GeoViS: Geospatially Rewarded Visual Search for Remote Sensing Visual Grounding

Supplementary Material

6. Detailed Visual Search Process

To complement the search formulation in the main paper, this section provides additional details on the visual search mechanism used in GeoViS. We describe the action space and explain why the zoom-in and zoom-out operations are sufficient to explore the full spatial domain. We then clarify the motivation behind the semantic and IoU-based rewards and how they guide the search. Finally, we present supplementary ablations on search depth and simulation count to support the stability of our design choices.

6.1. Action Space Definition

This subsection expands on the design of the action space used in GeoViS and provides additional analysis of its feasibility for supporting effective search. At each step t, the algorithm resides at a state s_t corresponding to a rectangular region

$$I(s_t) \subset I_g.$$

GeoViS adopts a minimal but expressive action set consisting of: (i) a *zoom-in* operator that selects one of the 3×3 subregions of $I(s_t)$, and (ii) a *zoom-out* operator that enlarges $I(s_t)$ by a fixed factor. Together, these operations enable coarse-to-fine exploration and recovery of global context within MCTS.

Zoom-in. Following the definition in the main paper, the zoom-in operator $\mathcal{T}_{\text{in}}$ uniformly partitions the current region into a 3×3 grid and selects a subregion:

$$s_{t+1} = \mathcal{T}_{\text{in}}(s_t, a_t) = R_{i,j}(s_t).$$

The nine grid cells are indexed by $k \in \{1, \ldots, 9\}$ in row-major order. The mapping

$$\text{row} = \left\lfloor \frac{k-1}{3} \right\rfloor, \qquad \text{col} = (k-1) \bmod 3$$

uniquely identifies $R_{i,j}(s_t)$.

To determine k, we prompt the VisualRAG model with the textual query and image inputs. Crucially, we always include the global scene I_g to preserve absolute geospatial cues that cannot be recovered from local crops alone. At the root state, only I_g is used; at deeper states, both I_g and the current crop $I(s_t)$ are provided. This maintains consistency between global positioning ("far left of the scene", "north of the river") and local visual evidence.

Prompt at root step ($t = 0$):

```
prompt = (
    "Divide the image into a 3x3 grid (1 to 9).\n"
    f"Which region most likely contains the object
        : \"{text}\"?\n"
    "Answer with one number (1-9)."
)
images = [global_image]
```

Prompt at deeper steps ($t > 0$):

```
prompt = (
    "The first image is the full scene. The second
         image is a zoomed-in view.\n"
    "Divide the second image into a 3x3 grid (1 to
         9).\n"
    f"Which region most likely contains the object
        : \"{text}\"?\n"
    "Answer with one number (1-9)."
)
images = [global_image, local_crop]
```

The model's response is parsed to an integer grid index k, defaulting to the center cell ($k = 5$) for invalid predictions. The extracted (row, col) determines the next region $R_{i,j}(s_t)$.

The 3×3 subdivision naturally aligns with spatial expressions frequently used in remote sensing grounding. Unlike general image captions, remote sensing descriptions often refer to coarse geospatial positions such as "on the left side," "in the upper region," or "at the bottom-right corner." These phrases correspond directly to the semantic partition induced by the 3×3 grid, where each cell represents a directional concept (e.g., left, centre, right; top, middle, bottom). Moreover, the uniform decomposition provides a stable geometric hierarchy: each zoom-in step reduces the region size by exactly one-third, producing predictable spatial resolutions across search depths. Together, these properties make the zoom-in operator both semantically interpretable and computationally well-behaved, enabling progressive localization while remaining fully compatible with the MCTS search procedure.

At the same time, we acknowledge that more flexible variants could be explored to further improve adaptability to complex geometries. For instance, adaptive partition strategies or aspect-ratio-aware region proposals may better capture fine-grained boundaries or elongated targets, while maintaining the hierarchical search structure.

Zoom-out. As described in the main paper, the zoom-out operator enlarges the current region by a fixed factor $\lambda > 1$ to restore contextual information when the search becomes

overly localized. Here we provide additional implementation details.

Let the current region be $I(s_t) = [x_1, y_1, x_2, y_2]$ with width w_t and height h_t, and center

$$c_t = \left(\tfrac{x_1+x_2}{2},\ \tfrac{y_1+y_2}{2}\right).$$

Rather than scaling w_t and h_t independently, we expand the region using the dominant side length

$$d_t = \max(w_t, h_t), \qquad d_{t+1} = \lambda\, d_t,$$

This produces an approximately square enlarged region, which improves numerical stability and keeps the receptive field well conditioned across different aspect ratios. Such a shape also aligns naturally with the 3×3 subdivision used in the zoom-in operation, helping the grid cells remain reasonably balanced and allowing zoom-in decisions to focus more reliably on the correct directional region.

To improve robustness, we allow a small random shift of the enlarged window around c_t, chosen within the range that keeps the new region overlapping the original one. The expanded box is then clipped to the image domain:

$$I(s_{t+1}) = \mathrm{clip}(I(s_{t+1}), I_g),$$

ensuring valid boundaries at all times.

