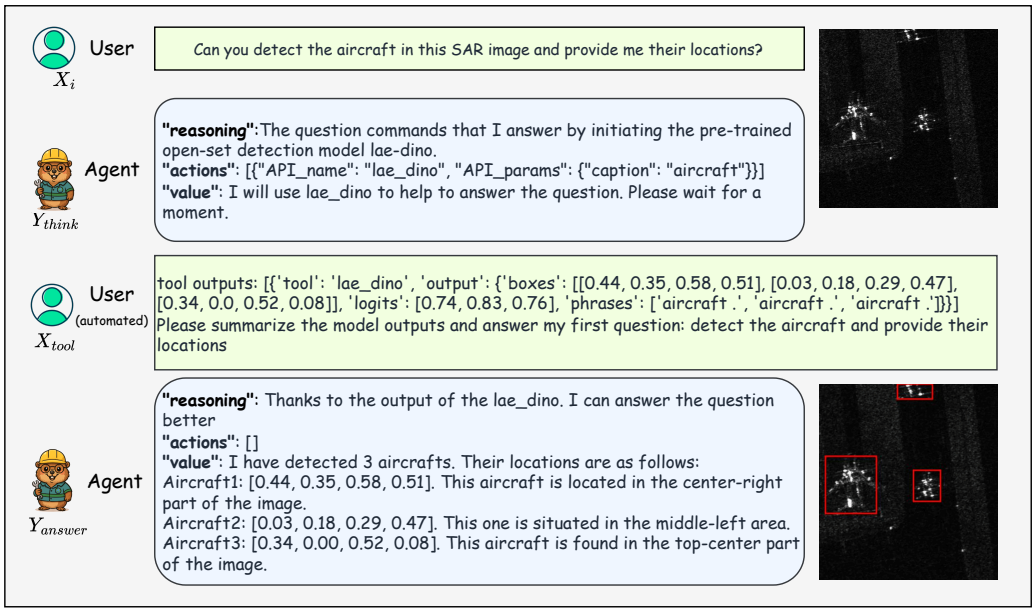

Figure 3. GeoPilot workflow on a tool-augmented object detection task. The process unfolds in two turns. In the first turn, the agent receives the user's image–query pair (X_i, X_q) and generates a plan (Y_{think}) that includes its reasoning and a call to an external tool. In the second turn, the agent receives the tool's output (X_{tool}) and synthesizes this new information to produce the final, grounded answer (Y_{ans}). "User (automated)" denotes an internal system step that reformats the tool output and feeds it back to the agent.

In the first turn, the user provides an image X_i and instruction X_q. The agent produces an intermediate output Y_{think} that contains its reasoning trace and, when necessary, a structured tool call. The corresponding tool is then executed outside the language model, and its output is reformatted as X_{tool}. In the second turn, the agent synthesizes this observation to produce the final answer Y_{ans}.

To support both tool-free and tool-augmented tasks under a unified interface, we represent the agent output using three fields: *reasoning*, *actions*, and *value*. The *actions* field is null when no external tool is required. During training, serialized tool outputs are derived from ground-truth annotations or curated source labels, which provide clean supervision for learning how to synthesize final answers from structured tool evidence. During inference, however, tool outputs are produced by the actual external specialist tools.

Therefore, the final deployment-time response depends on both GeoPilot's tool selection and parameter generation ability, as well as the accuracy of the invoked specialist tool.

2.2.3. Training Objective

The entire model is trained end-to-end using teacher-forced autoregressive decoding with a standard negative log-likelihood objective. The training loss is computed exclusively on the agent-generated tokens (Y_{think} and Y_{ans}) as follows:

$$\mathcal{L} = -\sum_{t \in Y_{\text{think}}} \log P_\theta(y_t \mid y_{<t}, X_i, X_q) - \sum_{t \in Y_{\text{ans}}} \log P_\theta(y_t \mid y_{<t}, X_i, X_q, Y_{\text{think}}, X_{\text{tool}}) \quad (6)$$

where θ denotes the model parameters and y_t is the t-th token. This targeted loss encourages the model to learn not only what to answer, but also when a tool is necessary and how to incorporate its result into the final response.

In practice, for tool-augmented samples, the two dialogue turns are concatenated into a single training sequence:

$$S = [\, X_i,\ X_q,\ Y^*_{\text{think}},\ X^*_{\text{tool}},\ Y^*_{\text{ans}} \,] \quad (7)$$

The loss in Equation (6) is computed via standard teacher-forced autoregressive decoding over this concatenated sequence, with the loss mask applied exclusively to agent-generated tokens (Y^*_{think} and Y^*_{ans}). Since the full sequence is processed autoregressively, Y^*_{ans} is naturally conditioned on all preceding tokens including Y^*_{think} and X^*_{tool}, which realizes the two-term decomposition in Equation (6). For tool-free samples, the actions field is null and no tool output exists, so the sequence reduces to $S = [\, X_i,\ X_q,\ Y^*_{\text{think}} \,]$, where the `value` field of Y^*_{think} directly contains the final answer.

2.2.4. Tool Orchestration Mechanism

Beyond the general workflow, an important feature of GeoPilot is that tool invocation is treated as a learnable decision rather than a fixed external rule. Formally, the tool selection can be expressed as follows:

$$a^* = \arg \max_{a_j \in \mathcal{A} \cup \{\emptyset\}} P_\theta(a_j \mid H_v, H_q) \quad (8)$$

where $\mathcal{A} = \{a_1, \ldots, a_K\}$ is the set of available tools and $\emptyset$ denotes no tool invocation. Each tool is associated with a task type, expected input schema, and output schema.

During inference, GeoPilot distinguishes tool-free and tool-required queries through the generated `actions` field. If `actions` is empty or null, the query is treated as a direct-response case, and the generated `value` field is returned as the final answer. If `actions` contains a valid tool call, the external controller executes the corresponding specialist tool and feeds the formatted output back to the model for final response synthesis. Importantly, we do not use an additional rule-based semantic tool selector. Post-processing is limited to syntactic parsing and schema validation, such as checking whether the generated tool name exists in the registry and whether the parameters conform to the expected input format.

In practice, tool orchestration proceeds through four conceptual stages: (1) *instruction parsing*, which identifies whether the user request requires a specialist tool; (2) *tool selection*, which maps the parsed intent to one of the registered external modules; (3) *tool execution*, handled outside the LLM by the corresponding specialist model; and (4) *result synthesis*, which converts the raw tool output into a user-facing grounded answer. Algorithm 1 summarizes the complete inference workflow.

Algorithm 1 GeoPilot Inference Workflow

Require: Image X_i, User query X_q, Tool repository $\mathcal{A}$
Ensure: Final response Y_{ans}
$Z_v \leftarrow \text{CLIP-ViT}(X_i)$ {Visual encoding}
$H_v \leftarrow \text{MLP}(Z_v)$ {Cross-modal projection}
$H_q \leftarrow \text{TokenEmbed}(X_q)$ {Text tokenization}
$H_{\text{input}} \leftarrow [H_v;\ H_q]$ {Multimodal fusion}
$Y_{\text{think}} \leftarrow \text{LLM}(H_{\text{input}})$ {First turn: planning}
Parse $Y_{\text{think}} \rightarrow \{\textit{reasoning}, \textit{actions}, \textit{value}\}$
if $\textit{actions} \neq \emptyset$ **then**
  for each action a_j in $\textit{actions}$ **do**
    $o_j \leftarrow \text{Execute}(a_j.\text{tool},\ a_j.\text{params},\ X_i)$
  end for
  $X_{\text{tool}} \leftarrow \text{FormatOutput}(o_1, \ldots, o_m)$
  $Y_{\text{ans}} \leftarrow \text{LLM}(H_{\text{input}}, Y_{\text{think}}, X_{\text{tool}})$ {Second turn: synthesis}
else
  $Y_{\text{ans}} \leftarrow \textit{value}$ from Y_{think} {Direct response}
end if
return Y_{ans}

This modular decomposition is important for extensibility: new tools can be added without changing the core interaction format, provided that their call interface and outputs follow the same structured schema.

2.3. RS Tool-Augmented Instruction Dataset

To serve as the foundation for training GeoPilot, we construct a large-scale tool-augmented instruction dataset containing 555,913 samples. The dataset is designed not only to teach the model to answer RS questions, but also to plan and execute tool-dependent workflows across both optical and SAR imagery. The optical portion accounts for 285,146 samples (51.3%) and the SAR portion accounts for 270,767 samples (48.7%), ensuring balanced cross-domain coverage. Table 2 summarizes the abilities covered and the underlying source datasets. The dataset consists of two types of data: *General Understanding Tasks* and *Tool-Augmented Tasks*.

Table 2. Overview of the abilities of GeoPilot and data statistics of our constructed RS tool-augmented instruction dataset. **For optical imagery**: DOTA [41], DIOR [42], FAIR1M [43] are used for panoptic/object detection; RiSBench [44], OpenEarthMap [45] for segmentation; RiSBench and iSAID [46] for instance/referring segmentation; SpaceNet [47] and CityScale [48] for road extraction; RICE [49] for cloud removal. **For SAR imagery**: SARLANG-1M [38] is used for general understanding; SARDet-100K [39] for panoptic detection and visual grounding.

