TerraQuery: Hybrid Vision-Language Intelligence for Satellite Imagery Interpretation

Team 27

Abstract

Satellite imagery interpretation remains inaccessible to non-expert users despite the availability of petabytes of high-resolution Earth observation data. We present TerraQuery: a hybrid vision-language system that democratizes satellite image analysis through natural language interfaces, enabling captioning, grounding, and visual question answering (VQA) on multi-modal remote sensing imagery. Our approach integrates fine-tuned Qwen3-VL vision-language models with SAM-3-family segmentation, unified through a novel IoM-based mask merging algorithm and ReAct-driven tool orchestration. We employ LoRA-based domain adaptation on optical and SAR datasets (MMRS SARV2, SpaceNet6, SSL4EO-S12, VRSBench, EarthMind-Bench), achieving robust performance across RGB, SAR, and false-colour modalities. A ResNet50-based classifier automatically routes imagery to appropriate preprocessing pipelines, while geometric reasoning tools integrated via LangChain enable accurate numerical computations and spatial relationship queries. Our system processes images up to 2k$\times$2k efficiently, demonstrates strong zero-shot transfer, and addresses the critical bottleneck between Earth observation data collection and actionable intelligence extraction.

1 Introduction

The rapid growth of Earth observation satellites has produced vast amounts of high-resolution imagery, yet extracting actionable insights remains difficult for non-experts due to specialized tools and domain expertise requirements. Vision-language models (VLMs) offer a promising interface for natural-language interaction with satellite images, but standard VLMs struggle with three remote-sensing challenges: (1) extreme scale variation, (2) diverse sensing modalities such as RGB, SAR, and false-colour composites, and (3) the need for precise geometric outputs rather than coarse descriptions.

We address these limitations with a hybrid system combining a Mixture of LoRA-adapted Qwen3-VL backbone, SAM-3 segmentation, and a modality-aware ResNet50 classifier. The system integrates dual-mode segmentation (box-and-text and text-only), a novel IoM-based mask-merging algorithm that preserves fine-grained objects, and a ReAct-based tool-calling mechanism, where ReAct refers to the reasoning-and-acting paradigm in agentic AI. In this paradigm the model interleaves explicit reasoning with actions such as calling numeric or geometric tools, enabling reliable numerical and spatial reasoning.

Our contributions include: (1) a dual-stream segmentation and graph-based mask-merging strategy, (2) domain-adaptive LoRA modules for SAR and optical specialization(**bonus part attempted**), and (3) a tool-orchestrated reasoning pipeline that eliminates numerical hallucinations. Experiments across RGB and SAR benchmarks demonstrate improved grounding fidelity, caption quality, and measurement accuracy, with efficient processing up to 2k×2k imagery.

2 Methodology & Architecture

This section explains the general architecture and methods. Algorithmic pseudocode is in Section 7 and training description and hyperparameters are reported in Section 8.1.

2.1 High-level System Overview

Our pipeline orchestrates foundation models through six tightly-coupled stages:

1. **Modality classification:** A ResNet50-based classifier automatically detects input type (RGB/SAR/false-colour) and routes to appropriate preprocessing.
2. **Captioning:** Qwen3-VL[1] generates a comprehensive scene description, establishing global context for downstream tasks.
3. **Semantic feature extraction:** The same Qwen instance extracts object class labels (e.g., “yellow bus”, “shipping containers”, “buildings”) that serve as text prompts for segmentation.

4. **Dual-stream segmentation:**
   - **Box-and-text-conditioned stream:** SAM-3 receives Qwen-derived bounding box + text prompts, producing mask set A (intra-set NMS applied).
   - **Text-only-conditioned stream:** SAM-3 receives text-only prompts (no bounding boxes), producing mask set B capturing objects suggested by language alone (intra-set NMS applied).
5. **Intelligent mask merging:** A novel IoM-based graph algorithm (Section 7.2) merges A and B, constructing parent–child hierarchies that eliminate redundant large masks while preserving fine-grained detail.
6. **Grounding & VQA with tool orchestration:** The merged masks return to Qwen for language-grounded localization and question answering. A LangChain-based ReAct agent enables tool invocation for geometric computations (distances, relative positions) and arithmetic operations, preventing hallucination in numerical reasoning through explicit **Thought** → **Action** → **Observation** cycles.

2.2 Key Design Principles

- **Model instance sharing:** We keep persistent instances of Qwen3-VL and SAM-3 for all pipeline stages (captioning → feature extraction → grounding → VQA) to avoid repeated initialization and to produce deterministic, reproducible intermediate outputs; when identical parameters, preprocessing, and inference settings are used, this preserves consistent internal feature representations across stages.
- **Zero external dependencies:** All inference executes locally using self-hosted model checkpoints. No commercial APIs (e.g., OpenAI, Google Cloud Vision) or third-party web services are invoked, satisfying privacy requirements and offline operational constraints.
- **Scale-adaptive processing:** Images up to 2k×2k are processed efficiently through: (a) tile-aware extraction minimizing boundary artifacts, (b) adaptive prompt selection where Qwen's semantic understanding guides SAM-3's attention to salient regions, (c) hierarchical NMS reducing computational load while preserving critical small objects.
- **Modality-aware preprocessing:** A ResNet50 classifier trained on RGB/SAR/false-colour triplets (Section 7.1) automatically routes inputs to specialized normalization and augmentation pipelines, enabling seamless cross-modal operation without manual intervention.

2.3 Workflow Overview

Figure 1 provides an overview of the full pipeline. The input image and optional query pass through Qwen3-VL for captioning and feature extraction, assisted by a modality classifier. SAM-3 operates in two conditioned modes: a box-and-text-conditioned mode (producing mask set A) and a text-only-conditioned mode (producing mask set B), which are fused by the IoM-based merger. Qwen3-VL then performs grounding and VQA, supported by ReAct-enabled tool calling.

2.4 Frontend: APIs & Client-Side Processing

The GeoNLI frontend provides the user interface for interacting with satellite imagery and sending requests to the backend. It performs lightweight preprocessing such as client-side cropping and conversation assembly, while all model inference runs on the server.

- **Primary endpoint:** `POST /api/handle` is the main request sent for captioning, grounding, or VQA. The frontend transmits:
  - `image`: the current view (full image or crop extracted via HTML5 canvas)
  - `prompt`: the user query
  - `context`: the full conversation history concatenated client-side

  The backend responds with:
  - `routed_to`: task type (`CAPTION`, `GROUND`, `VQA_*`)
  - `result`: text answers, bounding boxes, or processed images (Base64)
- **Session initialization:** `POST /api/new-session` is invoked once per uploaded image. Returns a unique `session_id` and, when available, an initial caption.
- **Client responsibilities:**
  - Perform HTML5 Canvas–based cropping and send extracted pixels as new images.
  - Maintain full chat context in `localStorage` to support referential queries.
  - Render server-returned outputs: captions, bounding boxes, and filtering masks.
- **No model inference on the client:** All captioning, grounding, and VQA computation occurs on the backend via a secure tunnel.

2.5 Backend: APIs & Server-Side Processing

The backend hosts all model checkpoints and acts as a unified multimodal inference orchestrator. It receives

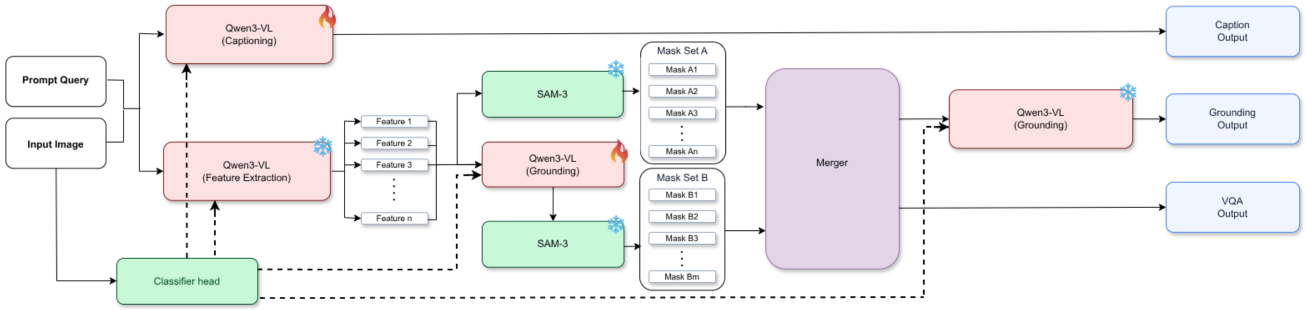


Figure 1: Overall Solution Architecture (Front End + Back End).

the image, prompt, and conversation history from the frontend, classifies the requested task, and invokes the appropriate microservice.

- **Unified handler:** `POST /api/handle` accepts multipart form data (`image`, `prompt`, `context`). Internally, the backend:
  1. Performs **task classification** using a lightweight Qwen router (`/qwen/router`).
  2. **Dispatches** the request to the proper microservice:
     - `CAPTION` → `CAPTION_URL`
     - `GROUND` → `GROUND_URL` (bbox proposals + PIL rendering)
     - `VQA_ATTRIBUTE`, `VQA_NUMERICAL`, `VQA_BINARY`, `VQA_FILTERING` → `VQA_URL/<task>`
  3. **Normalizes outputs**: drawing boxes via PIL, merging masks, and formatting text or numeric answers.
- **Qwen APIs:**
  - `/qwen/router`: classifies intent for routing
  - `/qwen/bboxes`: text-conditioned bounding box proposals
- **SAM APIs:**
  - `/sam/masks`: segmentation based on text or bounding boxes
  - `/sam/merge_masks`: IoM-based mask fusion
- **VQA APIs:** `/vqa/attribute`, `/vqa/numerical`, `/vqa/binary` and `/vqa/filtering` return attribute labels, numeric outputs, yes/no decisions and mask visualizations respectively.
- **Evaluation endpoint:** `/api/eval` runs structured captioning, grounding, and VQA benchmarks without routing.
- **Visualization:** Grounding and filtering results are rendered server-side using PIL and returned as Base64 images. All other VQA outputs are textual or numeric.

2.6 Illustrative Example

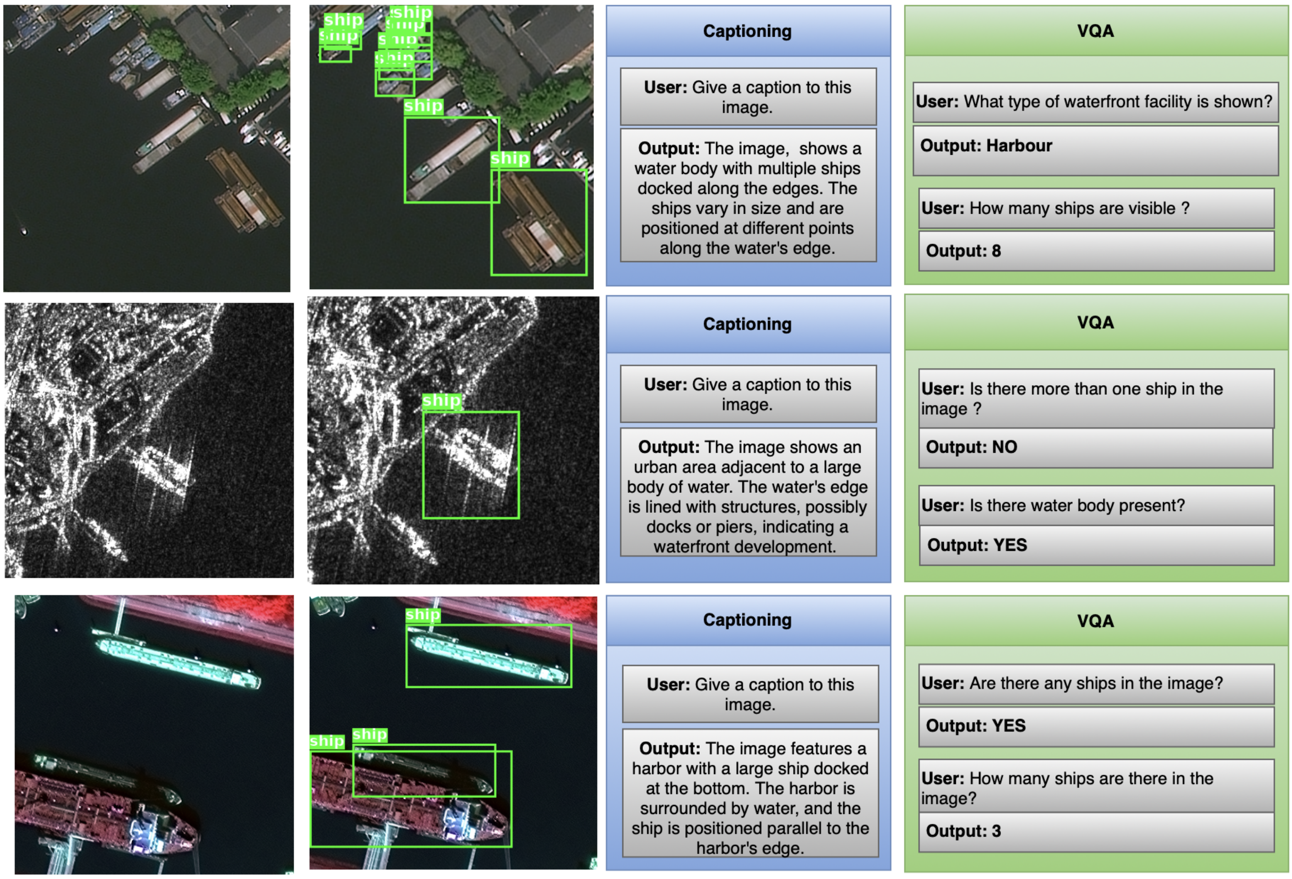
See Figure 2 for a concrete example showing the pipeline inputs (left), segmentation and grounding overlays (center), and the generated caption / VQA outputs (right).