	Abilities	Tools	Source	Size
Optical Imagery Tasks	General Understanding	-	GeoChat Instruct	198,326
	Panoptic Detection	LAE-DINO [50]	DOTA, DIOR, FAIR1M	18,982
	Object Detection	LAE-DINO	DOTA, DIOR, FAIR1M	12,557
	Visual Grounding	RemoteSAM [51]	RiSBench, DIOR	17,086
	Semantic Segmentation	Segearth-OV [52]	OpenEarthMap	6000
	Instance Segmentation	RemoteSAM	RiSBench, iSAID	13,000
	Referring Segmentation	RemoteSAM	RiSBench, iSAID	14,999
	Road Extraction	SAM-Road [48]	SpaceNet, CityScale	1236
	Cloud Removal	SpA-GAN [53]	RICE	2960
SAR Tasks	General Understanding	-	SARLANG-1M	249,488
	Panoptic Detection	DenoDet [54]	SARDet-100K	11,710
	Visual Grounding	Fine-tuned LAE-DINO	SARDet-100K	9569
Total	-	-	-	**555,913**

2.3.1. Data for General Understanding

To cultivate GeoPilot's internal reasoning for tool-free scenarios, we curate a corpus for general visual understanding across both optical and SAR domains. For optical imagery, we refine the GeoChat Instruct dataset [11] by filtering out samples whose core requirement is explicit grounding and then augmenting each retained instruction–response pair with a generated reasoning step. For SAR, we adapt SARLANG-1M [38] by converting image–caption pairs into instruction-following dialogue data with corresponding questions and thought processes. This unified format teaches the model to address descriptive and inferential queries using its internal multimodal knowledge.

2.3.2. Data for Tool-Augmented Tasks

To teach GeoPilot how and when to invoke external tools, we design a unified data generation pipeline and apply it to both optical and SAR datasets. For each source annotation, Gemini 2.5 Flash [55] is prompted to generate: (i) a plausible human-like query X_q; (ii) an intermediate agent response Y_{think} containing reasoning and a structured tool call; (iii) a formatted tool output X_{tool} derived from the ground-truth annotation; and (iv) a final grounded answer Y_{ans}. Importantly, Gemini 2.5 Flash is used as an annotation-grounded language generator rather than as the source of visual evidence. The target categories, coordinates, tool outputs, and grounding evidence are derived from source annotations and tool-specific schemas. This process yields two-turn training samples that explicitly model tool-aware reasoning and response synthesis.

2.3.3. Data Generation Pipeline

The data construction pipeline is one of the central components of our approach. Traditional instruction-tuning datasets typically map an input directly to an answer. By contrast, our tool-augmented formulation explicitly encodes the intermediate decision process that determines whether a tool should be used and how its output should be integrated. This is especially important in RS, where many tasks are not purely descriptive but require specialist perception modules.

Concretely, each sample is generated from a source label through four stages. First, a natural user instruction is produced to mimic realistic human interaction. Second, a planning response is created that specifies both the reasoning and the structured tool call. Third, the original annotation is transformed into a tool output representation consistent with the selected tool interface. Fourth, a final answer is generated that grounds the response in the tool evidence. This structured pipeline allows GeoPilot to learn the full sequence from user request to grounded response, rather than only the final textual output.

Figure 4 illustrates the in-context prompting setup used during data generation. In the first turn, a system prompt guides the LLM to formulate a realistic user query from descriptive ground-truth data. In the second turn, the system prompt provides detailed context about the specific tool, including the task definition, expected output format, coordinate system rules, and self-correction instructions. This structured prompting is essential for generating the high-quality, tool-aware reasoning chains that form the core of our training data.

2.3.4. Automated Data Quality Control

Because our dataset is partially generated through large language model prompting, quality control is essential. We distinguish structural validation from semantic validation.

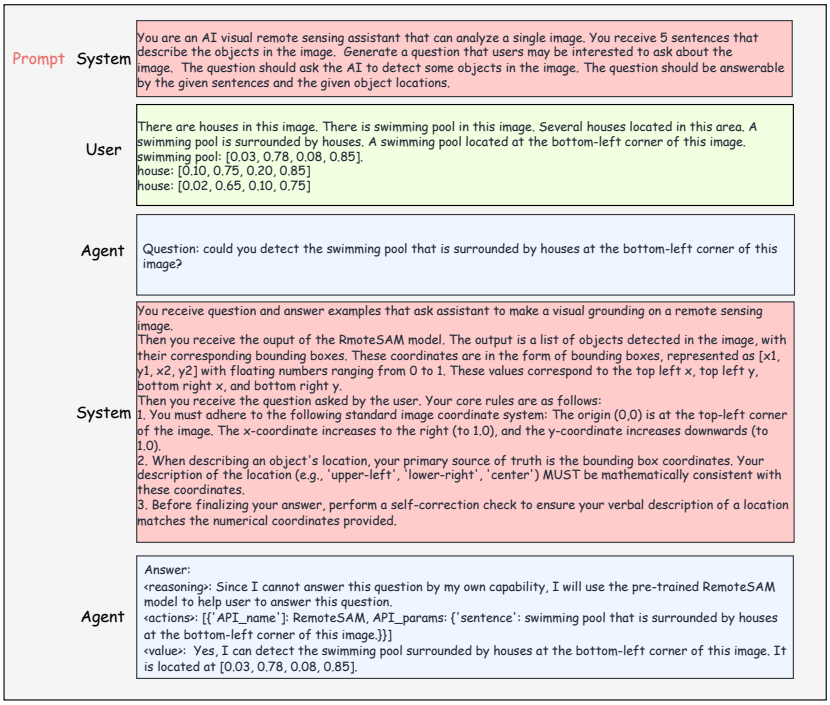

Figure 4. An example of instruction data construction using in-context learning. In the first turn, a system prompt guides the LLM to formulate a realistic user query from descriptive ground-truth data. In the second turn, the system prompt provides detailed context about the specific tool, including the task definition, expected output format, coordinate system rules, and self-correction instructions.

For structural validation, we integrate a suite of automated validators directly into the data generation pipeline. Specifically, we employ: (i) a **coordinate validator** that verifies all bounding box values fall within $[0, 1]$ and satisfy $x_1 < x_2$, $y_1 < y_2$; (ii) a **tool-query consistency checker** that verifies the tool specified in the action field matches the expected tool for each source dataset; and (iii) a **format validator** that ensures all API parameter fields conform to the required JSON schema. Any sample that fails validation is automatically regenerated by re-prompting Gemini 2.5 Flash until all checks are passed. During the construction process, approximately 1.2% of samples required at least one regeneration during structural/schema-level validation, with the majority of failures caused by minor coordinate boundary violations.

Beyond structural validation, our pipeline also performs semantic consistency validation to ensure that generated samples remain grounded in source annotations. We check four semantic dimensions: query answerability, tool call correctness, parameter consistency, and final answer faithfulness. Query answerability verifies whether the generated user query is clear and answerable from the image annotation or the formatted tool output. Tool call correctness checks whether the selected tool is appropriate for the task type. Parameter consistency verifies whether the generated API parameters are consistent with the user query and source annotation. Final answer faithfulness checks whether the final response faithfully summarizes the tool output in terms of object count, category, bounding boxes, masks, or other returned results without introducing hallucinated information. A sample is considered overall semantically valid only if it passes all four criteria.

On first-pass generated samples before final correction, the semantic validation achieved 100.00% query answerability, 100.00% tool call correctness, 97.20% parameter consistency, 95.20% final answer faithfulness, and 93.36% overall semantic validity. Samples failing any criterion were regenerated or corrected and then re-validated before inclusion. We also conducted a manual audit on 200 stratified first-pass samples with two annotators.

The agreement rates were 100.0%, 100.0%, 99.0%, 97.5%, and 97.5% for query answerability, tool call correctness, parameter consistency, final answer faithfulness, and overall semantic validity, respectively. For dimensions where kappa was applicable, Cohen's kappa ranged from 0.49 to 0.69. The final dataset is fully validated, with all 555,913 samples passing the complete quality control suite.

2.4. GeoPilotBench and Experimental Setup

2.4.1. Benchmark: GeoPilotBench

To systematically evaluate GeoPilot, we introduce ***GeoPilotBench***, a benchmark suite spanning both optical and SAR modalities across a range of RS tasks. GeoPilotBench is constructed from established datasets and reformulated into a unified instruction–response format, enabling consistent evaluation of both general understanding and tool-augmented reasoning. Importantly, all benchmark samples are strictly disjoint from the training instruction data, ensuring no evaluation leakage. For datasets that appear in both the training set (Table 2) and the benchmark, we enforce strict split-level separation: SAR VQA evaluation uses the SARDet-100K test split, while training samples are drawn exclusively from the training split. For SAR captioning, we revise the evaluation to use the official SARLANG-1M-Cap test split, which contains 13,682 caption samples. We verified that both the original 6000-sample held-out subset and the official test split have no overlap with the training data at either the image level or the image–caption pair level. Table 3 summarizes the benchmark.

Table 3. Overview of GeoPilotBench, including datasets, size, and input/output formats.

Task	Dataset	Size	Input/Output
Task Planning	Custom	500	Instruction → Tool Calls
VQA	RSVQA-LRBEN	10,004	Image + Question → Answer
	RSVQA-HRBEN	62,554	Image + Question → Answer
Referring Object Detection	GeoChat-Instruct	7593	Image + Referring Expression → Bounding Box
SAR VQA	SARDet-100K	11,955	Image + Question → Answer
SAR Image Captioning	SARLANG-1M-Cap	13,682	Image → Text

GeoPilotBench includes the following task categories:

- **Task Planning Accuracy:** A test set of 500 queries covering all tool categories and a "no-tool" category, stratified by difficulty (easy, medium, hard), measuring whether the agent correctly selects the required tool, provides correct parameters, and appropriately avoids tool use when unnecessary. The planning queries are constructed using a combination of manually designed templates and automatically generated paraphrases to ensure balanced coverage across tool categories and difficulty levels.
- **Visual Question Answering (VQA):** RSVQA-LRBEN and RSVQA-HRBEN [56], covering diverse geographic regions and question types.
- **Referring Object Detection:** GeoChat-Instruct [11], evaluated using Acc@0.5.
- **SAR Understanding:** SARDet-100K [39] for SAR VQA (object identification, counting, classification, positioning), and SARLANG-1M [38] for SAR captioning.

2.4.2. Evaluation Protocols and Metrics

We employ task-appropriate evaluation metrics in accordance with established practice. For VQA-style tasks (optical VQA and SAR VQA), we use accuracy. For task planning, we report four complementary metrics: Tool Selection Accuracy (TSA), which measures whether the correct tool is selected; Parameter Accuracy (PA), which evaluates whether the generated parameters are valid; No-Tool Judgment Accuracy (NTA), which measures

the ability to correctly abstain from tool use when unnecessary; and overall planning accuracy (OPA), which measures the percentage of queries for which the overall planning decision is correct: correct tool selection with valid parameters for tool-required queries, or correct abstention for no-tool queries. For referring object detection, we report Acc@0.5, where a prediction is counted as correct if the IoU between the predicted and ground-truth bounding boxes is at least 0.5. For SAR captioning, we adopt BLEU [57], ROUGE-L [58], and CIDEr [59].

2.4.3. Implementation, Training and Serving

We initialize GeoPilot using pre-trained CLIP-ViT-L/14 and LLaVA-v1.5 (7B) weights. We intentionally adopt a moderate-scale 7B backbone to isolate the contribution of tool use learning from improvements arising purely from model scaling. The GeoPilot framework is architecturally compatible with different multimodal LLM backbones because tool invocation is expressed through structured action schemas; however, all experiments in this paper are conducted with Vicuna-7B, and empirical validation with other backbone families is left for future work. All experiments were implemented using PyTorch 2.0.1+cu117, TorchVision 0.15.2, and CUDA 11.7. The model undergoes full-parameter fine-tuning for two epochs using the AdamW optimizer, a learning rate of 2×10^{-5}, a cosine learning rate scheduler, and a global batch size of 144. All images are processed at a resolution of 336×336. The full training process takes approximately 4 days on 8 NVIDIA RTX 6000 Ada GPUs (NVIDIA Corporation, Santa Clara, CA, USA). Unless otherwise noted, all results are reported using random seed 42. For VQA experiments, we additionally report mean $\pm$ std over three seeds (42, 123, 456) to assess stability.

The fine-tuning curriculum follows a two-stage process, formalized in Algorithm 2.

Algorithm 2 Two-Stage Training of GeoPilot

Require: Optical dataset $\mathcal{D}_{\text{opt}}$, SAR dataset $\mathcal{D}_{\text{SAR}}$, Mixing ratio α, Pre-trained weights θ_0
Ensure: Fine-tuned model θ^*

1: **Stage 1: Optical domain training**
2: $\theta_1 \leftarrow \theta_0$
3: **for** epoch $= 1$ to E **do**
4: **for** each batch (X_i, X_q, Y^*) in $\mathcal{D}_{\text{opt}}$ **do**
5: **if** Y^* contains tool call **then**
6: $S \leftarrow \text{Concat}(X_i, X_q, Y^*_{\text{think}}, X^*_{\text{tool}}, Y^*_{\text{ans}})$
7: $\mathcal{L} \leftarrow -\sum_{t \in Y^*_{\text{think}} \cup Y^*_{\text{ans}}} \log P_{\theta_1}(y_t \mid S_{<t})$
8: **else**
9: $S \leftarrow \text{Concat}(X_i, X_q, Y^*_{\text{think}})$
10: $\mathcal{L} \leftarrow -\sum_{t \in Y^*_{\text{think}}} \log P_{\theta_1}(y_t \mid S_{<t})$
11: **end if**
12: $\theta_1 \leftarrow \theta_1 - \eta \cdot \nabla \mathcal{L}$
13: **end for**
14: **end for**
15: **Stage 2: SAR domain training with optical replay**
16: $\mathcal{D}_{\text{replay}} \leftarrow \text{RandomSample}(\mathcal{D}_{\text{opt}},\ \alpha \cdot |\mathcal{D}_{\text{opt}}|)$
17: $\mathcal{D}_{\text{stage2}} \leftarrow \mathcal{D}_{\text{SAR}} \cup \mathcal{D}_{\text{replay}}$
18: $\theta_2 \leftarrow \theta_1$
19: **for** epoch $= 1$ to E **do**
20: **for** each batch (X_i, X_q, Y^*) in $\text{Shuffle}(\mathcal{D}_{\text{stage2}})$ **do**
21: Compute $\mathcal{L}$ as in Stage 1 (lines 5–11)
22: $\theta_2 \leftarrow \theta_2 - \eta \cdot \nabla \mathcal{L}$
23: **end for**
24: **end for**
25: $\theta^* \leftarrow \theta_2$
26: **return** θ^*

In the first stage, the model is trained on the optical RS dataset. In the second stage, training focuses on SAR data while mixing in optical image–text pairs to mitigate catastrophic forgetting. The mixing ratio $\alpha \in [0, 1]$ controls the proportion of optical samples replayed during stage 2:

$$\mathcal{D}_{\text{stage2}} = \mathcal{D}_{\text{SAR}} \cup \text{Sample}(\mathcal{D}_{\text{opt}},\ \alpha \cdot |\mathcal{D}_{\text{opt}}|) \tag{9}$$

where $\alpha = 0.2$ is selected based on the ablation study in Section 3.7. During training, each tool-augmented task is guided by a task-specific chain-of-thought template aligned with the target tool interface. For deployment and latency measurement, the 7B GeoPilot model is served through FastChat 0.2.36 [40] on a single NVIDIA A100 80GB GPU (NVIDIA Corporation, Santa Clara, CA, USA), and the registered specialist tools are invoked on the same machine as needed. With batch size 1, the average latency for direct-response queries is approximately 1.2 s, while tool-augmented queries require approximately 4.8 s due to additional tool execution and second-turn response synthesis.

2.4.4. Baseline Details

To contextualize the experimental comparison, Table 4 summarizes the backbone, training data scale, and training strategy of representative compared models. We note that the baselines differ in both architecture and data composition. GeoPilot shares the same Vicuna-7B backbone as GeoChat, VHM, and RSGPT, so performance differences are not solely attributable to backbone capacity. With 555,913 SFT samples, GeoPilot uses a moderate-scale instruction dataset—larger than VHM (>151,000 SFT samples) and GeoChat (∼318,000 instruction pairs), but substantially smaller than EarthGPT (>1 million image–text pairs) or InternVL2. Consequently, the modest VQA improvements may be partially attributable to differences in training data scale rather than architectural advantages.

Table 4. Comparison context of representative baselines used in the main experiments. Data sizes are taken from the corresponding papers when clearly reported; otherwise, we mark them as NR. Pretrain/SFT denotes approximate pretraining and supervised fine-tuning data scale, respectively. † indicates tool-augmented samples.

Model	Backbone LLM	Data Size (Pretrain/SFT)	Training Strategy
LLaVA-1.5	Vicuna-7B	558,000/665,000	2-stage (connector pretrain + visual instruction tuning)
MiniGPT-v2	LLaMA-2-7B	web-scale mixed data/NR	3-stage training
GeoChat	Vicuna-v1.5-7B	NR/∼318,000	LoRA SFT
RSGPT	Vicuna-7B	NR/2585	Q-Former + linear FT
LHRS-Bot	LLaMA-2-7B	∼1.15 million/∼74,000	3-stage curriculum training
VHM	Vicuna-v1.5-7B	∼1.4 million/>151,000	2-stage (RS pretrain + SFT)
EarthGPT	LLaMA-2	LAION-400M + COCO/>1 million	cross-modal alignment + RS tuning
InternVL2-8B	InternLM2-8B	NR/NR	MLP warmup + instruction tuning
Qwen2.5-VL-7B	Qwen2.5-7B	NR/NR	released pretrained/post-trained model
GeoPilot	**Vicuna-7B**	**595K (LLaVA init.) / 555,913 †**	**2-stage full SFT**

We also distinguish standard downstream evaluation from GeoPilotBench task-planning evaluation. For standard benchmarks such as VQA and captioning, we include published results when they are available and comparable. For task planning, however, each model must be executable under the same tool description prompt and its output must be parsable under the same `actions` schema. Therefore, RSGPT and LHRS-Bot are added as prompted RS-VLM planning baselines, while systems whose released resources, tool repositories, or action schemas are not directly compatible with GeoPilotBench are discussed separately rather than forced into the same numerical planning table.