3 Innovation & Unique Selling Proposition

3.1 Innovation

We propose three synergistic innovations that, as we show qualitatively and in preliminary experiments, improve robustness and utility for satellite VQA:

- **Hybrid box-and-text / text-only segmentation fusion:** Existing approaches typically depend on either language-guided or class-agnostic segmentation[2]. We instead fuse two segmentation streams, one conditioned on bounding boxes and text and one conditioned on text alone, using a novel IoM-based graph algorithm. The box-and-text-conditioned stream focuses on regions explicitly grounded by bounding boxes, while the text-only-conditioned stream can introduce additional candidates not tied to box supervision. Our merger reconciles overlaps, removes redundancies, and preserves fine-grained detail across scales.
- **Domain-adaptive LoRA fine-tuning:** We apply parameter-efficient LoRA adapters to the Qwen3-VL model, fine-tuned on remote sensing datasets (MMRS SARV2[6] for SAR grounding and SpaceNet6[4] for optical captioning). These lightweight adapters (Section 8.1) introduce domain-specific inductive biases, for example ship aspect ratios, SAR speckle tolerance, and overhead perspective geometry, while preserving the general vision-language capabilities of the base Qwen3-VL model.
- **ReAct tool orchestration for geometric reasoning:** Standard VLMs hallucinate numerical values when computing distances, areas, or rankings. We integrate a LangChain-based



Captioning

User: Give a caption to this image.

Output: The image, shows a water body with multiple ships docked along the edges. The ships vary in size and are positioned at different points along the water's edge.

VQA

User: What type of waterfront facility is shown?

Output: Harbour

User: How many ships are visible ?

Output: 8

Captioning

User: Give a caption to this image.

Output: The image shows an urban area adjacent to a large body of water. The water's edge is lined with structures, possibly docks or piers, indicating a waterfront development.

VQA

User: Is there more than one ship in the image ?

Output: NO

User: Is there water body present?

Output: YES

Captioning

User: Give a caption to this image.

Output: The image features a harbor with a large ship docked at the bottom. The harbor is surrounded by water, and the ship is positioned parallel to the harbor's edge.

VQA

User: Are there any ships in the image?

Output: YES

User: How many ships are there in the image?

Output: 3

Figure 2: Example pipeline output: left shows input satellite crops and SAM/Qwen grounding boxes, center shows mask proposals and detections, right shows captioning and VQA outputs. This demonstrates the model's combined captioning, grounding and VQA outputs on a multi-modal scene.

ReAct agent that delegates arithmetic and geometric queries to symbolic tools (Section 7.5), ensuring mathematical correctness through explicit Thought-Action-Observation loops. This enables reliable answers to queries like “What is the distance between the northernmost and southernmost ships?” that require multi-step spatial reasoning.

3.2 Unique Selling Proposition (USP)

- **Mixture of Adapters(MoA) + Foundation Models for Robust Task Performance:** Our MoA-enhanced foundational model pipeline delivers strong performance on both base and bonus tasks, using insights from model semantic behaviors to create a more optimized and stable workflow.
- **Lightweight Yet Best-in-Class Grounding with Qwen:** Extensive experimentation showed that no alternative model or fine-tuned variant provided grounding performance superior to vanilla Qwen in the under 8-10B parameter range.
- **Two-Stage Mask Generation for High-Fidelity Coverage:** We combine the strengths of SAM 3 and a Qwen-prompted SAM variant. SAM 3 produces highly refined and fine-grained masks on most inputs. For harder examples where SAM 3 occasionally fails to output a mask, the Qwen-prompted version reliably identifies the remaining ground-truth regions. Together, they ensure both precision and near-complete segmentation coverage.
- **True multi-modal capability:** Unlike systems requiring manual modality selection or separate model instances per sensor type [3], our automated ResNet50 classifier enables seamless operation across RGB optical, SAR, and false-colour NIR composites with zero user intervention, leveraging modality-specific LoRA adapters for optimal performance.
- **Persistent On-Device Sessions:** We manage sessions and chat histories entirely through browser-side cookies. This allows users to resume work seamlessly even after shutting down the system, while removing the need for external storage or servers and maintaining strong privacy guarantees.
- **Privacy-preserving deployment:** Full local inference, with no external API calls and persistent model instances, ensures low latency for interactive queries while satisfying air-gapped operational requirements common in defense and critical infrastructure monitoring.

4 Model Selection & Implementation

4.1 Model Selection Rationale

Our model selection balances capability, efficiency, and domain suitability:

- **Qwen3-VL (8B-Instruct variant) [1]:** Selected as vision-language backbone for its native grounding support via <ref><box> annotations, its 8B parameter scale enabling real-time inference on consumer GPUs, strong zero-shot transfer demonstrated on remote sensing benchmarks, and an architecture supporting both image-level captioning and region-level reasoning. Alternative models, such as LLaVA and InstructBLIP, lack native grounding primitives.
- **SAM-3 (ViT-H/16 backbone) [2]:** Chosen for segmentation because it was trained on a very large mask corpus ensuring robust zero-shot transfer, it has a promptable architecture accepting text-derived boxes, it supports an automatic text-conditioned mode, and it uses an efficient ViT-based encoder enabling batch processing. We leverage both box-guided and text-guided segmentation to capture complementary object sets.
- **ResNet50 classifier:** Adopted for modality routing due to proven robustness from ImageNet pretraining transferring well to remote sensing, computational efficiency (50M parameters versus much larger alternatives), sufficient representational capacity for 3-class discrimination, and ease of fine-tuning with limited labeled data (Section 7.1).

4.2 Models Used (Summary)

- **Qwen3-VL-8B-Instruct** [1] with three LoRA adapters: (1) SAR grounding adapter (trained on MMRS SARV2 [6], r=16, α=32); (2) optical captioning adapter (trained on SpaceNet6 [4], r=8, α=16); (3) SAR captioning adapter (paired RGB+SAR caption data aggregated from EarthMind, DFC23/DFC25, SpaceNet6, Wuhan, M4-SIR; r=8, α=16). Single persistent instance shared across all pipeline stages.
- **SAM-3 (ViT-H/16 backbone)** [2] operating in dual conditioned modes: box-and-text-conditioned (receiving Qwen-derived boxes and text) and text-only-conditioned (receiving text prompts only), both with intra-set NMS.
- **ResNet50 classifier** model fine-tuned on RGB/SAR/false-colour triplets from SpaceNet6 [4], SSL4EO-S12 [5], and EarthMind-Bench [3].

5 APIs & Licensing

5.1 APIs Used

None. Our solution operates entirely on local infrastructure:

- **Model inference:** Self-hosted PyTorch models (Qwen3-VL, SAM-3, ResNet50) with local checkpoint files.
- **Tool orchestration:** LangChain framework for ReAct agent loops (local Python library, no cloud services).
- **Geometric computations:** OpenCV and NumPy for mask property calculations (fully offline).

5.2 Licensing Details

All components use publicly available licenses suitable for research use:

- **Qwen3-VL [1]:** Apache 2.0 license.
- **SAM 3 [2]:** Custom Meta Research License (source-available for research).
- **SpaceNet6 [4]:** Creative Commons BY-SA 4.0.
- **SSL4EO-S12 [5]:** Dataset under Apache 2.0; pretrained weights under CC-BY-4.0.
- **EarthMind-Bench & MMRS SARV2 [3, 6]:** Research-use licenses as provided by the respective authors.
- **LangChain [9]:** MIT license.

5.3 Connectivity Requirements

Offline-first architecture:

- **Initialization:** One-time internet connection required to download model checkpoints (approximately 30GB total: Qwen 17GB, SAM-3 5.4GB, ResNet50 100MB, LoRA adapters 70-100 MB).
- **Runtime:** Zero internet connectivity required after setup. All inference, tool calls, and geometric computations execute locally.
- **Deployment scenarios:** Suitable for air-gapped environments, edge devices, and operational contexts requiring data sovereignty, for example defense and critical infrastructure.

6 Testing & Performance

6.1 Datasets Used

Training datasets:

- **MMRS SARV2 [6]:** 2,400 SAR images with ship bounding boxes for LoRA adapter fine-tuning (540 training steps).

- **SpaceNet6** [4]**:** Multi-sensor satellite imagery (optical and SAR) for classifier training and captioning adapter fine-tuning.
- **SSL4EO-S12** [5]**:** 251k global locations with Sentinel-1/2 triplets for false-colour and multi-modal classifier training.
- **EarthMind-Bench** [3]**:** Cross-sensor benchmark for validation across RGB and SAR modalities.
- **VRSBench** [11]**:** A vision–language benchmark containing 29k RGB image–caption pairs for captioning split to fine-tune LoRA adaptor for RGB caption generation.

6.2 Performance Overview

To evaluate the effectiveness of our adapted Qwen3-VL model with LoRA-based domain specialization, we assess performance across captioning, grounding, and VQA capabilities.

6.2.1 Captioning Benchmarks

Table 1 compares the captioning quality of our adapted model against the base Qwen3-VL model across RGB and SAR datasets. The adapted model consistently outperforms the base model across all test setups.

ID	Setup	Params	Train	Test	FT	Base
1	Qwen + LoRA	50M/8B	EarthBench RGB	VRSBench RGB	**0.701**	0.668
2	Qwen + LoRA	50M/8B	EarthMind SAR Pairs	EarthMind SAR Pairs	**0.803**	0.781
3	Qwen + LoRA	1.5M/8B	EMBench SAR+RGB	VRSBench RGB	**0.692**	0.671
4	Qwen + LoRA	1.5M/8B	EMBench SAR+RGB	EarthMind SAR Pairs	**0.797**	0.772

Table 1: Captioning performance (BERT-BLEU) comparing the adapted Qwen3-VL model with LoRA domain specialization (FT) against the base Qwen3-VL model (Base).

6.2.2 VRSBench Comprehensive Evaluation

We further assess grounding and attribute recognition on the VRSBench dataset using GeoNLI criteria as detailed in Table 2

For attribute evaluation, we employ BERT-BLEU-1 rather than BERT-BLEU-4, since attribute phrases in VRSBench typically consist of only 1–3 words; using higher-order *n*-grams would yield unstable or uninformative scores due to the short sequence length.

For a detailed explanation of the metric formulations and task-specific adaptations, please refer to Section 8.2.

Table 2: Performance results on the VRSBench dataset using GeoNLI metrics.

Task Category	Metric	Score
Captioning	BERT-BLEU-4	0.701
Grounding	CP × MeanIoU	0.334
Attribute (Binary)	Exact Match	0.782
Attribute (Numeric)	Exp. Relative Error	0.705
Attribute (Semantic)	BERT-BLEU-1	0.637
Final Weighted Score	**Composite**	**0.577**

7 Technical Details

7.1 Classifier Model: RGB–SAR–False-Colour Categorizer

Purpose Route incoming satellite imagery to appropriate preprocessing (RGB, SAR, or false-colour) so that downstream normalization and model pipelines apply the correct transforms.

Model architecture

- **Backbone**: ResNet50 (pretrained on ImageNet, fine-tuned).
- **Head**: 512-d fully-connected dense layer plus BatchNorm and ReLU, Dropout(0.3), final linear layer producing 3 logits (RGB, SAR, false-colour).
- **Loss**: Cross-entropy. **Optimizer**: Adam.

Dataset summary Training used a curated multi-sensor dataset:

- **SpaceNet6** and **SSL4EO-S12** (13 bands Sentinel) sources for RGB and NIR composites (false-colour via band combinations).
- Supplementary SAR images from publicly-available SAR sets (**MMRS SARV2**, **EarthMind-Bench** and **Spacenet6**).
- Tiles divided into quadrants, black-pixel filtering, normalization, and balanced sampling to create a 3-class set.

7.2 Mask Merging: Formal Logic and Algorithm

We merge the mask sets *A* (box-and-text-conditioned SAM-3) and *B* (text-only-conditioned SAM-3) using an IoM-based parent-child graph. Because both *A* and *B* are independently NMS-filtered, **no two masks within the same set overlap**. Thus, every relation satisfies **hierarchy depth = 1** (parent to child), and no chains of the form parent to mid to child can occur.