3. Results

3.1. Task Planning Accuracy

To evaluate task planning, we construct a test set of 500 queries spanning all nine tool categories (including a "no-tool" category), stratified across three difficulty levels: easy (40%, direct tool requests), medium (40%, implicit needs or paraphrased instructions), and hard (20%, ambiguous or compound tasks). All compared models receive the same system prompt containing identical tool descriptions to ensure fair comparison. RSGPT and LHRS-Bot are added as prompted RS-VLM baselines because their released checkpoint/inference pipelines can be evaluated under the same tool-description prompt and parsed using the same `actions` schema. Since they are not originally designed as tool use agents, invalid, missing, or unparsable tool calls are counted as incorrect. We also include a GeoPilot variant without explicit tool use traces to isolate the effect of structured tool supervision.

As shown in Table 5, GeoPilot achieves 92.6% overall planning accuracy, substantially outperforming all baselines. Notably, Qwen2.5-VL, despite possessing native function-calling capabilities, achieves only 51.6% OPA, indicating that general-purpose tool use abilities do not transfer to domain-specific RS tools without targeted training. RSGPT and LHRS-Bot achieve 28.0% and 38.0% OPA, respectively, showing that RS-domain VLMs provide some task-intent awareness but remain far below GeoPilot without explicit tool use training. The GeoPilot w/o tool use traces variant drops to 34.6% OPA, confirming that ordinary RS instruction tuning alone is insufficient for reliable structured tool orchestration. GeoPilot also demonstrates strong no-tool judgment (NTA = 97.0%), correctly avoiding unnecessary tool invocations that would introduce latency and potential errors.

Table 5. Comparison of task planning performance on GeoPilotBench (500 samples). TSA: Tool Selection Accuracy; PA: Parameter Accuracy; NTA: No-Tool Judgment Accuracy; OPA: overall planning accuracy. All models receive the same system prompt with tool descriptions. RSGPT and LHRS-Bot are evaluated as prompted RS-VLM baselines.

Model	TSA	PA	NTA	OPA
LLaVA-1.5-7B [60]	38.6	42.4	53.0	22.8
RSGPT [26]	40.2	48.0	51.0	28.0
GeoChat [11]	46.8	55.6	61.0	32.4
GeoPilot w/o tool use traces	48.8	57.0	62.0	34.6
LHRS-Bot [28]	51.4	58.2	62.0	38.0
InternVL2-8B [61]	58.4	65.2	68.0	44.2
Qwen2.5-VL-7B [9]	64.2	71.8	72.0	51.6
GeoPilot (Ours)	**96.4**	**94.8**	**97.0**	**92.6**

The w/o tool use traces variant is not an unfine-tuned model. It uses the same Vicuna-7B backbone and a comparable fine-tuning setting as GeoPilot, but explicit reasoning fields, `actions`, tool names, API parameters, serialized tool outputs, and tool output-conditioned answer synthesis are removed from tool-augmented samples. The large OPA drop from 92.6% to 34.6% demonstrates that explicit tool use traces are the key supervision signal for reliable planning.

Table 6 provides a per-tool breakdown of Tool Selection Accuracy. GeoPilot achieves above 94% across all tool categories, whereas baselines show particularly weak performance on less common tools such as road extraction and cloud removal. Table 7 further shows that GeoPilot maintains 84.0% OPA even on hard queries involving ambiguous or compound instructions, while the representative baselines shown do not exceed 32.0%. These results suggest that explicit tool use supervision during training is indispensable for reliable RS tool orchestration, and that neither general-purpose function-calling ability nor RS domain expertise alone is sufficient.

Table 6. Per-tool breakdown of Tool Selection Accuracy (%) on the 500-sample task planning benchmark. # Samples denotes the number of samples in each tool category. ∅ denotes the no-tool category, where no external tool invocation is required.

Tool Category	# Samples	LLaVA-1.5	Qwen2.5-VL	InternVL2	GeoChat	GeoPilot
Object Detection	80	35.0	68.8	62.5	52.5	**97.5**
Panoptic Detection	50	22.0	56.0	48.0	40.0	**96.0**
Referring Object Detection	70	28.6	60.0	52.9	44.3	**97.1**
Sem. Segmentation	50	30.0	62.0	56.0	38.0	**96.0**
Inst. Segmentation	50	26.0	54.0	48.0	36.0	**94.0**
Ref. Segmentation	40	25.0	52.5	47.5	35.0	**95.0**
Road Extraction	30	16.7	43.3	36.7	30.0	**96.7**
Cloud Removal	30	20.0	50.0	40.0	33.3	**96.7**
No Tool (∅)	100	53.0	72.0	68.0	61.0	**97.0**
Overall TSA	500	38.6	64.2	58.4	46.8	**96.4**

Table 7. Overall planning accuracy (%) by difficulty level on the task planning benchmark. # Queries denotes the number of queries at each difficulty level.

Difficulty	# Queries	LLaVA-1.5	Qwen2.5-VL	InternVL2	GeoPilot
Easy	200	32.0	65.5	58.0	**97.0**
Medium	200	20.5	49.0	42.0	**93.0**
Hard	100	8.0	32.0	24.0	**84.0**

3.2. Visual Question Answering

On both RSVQA-LRBEN and the more challenging RSVQA-HRBEN [56] benchmarks, GeoPilot achieves competitive performance among the compared baselines (Tables 8 and 9). We additionally evaluate Qwen2.5-VL-7B and InternVL2.5-8B under the same protocols. On RSVQA-LRBEN, Qwen2.5-VL-7B and InternVL2.5-8B achieve 68.23% and 72.55% average accuracy, respectively. On RSVQA-HRBEN, they achieve 67.43% and 70.63%, respectively. These results show that recent general-purpose VLMs are competitive but do not dominate RS-specialized models on these benchmarks. GeoPilot reaches 93.21 $\pm$ 0.21% average accuracy on RSVQA-LRBEN and 73.65 $\pm$ 0.20% on RSVQA-HRBEN. We note that these VQA improvements are modest and should not be interpreted as arising solely from the GeoPilot architecture. They may also reflect differences in training data scale, data diversity, and the two-stage curriculum. The main contribution of GeoPilot lies in learned tool orchestration and cross-domain optical/SAR support, while the VQA results mainly show that incorporating tool use supervision does not degrade general visual understanding and can yield slight but consistent gains.

Table 8. Comparison of GeoPilot with VLMs on RSVQA-LRBEN. Results are reported using accuracy (%). Presence, Comparison, and Rural/Urban denote the three question types in the RSVQA-LRBEN benchmark. GeoPilot results are mean $\pm$ std over three seeds.

Model	Presence	Comparison	Rural/Urban	Average
LLaVA-1.5 [60]	55.46	68.20	59.00	62.77
MiniGPTv2 [6]	55.16	55.22	39.00	54.96
Qwen2.5-VL-7B [9]	60.64	73.89	66.00	68.23
InternVL2.5-8B [62]	71.47	73.39	71.00	72.55
LHRS-Bot [28]	88.51	90.00	89.07	89.19
VHM [34]	90.11	89.89	88.00	89.33
GeoChat [11]	91.09	90.33	94.00	90.70
RSGPT [26]	91.03	91.70	94.00	92.29
GeoPilot	**92.10** $\pm$ 0.32	**91.91** $\pm$ 0.26	**95.21** $\pm$ 0.40	**93.21** $\pm$ 0.21

Table 9. Comparison of GeoPilot with VLMs on RSVQA-HRBEN. Results are reported using accuracy (%). Presence and Comparison denote the two question types in RSVQA-HRBEN. GeoPilot results are mean ± std over 3 seeds.

Model	Presence	Comparison	Average
MiniGPTv2 [6]	40.79	50.91	46.46
Qwen-VL [7]	66.44	60.41	63.06
Qwen2.5-VL-7B [9]	60.38	72.98	67.43
InternVL2.5-8B [62]	64.51	75.44	70.63
EarthGPT [29]	62.77	79.53	72.06
GeoChat [11]	58.45	83.19	72.30
InternVL2-8B [61]	**67.35**	76.91	72.70
VHM [34]	63.00	83.00	73.00
GeoPilot	63.51 ± 0.35	**83.90** ± 0.16	**73.65** ± 0.20

3.3. Referring Object Detection

As shown in Table 10, GeoPilot outperforms all compared VLM baselines on GeoChat-Instruct for referring object detection (i.e., visual grounding conditioned on a natural-language referring expression). The improvements are especially notable in the Multiple category, suggesting that tool-augmented grounding is particularly beneficial when the target object must be distinguished among several candidates. This finding highlights a key advantage of the tool-augmented paradigm: by delegating fine-grained localization to a specialist grounding model (RemoteSAM), GeoPilot can focus its language reasoning on disambiguating the referring expression, leading to stronger performance in multi-object scenarios where purely end-to-end VLMs tend to struggle.

Table 10. Comparison of GeoPilot with VLMs on GeoChat-Instruct. Results are reported using Acc@0.5. Small, Medium, and Large refer to object size categories; Single and Multiple indicate whether the referring expression targets one or multiple objects in the image.

Model	Small	Medium	Large	Single	Multiple
MiniGPTv2 [6]	1.70	9.90	21.90	9.10	3.60
GeoChat [11]	2.90	13.60	21.70	16.00	4.30
InternVL2-8B [61]	7.20	23.76	31.99	25.77	9.30
GeoPilot (Ours)	**10.63**	**25.11**	**32.45**	**26.32**	**16.84**

3.4. SAR Visual Question Answering

As shown in Table 11, GeoPilot substantially outperforms all baselines on SAR VQA, achieving improvements of over 15 points in Object Identification and nearly 10 points in Object Positioning compared to GeoChat. These results demonstrate the effectiveness of our two-stage training strategy with SAR-specific instruction data. The large margins across all four sub-tasks reveal that the SAR domain poses unique challenges that cannot be addressed by optical-domain knowledge alone: SAR imagery exhibits fundamentally different visual characteristics—such as speckle noise, radar shadows, and geometric distortions from layover—that require targeted supervision. Notably, even GeoChat, which is trained on substantial RS data, shows limited SAR understanding, confirming that optical RS expertise does not transfer to SAR without explicit adaptation.

Table 11. Comparison of GeoPilot with VLMs on SAR VQA. OI: object identification; IC: instance counting; OC: object classification; OP: object positioning. Results are reported using accuracy.

Model	OI	IC	OC	OP
LLaVA-1.5 [60]	53.46	45.20	29.00	12.77
Qwen2.5-VL [9]	55.46	47.20	25.00	13.23
GeoChat [11]	62.04	54.36	51.54	19.84
GeoPilot	**77.46**	**67.24**	**62.64**	**29.72**

3.5. SAR Image Captioning

Table 12 compares GeoPilot against both zero-shot and fine-tuned baselines on the official SARLANG-1M-Cap test split, which contains 13,682 caption samples. Compared with the originally used 6000-sample held-out subset, this official split provides a more standardized and reproducible evaluation protocol. We verified that the original subset and the official test split have no overlap with the training data at either the image level or the image–caption pair level.

Under the official split, GeoPilot obtains a lower CIDEr score than in the original 6000-sample evaluation. This decrease is expected because the official split is larger and more diverse, and CIDEr is sensitive to TF-IDF-weighted domain-specific n-gram overlap and repeated caption templates. GeoPilot achieves higher BLEU-2, BLEU-3, BLEU-4, and CIDEr than the strongest fine-tuned baseline, while Qwen2.5-VL remains stronger on BLEU-1 and ROUGE-L. We therefore interpret CIDEr together with BLEU and ROUGE-L rather than as a standalone indicator of captioning quality.

Among zero-shot baselines, InternVL2.5 obtains the strongest captioning results, suggesting that general multimodal capacity helps but remains insufficient without SAR-domain adaptation. Fine-tuning substantially improves all models, while GeoPilot's advantage is most visible in higher-order BLEU and CIDEr rather than unigram overlap. This pattern is consistent with the domain-specific phrase matching behavior of SAR captions.

Table 12. Comparison on SAR image captioning using the official SARLANG-1M-Cap test split with 13,682 caption samples. Baseline results are reported from the official SARLANG-1M-Cap benchmark when available. GeoChat and GeoPilot are evaluated under the same official split. ZS: zero-shot; FT: fine-tuned.

Model	Param	Setting	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr
LLaVA-1.5 [60]	7B	ZS	8.18	3.96	1.69	0.76	13.39	0.03
Qwen2-VL [63]	7B	ZS	7.43	3.53	1.52	0.69	12.70	0.01
GeoChat [11]	7B	ZS	11.62	5.42	2.43	1.28	14.78	0.17
DeepSeek-VL [64]	7B	ZS	20.79	10.04	4.65	2.60	18.39	3.68
Qwen2.5-VL [9]	7B	ZS	21.49	10.77	6.39	3.46	21.09	6.14
InternVL2.5 [62]	8B	ZS	28.81	19.31	13.56	8.68	29.97	10.50
LLaVA-1.5 [60]	7B	FT	29.61	18.86	13.14	9.21	27.12	28.31
Qwen2-VL [63]	7B	FT	30.18	19.43	13.99	10.10	27.56	28.96
Qwen2.5-VL [9]	7B	FT	**35.27**	26.65	21.38	17.31	**34.37**	62.85
GeoPilot	7B	FT	32.74	**26.91**	**21.74**	**17.86**	33.18	**70.24**

3.6. End-to-End Tool Task Evaluation

To verify that GeoPilot's task planning accuracy translates into practical utility, we conduct end-to-end evaluation on two representative tool-augmented tasks. We compare GeoPilot against the Standalone Tool baseline, which bypasses GeoPilot entirely and invokes the specialist tool directly with ground-truth parameters. This comparison isolates the overhead introduced by GeoPilot's learned tool selection and parameter generation.

As shown in Table 13, the performance gap between GeoPilot and the standalone tool is only 1.5 mIoU on semantic segmentation and 3.0 Acc@0.5 on referring object detection, confirming that the learned tool policy introduces minimal overhead. Combined with the 92.6% overall planning accuracy in Table 5, these results demonstrate that GeoPilot serves as a reliable orchestrator for specialist tools.

Table 13. End-to-end evaluation of tool-augmented tasks. *Standalone Tool* directly invokes the specialist tool with ground-truth parameters, while *GeoPilot* autonomously selects the tool and generates parameters. Semantic segmentation is evaluated on the official OpenEarthMap test split using Segearth-OV, and referring object detection is evaluated on the GeoChat-Instruct test split using RemoteSAM.

Task	Metric	Standalone Tool	GeoPilot (Ours)	Gap
Semantic Segmentation	mIoU	39.8	38.3	−1.5
Referring Object Detection	Acc@0.5	29.1	26.1	−3.0

For semantic segmentation, we evaluate on the official OpenEarthMap test split using Segearth-OV [52] as the specialist tool. The 6000 training samples in our instruction dataset are drawn exclusively from the OpenEarthMap training split, ensuring no data leakage. For referring object detection, we use the GeoChat-Instruct test split with RemoteSAM [51] as the specialist tool.

3.7. Ablation Study: Training Strategy and Optical Data Mixing Ratio

To validate the effectiveness of our training strategy, we conduct comprehensive ablations examining both the two-stage curriculum and the optical data mixing ratio α used during stage 2. We evaluate on SAR VQA (SARDet-100K) and optical VQA (RSVQA-HRBEN) simultaneously, enabling assessment of cross-domain performance balance.

Table 14 reveals several important findings. First, comparing Row 1 and Row 2, even without optical data replay, the two-stage curriculum improves SAR Avg from 35.98% to 53.18%. This demonstrates that separating optical-domain alignment and SAR-domain adaptation provides a substantially better starting point for SAR learning than single-stage mixed training. Although HRBEN also increases from 64.46% to 67.08% in this comparison, it remains far below the replayed settings, indicating that optical replay is still needed to preserve optical-domain capability.

Table 14. Ablation study on training strategy and optical data mixing ratio α. SAR Avg denotes the average performance over the four SAR VQA subtasks, while HRBEN Avg denotes the average accuracy on RSVQA-HRBEN. The first row (X) denotes single-stage training on mixed optical and SAR data without curriculum separation. A checkmark indicates that multi-stage training is used.

Multi-Stage	α	OI	IC	OC	OP	SAR Avg	HRBEN Avg
X	–	54.49	47.46	26.38	15.59	35.98	64.46
✓	0%	70.95	61.28	56.47	24.02	53.18	67.08
✓	10%	72.38	62.15	58.72	25.40	54.66	71.24
✓	**20%**	**77.46**	**67.24**	**62.64**	**29.72**	**59.27**	73.68
✓	30%	76.82	66.50	61.90	28.94	58.54	73.96
✓	50%	73.60	63.18	57.46	26.30	55.14	73.70

Second, Rows 2–6 reveal that SAR and optical performance exhibit complementary trends as α varies. SAR performance peaks at $\alpha = 20\%$ (59.27%) and then declines, whereas optical performance improves rapidly from $\alpha = 0\%$ to $\alpha = 20\%$ and then saturates around $\alpha = 20$–30%. At $\alpha = 0\%$, HRBEN accuracy remains only 67.08%, which is substantially below the replayed settings, confirming that optical capability is not sufficiently preserved

without replay. At $\alpha = 20\%$, optical performance reaches 73.68%, only 0.28 points below the $\alpha = 30\%$ configuration and 0.02 points below the $\alpha = 50\%$ configuration.

Third, $\alpha = 20\%$ achieves the best cross-domain balance: SAR VQA reaches its maximum while optical VQA is nearly fully preserved. Further increasing α to 30% or 50% yields only marginal HRBEN gains (+0.28 and +0.02) at the cost of SAR degradation (-0.73 and -4.13). This validates our selection of $\alpha = 20\%$ as the optimal operating point for the final GeoPilot training.

3.8. Qualitative Analysis

Figure 5 presents qualitative examples that illustrate GeoPilot's breadth of capabilities. In optical imagery, the model supports both holistic understanding tasks and fine-grained spatial operations. In SAR imagery, it demonstrates strong cross-domain generalization on tasks such as SAR-specific grounding, positioning, and panoptic detection. Particularly in grounded tasks, GeoPilot is able not only to identify relevant targets but also to provide interpretable spatial descriptions that align with the tool outputs.