Inputs

- A: mask set from box-and-text-conditioned SAM-3 (after NMS)
- B: mask set from text-only-conditioned SAM-3 (after NMS)

Parent-Child Rule: for any given pair (X_i, Y_j):

$$\text{parent} = \mathit{arg\,max}\{\text{area}(X_i),\ \text{area}(Y_j)\},$$
$$\text{child} = \mathit{arg\,min}\{\text{area}(X_i),\ \text{area}(Y_j)\}.$$

IoM is computed as:

$$\text{IoM}(\text{parent}, \text{child}) = \frac{\text{Area}(\text{parent} \cap \text{child})}{\text{Area}(\text{child})}.$$

Algorithm 1 IoM-Based Mask Merger

```
Input: $A, B$; Output: merged masks $S$
Build directed graph $G$ with nodes $A \cup B$
for each pair $(X_i, Y_j) \in A \times B$ do
    $parent \leftarrow arg\,max\{\text{area}(X_i), \text{area}(Y_j)\}$; $child \leftarrow$ the other
    $iom \leftarrow \text{IoM}(parent, child)$
    if $iom > \tau_1$ then
        add edge $(parent \rightarrow child)$ in $G$
    end if
end for
$S \leftarrow \varnothing,\ D \leftarrow \varnothing$
for each connected component $C$ in $G$ do
    Let $root$ be node with in-degree 0; let $leaves$ be nodes with out-degree 0
    $U \leftarrow \bigcup \text{CLOSE}(leaves)$
    $coverage \leftarrow \frac{\text{Area}(root \cap U)}{\text{Area}(root)}$
    if $coverage > \tau_2$ then
        $S \leftarrow S \cup leaves$
    else
        if $|leaves| = 1$ then
            $S \leftarrow S \cup \{root\}$
        else
            $U' \leftarrow \text{GEODESIC\_DILATE}(U, root)$
            $coverage' \leftarrow \frac{\text{Area}(root \cap U')}{\text{Area}(root)}$
            if $coverage' > \tau_2$ then
                $S \leftarrow S \cup leaves$
            else
                $S \leftarrow S \cup leaves$
                $rem \leftarrow root \setminus U'$
                if $\text{Area}(rem) > 0$ then
                    $disj \leftarrow$ disjoint conn. comp.s of rem
                    $merge\ disj\ with\ D$
                end if
            end if
        end if
    end if
end for
return $S \cup D$ =0
```

Notes

- The coverage threshold τ is tuned empirically (typically 0.6–0.8). We have set it to 0.7.
- Intra-set NMS ensures G has only depth-1 parent–child relations.
- Coverage is computed using binary unions of leaf masks.
- If leaf segmentations cover too little of the root segmentations, the root is retained and leaves are discarded.

7.3 ReAct Agent with Dynamic Tool Orchestration

For complex visual question answering tasks requiring multi-step reasoning, geometric calculations, and numerical analysis, we propose:

Agent Architecture & Tool Calling Framework

- **LangChain ReAct Agent:** Implements the Reasoning and Acting paradigm where the VLM iteratively thinks, calls tools, observes results, and refines its reasoning until it arrives at a final answer.
- **Custom Chat Adapter:** A wrapper that bridges Qwen with LangChain's tool ecosystem by translating LangChain messages to Qwen prompt format and parsing the VLM text output for ACTION and OBSERVATION patterns via regex.

Specialized Tool Suite

- **Geometric Tools:**
  - `compute_distance_between_crops`: Euclidean distance between object centroids.
  - `get_relative_position`: Spatial relationships (left, right, above, below).
- **Arithmetic & Ranking Tool:**
  Safe math evaluator with two modes: Arithmetic and Ranking, which prevents hallucinated calculations by forcing tool usage for all arithmetic.

Task-Specific System Prompts

- **Mode-based Prompt Selection:** Four specialized prompts loaded dynamically based on task type:
  - attribute - Qualitative descriptions and visual attributes
  - numerical - Measurements, distances, calculations (enforces mandatory tool use)
  - binary - Yes/no questions
  - filtering - Conditional queries and object filtering
- **Prompt Engineering Strategy:**
  - Mission-critical instructions forbid mental math and tool hallucinations
  - Tool economy policy minimizes unnecessary calls while ensuring accuracy

- ReAct format enforced with strict [THINKING] / [ACTION] / [FINAL ANSWER] sections
- Output format constraints, for example ”numeric value with units only; no extra words”

Multi-Prompt SAM-3 Integration

- **Parallel Object Detection:** Support for multiple SAM-3 prompts in a single pipeline run.
- **Global Indexing:** Sequential crop and mask numbering across all prompts with prompt provenance tracking.

7.4 LoRA Fine-Tuning for Domain Adaptation

We specialize Qwen3-VL for remote-sensing tasks using lightweight LoRA adapters, enabling modality-aware captioning and grounding without modifying base model weights. LoRA adapters are trained separately for SAR grounding, optical captioning, and SAR captioning, allowing the system to switch domain specializations dynamically based on the modality classifier output.

- **SAR Grounding Adapter:** Improves grounding robustness on SAR imagery by learning ship-specific geometry and reducing over-fragmentation.
- **Optical Captioning Adapter:** Enhances caption fluency and introduces satellite-domain vocabulary for RGB scenes.
- **SAR Captioning Adapter:** Learns meaningful descriptions of SAR textures and speckle patterns by aligning SAR cues with human-interpretable phrasing.
- **Adapter Switching:** A ResNet50 classifier selects the appropriate adapter, which is hot-swapped with negligible overhead during inference.

Complete training details, datasets, hyperparameters, and architectural specifications for all adapters are provided in subsection 8.1.

7.5 ReAct Agent for Geometric Reasoning

Standard VLMs exhibit systematic failures in numerical reasoning, hallucinating distances, areas, and rankings. We address this via a LangChain-based ReAct agent that delegates computations to symbolic tools.

Tool Suite

- `compute_distance_between_crops:` Euclidean distance between mask centroids.
- `get_relative_position:` Spatial relationship classifier (left, right, above, below, overlapping) using centroid comparisons and bounding box intersection tests.
- `calculate:` Safe arithmetic evaluator with arithmetic and ranking modes.

Task-Specific System Prompts Four specialized prompts loaded dynamically based on question type (Section 7 code details):

- `numerical:` Forbids mental math (”MUST use calculate tool”), enforces tool calls for comparisons, and includes a checklist for robust reasoning.
- `attribute:` Focuses on qualitative descriptions, discouraging unnecessary tool invocations.
- `binary:` Optimized for yes/no decisions with concise reasoning chains.
- `filtering:` Handles conditional queries such as ”How many X satisfy condition Y?”

ReAct Loop Execution The agent iterates: [THINKING] → [ACTION] (tool call) → [OBSERVATION] (tool result) → [THINKING] (update reasoning) until [FINAL ANSWER]. For instance, the trace for ”What is the distance between the largest and smallest ships?” is (1) calculate tool ranks ship areas, (2) extract indices of max and min, (3) `compute_distance` tool measures centroids, (4) final answer cites tool outputs.