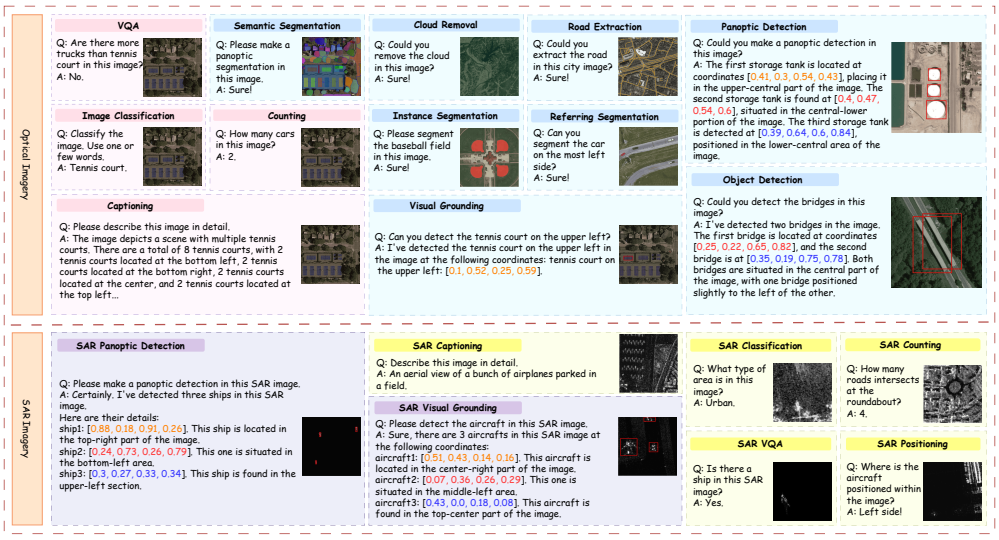

Figure 5. Qualitative results demonstrating GeoPilot's versatility across optical and SAR imagery. Each example shows the user query (Q) and model response (A). Tool-augmented tasks additionally display the visual output produced by the invoked specialist tool. The examples illustrate the model's ability to address both general understanding and tool-augmented tasks, highlighting robust multimodal reasoning and grounded response generation across diverse remote sensing scenarios.

Figure 6 presents additional qualitative comparisons on SAR VQA tasks. GeoPilot generally provides more accurate and contextually grounded answers compared to the other VLMs evaluated. For instance, in object identification and counting tasks, our model correctly interprets complex SAR image features where other models often fail. In the spirit of transparent analysis, we also present several failure cases in the lower panel of Figure 6. Typical errors occur under conditions of high ambiguity or when dealing with object categories that are underrepresented in the training data. For example, the model may misclassify visually similar objects (e.g., a highway interchange confused with a bridge) or struggle with precise counting when objects are densely clustered. These failure modes highlight directions for future improvement through more diverse data augmentation and more sophisticated visual reasoning modules.

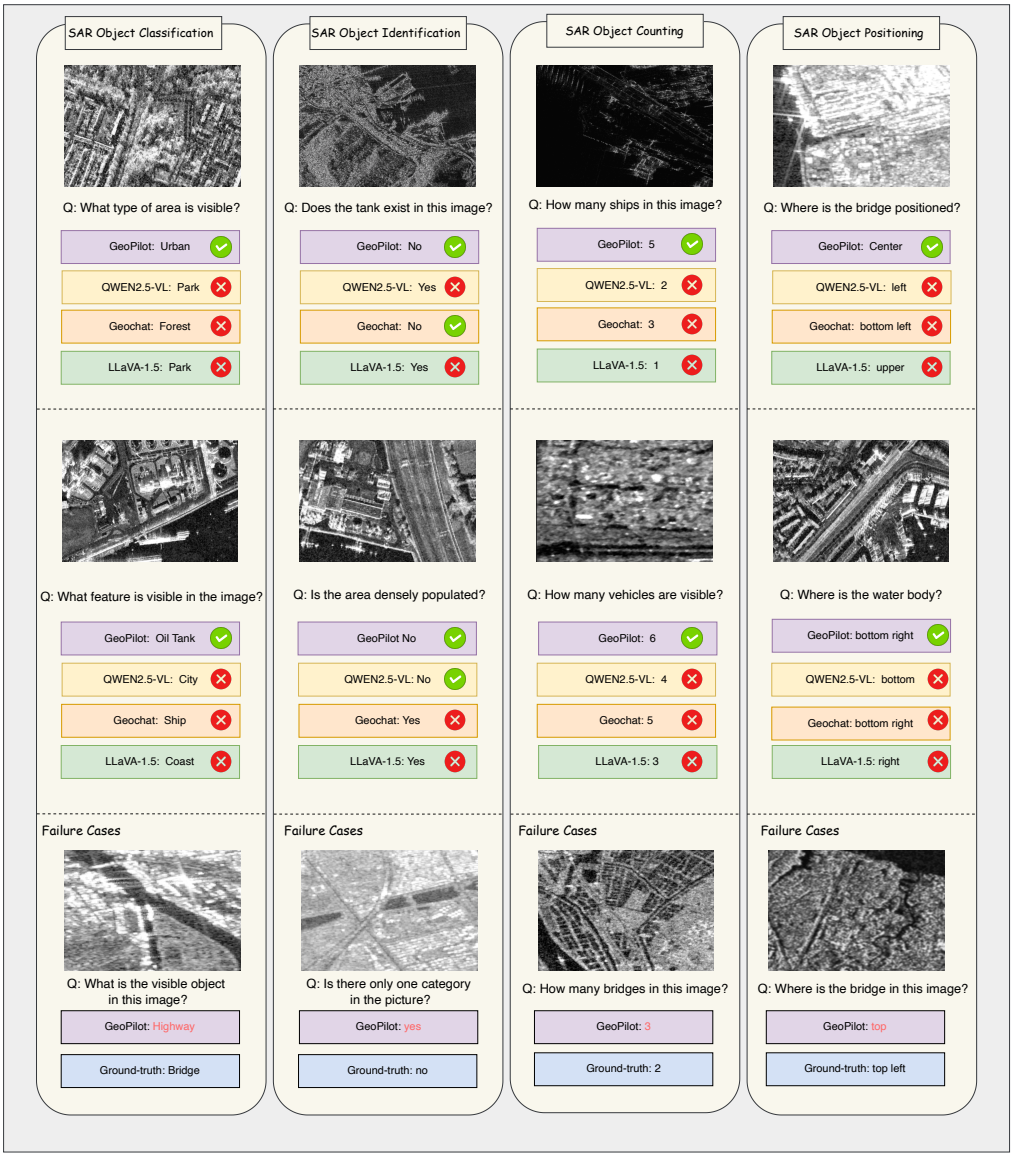

Figure 6. Qualitative results on SAR VQA tasks compared with other VLMs, including failure cases. Green checkmarks and red crosses indicate correct and incorrect answers, respectively. Failure cases highlight typical error modes such as misclassifying visually similar objects and imprecise counting in cluttered scenes.

4. Discussion

The experimental results support several key observations, which we discuss in turn.

Finding 1: Tool use training is essential and non-transferable.

The task planning evaluation reveals a notable gap between models with and without explicit tool use training. GeoPilot achieves 92.6% OPA, while the strongest baseline (Qwen2.5-VL, which has native function-calling support) reaches only 51.6%. Additional prompted RS-VLM baselines, including RSGPT and LHRS-Bot, remain far below GeoPilot, and the GeoPilot w/o tool use traces variant drops to 34.6% OPA. This indicates that providing tool descriptions at inference time is insufficient; models must be trained on structured tool use traces to internalize reliable planning policies. Unlike prompt-based tool use, GeoPilot learns tool invocation policies directly from structured tool use traces during instruction tuning, which is key to its strong planning performance. We note that GeoPilot's advantage is most pronounced in tool orchestration and SAR understanding, rather than in

general VQA. This is consistent with our design goal: GeoPilot's primary contribution lies in learned tool invocation policies and cross-domain training, not in replacing task-specific VQA models.

Finding 2: Cross-domain training benefits both modalities.

The ablation on training strategy and optical replay ratio demonstrates that both curriculum design and moderate replay are important for building robust cross-domain RS assistants. Single-stage mixed training shows weak SAR adaptation and limited optical retention. The two-stage setting without replay improves SAR performance and moderately improves HRBEN accuracy, but HRBEN remains substantially below the replayed settings, indicating that optical replay is still necessary to preserve optical-domain capability. A moderate replay ratio ($\alpha = 20\%$) achieves the best overall balance, where SAR VQA reaches its highest average score and optical VQA remains close to the best HRBEN result. This favorable trade-off validates our training strategy and provides practical guidance for future cross-domain multimodal training.

Finding 3: Tool-augmented tasks benefit from orchestration rather than raw model capacity.

For tool-augmented tasks such as referring object detection and semantic segmentation, the final output quality depends on both correct tool selection by the model and accurate execution by the specialist tool. The end-to-end evaluation (Table 13) confirms that GeoPilot introduces minimal overhead compared to standalone tool execution (-1.5 mIoU on semantic segmentation, -3.0 Acc@0.5 on referring object detection), validating the framework's role as a reliable orchestrator. This design philosophy is important for scalability: as better specialist tools become available, GeoPilot can benefit from them without retraining.

Finding 4: SAR captioning performance is strong but domain-dependent.