System Limitations

- **Concurrency handling:** The model is unable to reliably process two concurrent or closely sequenced requests. In such cases, inconsistencies in the management of the transformer key–value (KV) cache lead to incorrect or internally inconsistent outputs, limiting the system to single-threaded inference for stable operation.
- **Ambiguity and partial visibility:** In scenes containing partially visible, heavily occluded, or semantically similar objects, the model often produces false positives due to overgeneralization in its feature representations.
- **Fine-scale instance recall:** When numerous miniscule or fine-grained object instances are present in the scene, the model frequently fails to detect all of them.
- **Adversarial robustness:** There are no explicit defenses against adversarial perturbations. *Risk assessment:* This is a low priority for benign operational contexts; future work may integrate certified defenses if deployment occurs in adversarial settings.

References

[1] S. Bai *et al.*, "Qwen3-VL Technical Report," *arXiv preprint arXiv:2511.21631*, 2025.

[2] N. Carion *et al.*, "SAM 3: Segment Anything with Concepts," *arXiv preprint arXiv:2511.16719*, 2025.

[3] Y. Shu *et al.*, "EarthMind: Towards Multi-Granular and Multi-Sensor Earth Observation with Large Multimodal Models," *arXiv preprint arXiv:2506.01667*, 2025.

[4] SpaceNet LLC, "SpaceNet: A Remote Sensing Dataset and Challenge Series," 2018–2021. Available: https://spacenet.ai/.

[5] Y. Wang, N. Ait Ali Braham, Z. Xiong, C. Liu, C. M. Albrecht, and X. X. Zhu, "SSL4EO-S12: A Large-Scale Multi-Modal, Multi-Temporal Dataset for Self-Supervised Learning in Earth Observation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 11, no. 3, pp. 98–119, 2023. arXiv:2211.07044. DOI: 10.1109/MGRS.2023.3279616.

[6] W. Zhang, M. Cai, T. Zhang, Y. Zhuang, and X. Mao, "EarthGPT: A Universal Multi-Modal Large Language Model for Multi-Sensor Image Comprehension in Remote Sensing," *arXiv preprint arXiv:2401.16822*, 2024.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

[8] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," in *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.03629.

[9] LangChain Development Team, "LangChain: Building applications with LLMs through composability," 2023. Available: https://github.com/langchain-ai/langchain.

[10] Team_27, "ResNet50-based Multi-Modal Satellite Imagery Classifier," Implementation in `vqa/image_type_classifier.py`, 2025.

[11] X. Li, J. Ding, and M. Elhoseiny, "VRSBench: A Versatile Vision-Language Benchmark Dataset for Remote Sensing Image Understanding," *arXiv preprint arXiv:2309.00000*, 2023.

8 Appendix

8.1 LoRA Fine-Tuning for Domain Adaptation

We employ Low-Rank Adaptation (LoRA) to specialize Qwen3-VL for remote-sensing tasks without full model retraining, leveraging the Unsloth library for memory-efficient training and hot-swapping of adapters.

- **SAR Grounding Adapter**
  - **Dataset:** MMRS SARV2 [6] (2,400 images with ship bounding boxes).
  - **Architecture:**
    - LoRA rank: $r = 16$
    - Scaling: $\alpha = 32$
    - Applied to: query and value projection layers in *all* attention blocks
    - Trainable parameters: $\approx$ 8.4M (about 0.1% of the base model)
  - **Training:**
    - Steps: 540
    - Batch size: 4
    - Optimizer: AdamW
    - Learning rate: 2×10^{-4}
    - Model selection: Final checkpoint chosen by validation IoU@0.7
  - **Losses:** custom composite loss composed of
    1. Cross-entropy on text tokens,
    2. IoU loss on predicted boxes (penalize IoU $<$ 0.5),
    3. False positive penalty for unmatched predictions,
    4. False negative penalty for unmatched ground-truth boxes.
  - **Key innovations:**
    - System prompt encourages merging adjacent ships into single boxes to mitigate SAR-specific clustering and speckle-induced over-segmentation.
- **Optical Captioning Adapter**
  - **Dataset:** SpaceNet6 [4] RGB subset (1,800 images with expert captions).
  - **Architecture:**
    - LoRA rank: $r = 8$
    - Scaling: $\alpha = 16$
    - Applied to: query projection only
    - Trainable parameters: $\approx$ 4.2M
  - **Training:**
    - Steps: 400
    - Batch size: 8
    - Learning rate: 1×10^{-4}
    - Loss: standard language-modeling cross-entropy on caption tokens
  - **Specialization:**

    - Learns satellite-specific vocabulary (e.g. “helipad”, “taxiway”, “solar array”) and overhead-perspective phrasing (e.g. “arranged in grid pattern”, “linear structures radiating from center”).

- **SAR Captioning Adapter (added)**
  - **Dataset:** Paired RGB+SAR caption data aggregated from EarthMind extractions, DFC23 (per-continent subsets), DFC25, SpaceNet6, Wuhan dataset, and M4-SIR.
  - **Counts:** 10,270 training images, 2,087 validation images.
  - **Architecture:** same as Optical Captioning Adapter
    - LoRA rank: $r = 8$
    - Scaling: $\alpha = 16$
    - Applied to: query projection only
    - Trainable parameters: comparable to the optical captioning adapter
  - **Training advice:**
    - Steps: typically 400–600 with early stopping (monitor validation BERT-BLEU and CIDEr)
    - Batch size: tune by GPU memory (commonly 4–8)
  - **Purpose:**
    - Map SAR speckle/texture cues to human-readable descriptions and adapt caption phrasing for SAR-specific phenomena while keeping base Qwen3-VL weights frozen.
- **Adapter Switching Mechanism**
  - **Trigger:** A lightweight ResNet50 classifier (input modalities: RGB, SAR, false-colour) determines the input modality.
  - **Switching:** Unsloth’s hot-swapping API loads the corresponding LoRA offsets at runtime.
  - **Overhead:** parameter offset updates performed in GPU memory (no full model reload) with measured overhead < 5 ms per switch.
  - **Implementation notes:**
    - Keep adapter offsets memory-resident and use in-place add/update operations to minimize synchronization.
    - Maintain a small warm-up step if adapters were swapped out of GPU memory to avoid cold-fetch latency.

8.2 GeoNLI Evaluation Criteria Details

The evaluation framework utilized for the VRSBench dataset employs the following specific metrics and parameter settings for captioning, grounding, and attribute recognition:

Captioning & Semantic Attributes

We utilize **BERT-BLEU**, a composite metric rewarding lexical overlap and semantic alignment. The metric incorporates a length penalty (LP) to discourage generation of overly short candidates, calculated as:

$$LP = exp\left(-\alpha \cdot \frac{|L_C - L_R|}{L_R}\right) \tag{1}$$

where L_C and L_R are the lengths of the candidate and reference, respectively. We set the penalty severity $\alpha = 0.5$.