On the official SARLANG-1M-Cap test split, GeoPilot achieves the best CIDEr score among the compared fine-tuned models, but the absolute CIDEr value is lower than that obtained on the original 6000-sample subset. This more conservative result is expected because the official split is larger and more diverse, and CIDEr is sensitive to domain-specific n-gram overlap. GeoPilot remains stronger on higher-order BLEU metrics and CIDEr, while Qwen2.5-VL remains stronger on BLEU-1 and ROUGE-L. This suggests that GeoPilot captures domain-specific phrases effectively, but CIDEr should be interpreted together with other captioning metrics.

SAR-specific challenges and future tool integration.

Recent SAR-specific perception studies further highlight the importance of robustness under speckle noise, ocean clutter, sparse scattering, small targets, and quantity-aware vessel reasoning. For example, triple-level sparsity-aware ship surveillance [65], density knowledge mining for quantity-aware vessel monitoring [66], eagle-eye vision-inspired progressive screening [67], and tri-state prototype self-distillation for SAR ocean panoptic segmentation [68] address perception challenges that are complementary to GeoPilot. Although these methods are not direct RS-VLM or tool-planning baselines, they could serve as stronger SAR-specialized tools in future extensions of the GeoPilot tool repository.

Limitations and future directions.

Several limitations point to future work. First, the tool repository is manually curated, constraining adaptability to new tasks. Automatic tool discovery and integration remain open challenges. Second, the current focus on optical and SAR imagery excludes other important modalities such as hyperspectral, LiDAR, and temporal sequences. Third, the

two-turn interaction format, while effective for many RS tasks, does not naturally support complex multi-step reasoning or iterative tool chains. Fourth, GeoPilot currently reports object locations in normalized image coordinates relative to the input image, rather than as georeferenced coordinates such as latitude/longitude or projected map coordinates. This follows the annotation format of the benchmark datasets and the input/output convention of the specialist tools, which typically provide bounding boxes, masks, or normalized coordinates. Many benchmark images are cropped or resized and do not consistently provide complete geospatial metadata, such as coordinate reference systems, affine transforms, RPC parameters, or GeoTIFF metadata. In practical GIS-oriented applications, future versions of GeoPilot could integrate such metadata to transform image-local boxes, masks, or points into georeferenced outputs. Finally, while the GeoPilot framework is architecturally compatible with other multimodal LLM backbones, all experiments in this paper use Vicuna-v1.5 (7B); evaluating stronger or different backbone families is left for future work.

5. Conclusions

In this paper, we addressed two major limitations of existing RS vision–language systems: their predominant restriction to optical imagery and their limited ability to dynamically orchestrate specialist tools. We introduced GeoPilot, a modular multimodal assistant trained on a large-scale instruction dataset spanning both optical and SAR data and incorporating explicit tool use supervision. To evaluate this setting systematically, we proposed GeoPilotBench, a benchmark that covers optical and SAR tasks for both general understanding and tool-augmented reasoning. GeoPilot achieves strong performance on task planning (92.6% OPA), SAR understanding, and referring object detection. End-to-end evaluation confirms that the learned tool policy introduces minimal overhead (-1.5 mIoU on semantic segmentation, -3.0 Acc@0.5 on referring object detection), validating GeoPilot's role as a reliable tool orchestrator. On general VQA benchmarks, GeoPilot achieves competitive results, with improvements primarily attributable to the scale and diversity of the cross-domain training data. Comprehensive ablation studies validate the importance of the two-stage training curriculum and the optical data mixing strategy. Overall, our findings suggest that robust RS assistants should combine flexible tool-centric architectures, diverse cross-domain instruction data, and standardized evaluation protocols. This work takes a step toward practical geospatial assistants and opens up future directions in broader modality support, automatic tool expansion, and more advanced multi-step reasoning.

Author Contributions: Conceptualization, X.Z., X.W., Z.Q., Y.S., H.K. and T.W.; methodology, X.Z. and X.W.; validation, X.Z. and X.W.; writing—original draft preparation, X.Z.; writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Scholarship Council (CSC) grant numbers 202308070070 (X.Z.) and 202308070072 (X.W.).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available and cited in the manuscript. The GeoPilot codebase, the constructed instruction dataset, and the GeoPilotBench benchmark resources are publicly available at https://github.com/xuanZhou111/GeoPilot (accessed on 10 April 2026). During peer review, these materials can be made available to the editors and reviewers upon reasonable request.

Acknowledgments: During the preparation of the training dataset (Section 2.3), the authors used Gemini 2.5 Flash for the purposes of generating instruction–response pairs and structured tool use

reasoning traces. The authors validated the generated outputs through structural validation, semantic consistency checks, and manual audit as described in Section 2.3.4, and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Curran Associates Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 34892–34916.
2. Liu, S.; Cheng, H.; Liu, H.; Zhang, H.; Li, F.; Ren, T.; Zou, X.; Yang, J.; Su, H.; Zhu, J.; et al. LLaVA-Plus: Learning to Use Tools for Creating Multimodal Agents. In *Proceedings of the Computer Vision–ECCV 2024*; Springer: Cham, Switzerland, 2024; pp. 126–142. [CrossRef]
3. Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. How far are we to gpt-4v? Closing the gap to commercial multimodal models with open-source suites. *Sci. China Inf. Sci.* **2024**, *67*, 220101. [CrossRef]
4. Chen, K.; Zhang, Z.; Zeng, W.; Zhang, R.; Zhu, F.; Zhao, R. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv* **2023**, arXiv:2306.15195.
5. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
6. Chen, J.; Zhu, D.; Shen, X.; Li, X.; Liu, Z.; Zhang, P.; Krishnamoorthi, R.; Chandra, V.; Xiong, Y.; Elhoseiny, M. MiniGPT-v2: Large Language Model as a Unified Interface for Vision-Language Multi-Task Learning. *arXiv* **2023**, arXiv:2310.09478.
7. Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; Zhou, J. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv* **2023**, arXiv:2308.12966.
8. Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024*; IEEE: New York, NY, USA, 2024; pp. 24185–24198.
9. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL Technical Report. *arXiv* **2025**, arXiv:2502.13923. [CrossRef]
10. Zhan, Y.; Xiong, Z.; Yuan, Y. SkyEyeGPT: Unifying Remote Sensing Vision-Language Tasks via Instruction Tuning with Large Language Model. *ISPRS J. Photogramm. Remote Sens.* **2025**, *221*, 64–77. [CrossRef]
11. Kuckreja, K.; Danish, M.S.; Naseer, M.; Das, A.; Khan, S.; Khan, F.S. GeoChat: Grounded Large Vision-Language Model for Remote Sensing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024*; IEEE: New York, NY, USA, 2024; pp. 27831–27840.
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML 2021), Virtual, 18–24 July 2021; Volume 139, pp. 8748–8763.
13. Li, J.; Li, D.; Xiong, C.; Hoi, S. BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation. In Proceedings of the 39th International Conference on Machine Learning (ICML 2022), Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 12888–12900.
14. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023), Honolulu, HI, USA, 23–29 July 2023; Volume 202, pp. 19730–19742.
15. Dai, W.; Li, J.; Li, D.; Tiong, A.M.H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; Hoi, S. InstructBLIP: Towards General-Purpose Vision-Language Models with Instruction Tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Curran Associates Inc.: Red Hook, NY, USA, 2023; pp. 49250–49267.
16. Ruan, J.; Chen, Y.; Zhang, B.; Xu, Z.; Bao, T.; Du, G.; Shi, S.; Mao, H.; Li, Z.; Zeng, X.; et al. TPTU: Large language model-based AI agents for task planning and tool usage. *arXiv* **2023**, arXiv:2308.03427. [CrossRef]
17. Newell, A.; Simon, H.A. Computer Science as Empirical Inquiry: Symbols and Search. *Commun. ACM* **1976**, *19*, 113–126. [CrossRef]
18. Brooks, R.A. Intelligence without Representation. *Artif. Intell.* **1991**, *47*, 139–159. [CrossRef]
19. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
20. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]

21. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1126–1135.
22. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022*; Curran Associates Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 27730–27744.
23. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
24. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Curran Associates Inc.: Red Hook, NY, USA, 2023; Volume 36.
25. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and Its Friends in Hugging Face. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Curran Associates Inc.: Red Hook, NY, USA, 2023; Volume 36.
26. Hu, Y.; Yuan, J.; Wen, C.; Lu, X.; Liu, Y.; Li, X. Rsgpt: A remote sensing vision language model and benchmark. *ISPRS J. Photogramm. Remote Sens.* **2025**, *224*, 272–286. [CrossRef]
27. Guo, H.; Su, X.; Wu, C.; Du, B.; Zhang, L.; Li, D. Remote Sensing ChatGPT: Solving Remote Sensing Tasks with ChatGPT and Visual Models. In *Proceedings of the IGARSS 2024—2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024*; IEEE: New York, NY, USA, 2024; pp. 11474–11478. [CrossRef]
28. Muhtar, D.; Li, Z.; Gu, F.; Zhang, X.; Xiao, P. LHRS-Bot: Empowering Remote Sensing with VGI-Enhanced Large Multimodal Language Model. In *Proceedings of the Computer Vision–ECCV 2024, Milan, Italy, 29 September–4 October 2024*; Springer: Cham, Switzerland, 2025; pp. 440–457.
29. Zhang, W.; Cai, M.; Zhang, T.; Zhuang, Y.; Mao, X. EarthGPT: A Universal Multimodal Large Language Model for Multisensor Image Comprehension in Remote Sensing Domain. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5917820. [CrossRef]
30. Zhang, W.; Cai, M.; Zhang, T.; Li, J.; Zhuang, Y.; Mao, X. EarthMarker: Visual Prompt Learning for Region-Level and Point-Level Remote Sensing Imagery Comprehension. *arXiv* **2024**, arXiv:2407.13596.
31. Chen, W.; Chen, J.; Deng, Y.; Chen, J.; Feng, Y.; Xi, Z.; Liu, D.; Li, K.; Meng, Y. LRSCLIP: A Vision-Language Foundation Model for Aligning Remote Sensing Image with Longer Text. *arXiv* **2025**, arXiv:2503.19311.
32. Lin, H.; Hong, D.; Ge, S.; Luo, C.; Jiang, K.; Jin, H.; Wen, C. RS-MoE: A Vision-Language Model with Mixture of Experts for Remote Sensing Image Captioning and Visual Question Answering. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5614918. [CrossRef]
33. Yao, K.; Xu, N.; Yang, R.; Xu, Y.; Gao, Z.; Kitrungrotsakul, T.; Ren, Y.; Zhang, P.; Wang, J.; Wei, N.; et al. Falcon: A Remote Sensing Vision-Language Foundation Model. *arXiv* **2025**, arXiv:2503.11070. [CrossRef]
34. Pang, C.; Weng, X.; Wu, J.; Li, J.; Liu, Y.; Sun, J.; Li, W.; Wang, S.; Feng, L.; Xia, G.S.; et al. VHM: Versatile and Honest Vision Language Model for Remote Sensing Image Analysis. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025*; AAAI Press: Palo Alto, CA, USA, 2025; Volume 39, Number 6, pp. 6381–6388. [CrossRef]
35. Liu, F.; Dai, W.; Zhang, C.; Zhu, J.; Yao, L.; Li, X. Co-LLaVA: Efficient remote sensing visual question answering via model collaboration. *Remote Sens.* **2025**, *17*, 466. [CrossRef]
36. Bazi, Y.; Bashmal, L.; Al Rahhal, M.M.; Ricci, R.; Melgani, F. RS-LLaVA: A Large Vision-Language Model for Joint Captioning and Question Answering in Remote Sensing Imagery. *Remote Sens.* **2024**, *16*, 1477. [CrossRef]
37. Xu, W.; Yu, Z.; Mu, B.; Wei, Z.; Zhang, Y.; Li, G.; Wang, J.; Peng, M. RS-Agent: Automating Remote Sensing Tasks through Intelligent Agent. *arXiv* **2026**, arXiv:2406.07089. [CrossRef]
38. Wei, Y.; Xiao, A.; Ren, Y.; Zhu, Y.; Chen, H.; Xia, J.; Yokoya, N. SARLANG-1M: A benchmark for vision-language modeling in SAR image understanding. *IEEE Trans. Geosci. Remote Sens.* **2026**, *64*, 5201320. [CrossRef]
39. Li, Y.; Li, X.; Li, W.; Hou, Q.; Liu, L.; Cheng, M.M.; Yang, J. SARDet-100K: Towards Open-Source Benchmark and Toolkit for Large-Scale SAR Object Detection. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024*; Curran Associates Inc.: Red Hook, NY, USA, 2024; 32p.
40. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Curran Associates Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 46595–46623.

41. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018*; IEEE Computer Society: Los Alamitos, CA, USA, 2018; pp. 3974–3983. [CrossRef]
42. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
43. Sun, X.; Wang, P.; Yan, Z.; Xu, F.; Wang, R.; Diao, W.; Chen, J.; Li, J.; Feng, Y.; Xu, T.; et al. FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 116–130. [CrossRef]
44. Dong, Z.; Sun, Y.; Liu, T.; Zuo, W.; Gu, Y. Cross-Modal Bidirectional Interaction Model for Referring Remote Sensing Image Segmentation. *arXiv* **2025**, arXiv:2410.08613 [CrossRef]
45. Xia, J.; Yokoya, N.; Adriano, B.; Broni-Bediako, C. OpenEarthMap: A Benchmark Dataset for Global High-Resolution Land Cover Mapping. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023*; IEEE: New York, NY, USA, 2023; pp. 6254–6264.
46. Zamir, S.W.; Arora, A.; Gupta, A.; Khan, S.; Sun, G.; Shahbaz Khan, F.; Zhu, F.; Shao, L.; Xia, G.S.; Bai, X. iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 28–37.
47. Van Etten, A.; Lindenbaum, D.; Bacastow, T.M. Spacenet: A remote sensing dataset and challenge series. *arXiv* **2018**, arXiv:1807.01232.
48. Hetang, C.; Xue, H.; Le, C.; Yue, T.; Wang, W.; He, Y. Segment Anything Model for Road Network Graph Extraction. *arXiv* **2024**, arXiv:2403.16051. [CrossRef]
49. Lin, D.; Xu, G.; Wang, X.; Wang, Y.; Sun, X.; Fu, K. A Remote Sensing Image Dataset for Cloud Removal. *arXiv* **2019**, arXiv:1901.00600. [CrossRef]
50. Pan, J.; Liu, Y.; Fu, Y.; Ma, M.; Li, J.; Paudel, D.P.; Van Gool, L.; Huang, X. Locate Anything on Earth: Advancing Open-Vocabulary Object Detection for Remote Sensing Community. In *Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025*; AAAI Press: Palo Alto, CA, USA, 2025; Volume 39, Number 6, pp. 6281–6289. [CrossRef]
51. Yao, L.; Liu, F.; Chen, D.; Zhang, C.; Wang, Y.; Chen, Z.; Xu, W.; Di, S.; Zheng, Y. RemoteSAM: Towards Segment Anything for Earth Observation. In *Proceedings of the 33rd ACM International Conference on Multimedia, Dublin, Ireland, 27–31 October 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 3027–3036. [CrossRef]
52. Li, K.; Liu, R.; Cao, X.; Bai, X.; Zhou, F.; Meng, D.; Wang, Z. SegEarth-OV: Towards Training-Free Open-Vocabulary Segmentation for Remote Sensing Images. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025*; IEEE: New York, NY, USA, 2025; pp. 10545–10556.
53. Pan, H. Cloud Removal for Remote Sensing Imagery via Spatial Attention Generative Adversarial Network. *arXiv* **2020**, arXiv:2009.13015. [CrossRef]
54. Dai, Y.; Zou, M.; Li, Y.; Li, X.; Ni, K.; Yang, J. DenoDet: Attention as Deformable Multisubspace Feature Denoising for Target Detection in SAR Images. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *61*, 4729–4743. [CrossRef]
55. Doshi, T. Start Building with Gemini 2.5 Flash. *Google Developers Blog*, 17 April 2025. Available online: https://developers.googleblog.com/en/start-building-with-gemini-25-flash/ (accessed on 10 April 2026).
56. Lobry, S.; Marcos, D.; Murray, J.; Tuia, D. RSVQA: Visual Question Answering for Remote Sensing Data. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8555–8566. [CrossRef]
57. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002*; Association for Computational Linguistics: Philadelphia, PA, USA, 2002; pp. 311–318. [CrossRef]
58. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL Workshop Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
59. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-Based Image Description Evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015*; IEEE: New York, NY, USA, 2015; pp. 4566–4575. [CrossRef]
60. Liu, H.; Li, C.; Li, Y.; Lee, Y.J. Improved Baselines with Visual Instruction Tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024*; IEEE: New York, NY, USA, 2024; pp. 26296–26306. [CrossRef]
61. OpenGVLab Team. InternVL2: Better than the Best—Expanding Performance Boundaries of Open-Source Multimodal Models with the Progressive Scaling Strategy. *InternVL Project Blog*, 4 July 2024. Available online: https://internvl.github.io/blog/2024-07-02-InternVL-2.0/ (accessed on 10 April 2026).
62. Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling. *arXiv* **2025**, arXiv:2412.05271.
63. Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv* **2024**, arXiv:2409.12191.

64. Lu, H.; Liu, W.; Zhang, B.; Wang, B.; Dong, K.; Liu, B.; Sun, J.; Ren, T.; Li, Z.; Yang, H.; et al. DeepSeek-VL: Towards Real-World Vision-Language Understanding. *arXiv* **2024**, arXiv:2403.05525.
65. Zhang, T.; Zhang, X. Triple-Level Sparsity Awareness for Marine Ship Surveillance Using Satellite Synthetic Aperture Radar. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 5155–5166. [CrossRef]
66. Zhang, T.; Zhang, X.; Gao, G. Density Knowledge Mining for Quantity-Aware Marine Vessel Surveillance Using Satellite SAR Data. *IEEE Trans. Ind. Inform.* **2026**, *early access*.
67. Zhang, T.; Gao, G.; Zhang, X. Glance-Focus-Gaze: A Novel Eagle-Eye Vision-Inspired Panorama-Population-Individual Progressive Screening Paradigm to Capture Ships in SAR Images. *ISPRS J. Photogramm. Remote. Sens.* **2026**, *235*, 241–260. [CrossRef]
68. Deng, R.; Zhang, T.; Xu, X.; Zhang, X.; Gao, G. Tri-State Prototype Self-Distillation for SAR Ocean Imagery Panoptic Segmentation. *IEEE Geosci. Remote. Sens. Lett.* **2026**, *early access*. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.