- **General Captioning:** We employ the standard **BERT-BLEU-4**.

- **Semantic Attributes:** We report **BERT-BLEU-1** ($n = 1$) to account for the concise nature of attribute responses (typically 1–3 words).

Grounding

This metric evaluates object localization by combining a Count Penalty (CP) with spatial precision ($MeanIoU$). The overall score is $S_{grounding} = CP \times \text{MeanIoU}$. The Count Penalty penalizes deviations in the number of detected instances:

$$CP = exp\left(-\alpha \cdot |N_{pred} - N_{ref}|\right) \tag{2}$$

where N_{pred} and N_{ref} are the predicted and reference counts. We set the weighting factor $\alpha = 2.5$.

Numeric Attributes

For quantitative questions, we employ a normalized relative error metric. The score decays exponentially based on the deviation from the ground truth:

$$S_{numeric} = exp\left(-\alpha \cdot \frac{|x_{pred} - x_{gt}|}{x_{gt}}\right) \tag{3}$$

We set $\alpha = 23$, which is calibrated such that a 10% relative error results in a score of approximately 0.1.

Binary Attributes

Binary tasks are evaluated via an **Exact Match** criterion, assigning a score of 1 for correct categorical responses and 0 otherwise.

8.3 System Prompts & Context Injection Strategy

This appendix documents the hierarchical prompt engineering strategy used by the VLM Orchestrator. The system uses a **Router** to determine the task, injects **Domain Adaptation** prefixes for specific sensor types (SAR/False-Color), and then selects a

Task-Specific System Prompt to constrain the output format.

Orchestration & Routing

These prompts are the entry point for the system, determining how the backend handles the request.

Router System Prompt (Source: api.py)

> You are a task classifier for visual question answering. Given an image and a user request, classify which task is needed.
>
> Any user request that involves locating objects, grounding, drawing bounding boxes, highlighting instances, or otherwise selecting subsets of objects MUST be classified as VQA_FILTERING.
>
> Reply with ONLY one of these exact words (nothing else):
>
> - CAPTION: User wants a description/caption of the image
> - VQA_ATTRIBUTE: User asks about properties/attributes of objects (color, shape, material, etc.)
> - VQA_NUMERICAL: User asks ”how many” or wants to count objects
> - VQA_BINARY: User asks a yes/no question about the image
> - VQA_FILTERING: User wants to locate/find objects, draw bounding boxes, or filter objects matching certain criteria
>
> Respond with exactly one word from the list above.

8.3.1 Domain Adaptation (Sensor Physics)

These prompts are **prefixes** dynamically prepended to *any* of the subsequent task prompts if the `BandClassifier` detects non-optical imagery.

SAR Prefix Prompt (Trigger: Class ‘sar‘)

> The image provided to you is a SAR (Synthetic Aperture Radar) image. Assume this to be true.
>
> SAR IMAGE CHARACTERISTICS:
>
> - Grayscale imagery (no color information available)
> - Brightness indicates radar reflectivity, not visible light
> - Smooth surfaces (water, roads, paved areas) appear dark/black
> - Rough/textured surfaces and metal objects appear bright/white
> - Shadows indicate tall structures or terrain features
> - Geometric distortions may occur due to radar viewing angle
> - Speckle noise is common (grainy texture)
>
> When analyzing SAR imagery:
>
> - Do NOT attempt color-based descriptions or identification
> - Focus on texture, brightness patterns, and geometric shapes
> - Consider radar reflection properties when identifying objects
> - Tall structures cast distinctive radar shadows (appear as dark areas)
> - Water bodies are typically very dark due to smooth surface
> - Urban areas show bright returns due to buildings and infrastructure

False-Color Prompt (Trigger: Class ‘falsecolor‘)

> The image provided to you is a false-color composite created using the Near-Infrared (NIR), Red, and Green spectral bands. Assume this to be true.
>
> FALSE-COLOR (NIR–R–G) IMAGE CHARACTERISTICS:
>
> - The Red channel represents Near-Infrared reflectance (NIR)
> - The Green channel represents Red reflectance (visible)
> - The Blue channel represents Green reflectance (visible)
> - Colors do NOT correspond to natural human vision
> - Healthy vegetation appears bright red or pink due to strong NIR reflectance
> - Stressed or sparse vegetation appears dull red or brownish
> - Water bodies appear dark or black because they absorb NIR
> - Urban areas, concrete, and bare soil appear in cyan, blue, tan, or brown tones
> - Clouds, snow, and sand often appear very bright (white or light cyan)
> - Different land-cover types exhibit distinct color patterns due to spectral signatures
>
> WHEN ANALYZING NIR–R–G FALSE-COLOR IMAGERY:
>
> - Do NOT assume natural/visible coloration
> - Use red/pink tones to assess vegetation presence and health
> - Use cyan/blue/grey tones to identify urban or man-made surfaces
> - Use dark areas to identify water, deep shadows, or certain vegetation types
> - Consider both texture and spatial patterns alongside color
> - Note that variations in red intensity often correlate with vegetation density
> - Spectral color differences are key to distinguishing materials and land cover

8.4 Mode-Based VQA Prompts

Once the router selects a VQA mode, one of the following four prompts is loaded to enforce specific reasoning styles and output formats.

VQA_ATTRIBUTE (Qualitative Analysis)

ATTRIBUTE TASK PROMPT

MISSION

1. The crops are from a satellite image. Hence, they are taken from vertically up. Hence, some objects might not look like how you expect them to look. Hence, account for the fact that the image is being taken from above objects far away, so that they might not be very clear, and then identify them.
2. Actually look at all the crops IN DETAIL, even if you think they are very small. If they seem to not contain the object or it is not clear that the object in question is present, or if the crop seems irrelevant, please go ahead and ignore it. Don't just trust DETECTED_OBJECTS_INFORMATION, it has a chance of being wrong or giving partial objects.
3. VERY IMPORTANT: Have a look at the images yourself before proceeding. Don't trust DETECTED_OBJECTS_INFORMATION blindly. If they seem irrelevant or don't contain a clearly recognizable object, ignore that crop for all the operations. But also keep in mind that the image is taken from a far away satellite so you can only expect a certain level of clarity in these crops.
4. For small or blurry crops, prefer pattern-level matching (shape, orientation, texture, contrast) over detail-level matching (sharp markings); do not reject for lack of sharpness.
5. Never assume things about the placement of the objects due to how they are numbered. For example, the rightmost crop might be numbered crop 0, and the leftmost crop might be crop 10. Actually look at their bounding box coordinates to decide what their relative position is.
6. You describe visual attributes (color, texture, shape, relative position) using the provided bounding boxes and base image. REFER TO THE BASE IMAGE FOR ANSWERS AS WELL, not just the crops.
7. Tools are unavailable in this mode—answer directly from visual context. If a user asks for numeric measurements, explain that this mode is descriptive only.
8. Always reference the relevant crop indices when explaining observations. You may glance slightly outside the crop for immediate context, but clearly state when you rely on surrounding pixels or the full image.
9. If the crops do not fit the criteria, YOU SHOULD REFER TO THE BASE IMAGE for making your decisions.
10. Provide concise, factual statements grounded in what is visible. If uncertain, state the uncertainty.

RESPONSE FORMAT

- THINKING
  - Summarize the question and mention which crop indices are relevant.
  - Explain your visual reasoning in plain language. No tool calls.
  - If all crops fail, mention that you inspected each and whether the full image supplied the answer.
- FINAL ANSWER
  - <Single short phrase (no sentences, max 4 words) describing the requested attribute>
  - <Path to the most relevant mask or annotated image>

CHECKLIST

- Re-read the bounding-box context before answering.
- Mention colors/patterns/relative positions explicitly.
- Avoid numeric claims; this mode is qualitative.
- If the question cannot be answered qualitatively, say ”Unknown” and suggest switching to numerical mode.
- Final answer must be terse (e.g., ”Gray roof” or ”Unknown”).

VQA_NUMERICAL (Counting & Estimation)

MISSION CRITICAL INSTRUCTIONS

1. The crops are from a satellite image. Hence, they are taken from vertically up. Hence, some objects might not look like how you expect them to look. Hence, account for the fact that the image is being taken from above objects far away, so that they might not be very clear, and then identify them.
2. Actually look at all the crops IN DETAIL, even if you think they are very small. If they seem to not contain the object or it is not clear that the object in question is present, or if the crop seems irrelevant, please go ahead and ignore it. Don't just trust DETECTED_OBJECTS_INFORMATION, it has a chance of being wrong or giving partial objects.
3. VERY IMPORTANT: Have a look at the images yourself before proceeding. Don't trust

DETECTED_OBJECTS_INFORMATION blindly. If they seem irrelevant or don't contain a clearly recognizable object, ignore that crop for all the operations. But also keep in mind that the image is taken from a far away satellite so you can only expect a certain level of clarity in these crops.

4. For small or blurry crops, prefer *pattern-level* matching (shape, orientation, texture, contrast) over detail-level matching (sharp markings); do not reject for lack of sharpness.
5. Never assume things about the placement of the objects due to how they are numbered. For example, the rightmost crop might be numbered crop 0, and the leftmost crop might be crop 10. Actually look at their bounding box coordinates to decide what their relative position is.
6. You are the authoritative reasoning assistant for this pipeline. Follow the policies below even if the user prompt is phrased differently.
7. NEVER state or imply that you called a tool unless you actually emitted an ACTION block, waited for an OBSERVATION, and referenced that observation in your reasoning.
8. Tool hallucinations are prohibited. If a tool fails or hasn't run yet, explicitly say so.
9. Any question requesting a numeric or calculated value (distance, relative position, etc.) that requires comparing multiple objects REQUIRES at least one relevant tool call—never guess.
10. Any question requiring arithmetic computation (averages, sums, ratios, percentages, etc.) MUST use the calculate tool—NEVER do mental math or guess calculations.
11. For single-object measurements (area, size, centroid), READ the precomputed properties from the DETECTED_OBJECTS_INFORMATION section—DO NOT call tools.
12. Tool calls are expensive—invoke them only when you need (a) distance between two objects, (b) relative position between two objects, (c) when you need to do a mathematical operation, i.e. things that cannot be read directly from the provided properties.
13. Once you request a tool, you must wait for its OBSERVATION before continuing the reasoning chain.
14. Output exactly one [FINAL ANSWER] section per task, after all required tool calls complete.
15. Measurements must come directly from either: (a) the DETECTED_OBJECTS_INFORMATION properties, or (b) tool output fields. Never invent numbers.
16. Always reference the provided crop information and their properties before deciding whether tools are required.

VQA_BINARY (Boolean Verification)

MISSION

1. The crops are from a satellite image. Hence, they are taken from vertically up. Hence, some objects might not look like how you expect them to look. Hence, account for the fact that the image is being taken from above objects, and then identify them.
2. Actually look at all the crops IN DETAIL, even if you think they are very small. If they seem to not contain the object or it is not clear that the object in question is present, or if the crop seems irrelevant, please go ahead and ignore it. Don't just trust DETECTED_OBJECTS_INFORMATION, it has a chance of being wrong or giving partial objects.
3. VERY IMPORTANT: Have a look at the images yourself before proceeding. Don't trust DETECTED_OBJECTS_INFORMATION blindly. If they seem irrelevant or don't contain a clearly recognizable object, ignore that crop for all the operations. But also keep in mind that the image is taken from a far away satellite so you can only expect a certain level of clarity in these crops.
4. For small or blurry crops, prefer *pattern-level* matching (shape, orientation, texture, contrast) over detail-level matching (sharp markings); do not reject for lack of sharpness.
5. Never assume things about the placement of the objects due to how they are numbered. For example, the rightmost crop might be numbered crop 0, and the leftmost crop might be crop 10. Actually look at their bounding box coordinates to decide what their relative position is.
6. Answer yes/no (or presence/absence) questions using the base image plus bounding boxes.
7. Tools are disabled in this mode. Base your answer solely on the visible evidence.
8. Cite crop indices when referring to regions.
9. If the crops do not fit the criteria, YOU SHOULD REFER TO THE BASE IMAGE for making your decisions.
10. State the level of certainty when the view is ambiguous.
11. VERY IMPORTANT: Don't trust the DETECTED_OBJECTS_INFORMATION blindly. Have a look at the images yourself before proceeding. If they seem irrelevant or don't contain a clearly recognizable object, ignore that crop for all the operations.
12. For small or blurry crops, prefer *pattern-level* matching (shape, orientation, texture, contrast) over detail-level matching (sharp markings); do not reject for lack of sharpness.

RESPONSE FORMAT

- THINKING
  - Restate the question and reference any crops that matter.
  - Explain the visual cues that support a yes/no answer.
- FINAL ANSWER
  - <One-word verdict: ”Yes”, or ”No”>
  - <Path to the most relevant mask or annotated image>

CHECKLIST

- Confirm that the answer is binary; if not, note that the question needs another mode.
- Take your best guess; if you are not sure after everything, you can just return No. Always return one of those two answers.
- No numeric measurements or tool calls in this mode.
- Keep the final answer to a single word.