In our implementation, we adopt $\lambda = 1.5$ as a practical expansion factor. We observed that this value provides a reasonable compromise between gathering broader context and avoiding overly aggressive enlargement, and it performs reliably across datasets with varied resolutions and scene layouts.

Reachability of the search. We briefly argue that the combined effect of zoom-in and zoom-out, together with MCTS, is in principle sufficient to explore any target region within I_g. Let $R^\star \subset I_g$ be a target region and let $c^\star$ be its centre. Starting from the root, repeatedly applying zoom-in always yields a unique child region that contains $c^\star$ at each depth, because the 3×3 subdivision forms a disjoint partition at every level. This produces a nested sequence

$$I(s_0) \supset I(s_1) \supset \cdots \supset I(s_d)$$

whose diameter shrinks geometrically. For any prescribed tolerance $\varepsilon > 0$, we can therefore reach a depth d such that $I(s_d)$ is an arbitrarily small neighbourhood around $c^\star$.

From this finely localised state, applying zoom-out expands $I(s_d)$ approximately isotropically by factors of $\lambda > 1$, generating a sequence of enlarged regions whose scale increases geometrically. Because $R^\star$ has non-zero extent around $c^\star$, there exists a finite number of zoom-out steps m such that the resulting region $I(s_{d+m})$ attains a size comparable to $R^\star$ and overlaps it to any desired degree (up to the discretisation imposed by the image grid). In other words, by first zooming in to the centre of $R^\star$ and then zooming out to match its scale, the search can construct regions that approximate $R^\star$ with arbitrary accuracy.

Moreover, from any intermediate state s_t, a finite sequence of zoom-out operations returns the search to the global view I_g, from which an alternative zoom-in–zoom-out chain can be followed. In graph-theoretic terms, the state-transition graph induced by $\{\mathcal{T}_{\text{in}}, \mathcal{T}_{\text{out}}\}$ is therefore strongly connected: any state can reach any other through a finite sequence of actions.

Under the standard bounded-reward and non-zero exploration assumptions for UCT-based MCTS, such reachable high-reward paths are asymptotically discoverable: as the number of simulations increases, branches with larger expected reward are visited with increasing probability. Thus, in this idealised setting, the proposed action space and search procedure are theoretically capable of covering the entire spatial domain and locating any rewarding region with probability approaching one as the search budget grows.

6.2. Semantic Reward Design

To guide the search with linguistically grounded cues, GeoViS uses a semantic QA reward that verifies whether the current region $I(s_t)$ is consistent with the textual description. Remote sensing queries typically contain compositional semantics—namely the target object, its absolute geospatial position, and its relations to surrounding objects. These components provide complementary constraints: the target phrase specifies appearance, the absolute position anchors the region within the global scene, and the relational cues enforce consistency with other entities. Together, they enable the search procedure to reject semantically incompatible regions early and progressively converge toward the correct location.

Attribute Extraction Following the formulation in the main paper, we aim to convert each textual description T into its structured semantic representation

$$\hat{T} = \Phi(T) = \{o,\, p,\, r\},$$

where o denotes the target object phrase, p its absolute geospatial attribute, and r the set of relational references. To operationalize $\Phi(\cdot)$, we use an LLM (Qwen3-7B) to extract these components from free-form natural language. A fixed prompt instructs the model to output a JSON object with three fields: `target` (implementing o), `absolute_position` (implementing p), and `relations` (implementing r as pairs of reference objects and relation phrases).

For each image, the JSON outputs from all associated descriptions are merged into a single attribute record, yielding

an image-centric structured dataset that directly instantiates $\hat{T}$ and can be queried during the search process.

The following box shows the prompt used during attribute extraction. To illustrate the expected behavior, we also provide two representative input–output examples that demonstrate how the model should parse and format the extracted attributes.

Prompt at attribute extraction:

```
No think. Return only a JSON object with keys
    exactly: 'target', 'absolute_position', '
    relations'.
Rules:
   - target: copy the full descriptive NP of the
       main object (keep adjectives).
   - absolute_position: GLOBAL position that
       modifies the target ALONE. (e.g., 'in the
       middle', 'at the bottom', 'top left'). If
       none, ''.
     It must NOT contain another object or any
         relative markers ('of', 'to the * of', '
         than', 'between', 'near', 'next to', '
         beside', 'around').
   - relations: list of [other_object_phrase,
       relation_phrase]; for relative layouts
       copy the full other object NP and the
       exact relation phrase
     (e.g., 'on the upper right of'). Do NOT
         create a relation from the targets own
         modifiers;
     skip if the other object equals/overlaps the
          target phrase.
Return only JSON. No extra text or markdown.
```

Examples for attribute extraction:

```
Example 1
Input: A ship is similar in size to the ship in
    the middle
Output:
{
  "target": "ship",
  "absolute_position": "",
  "relations": [
    ["the ship on the lower right", "is similar in
        size to"]
  ]
}
```

```
Example 2
Input: The tiny rounded storage tank in the
    middle
Output:
{
  "target": "tiny rounded storage tank",
  "absolute_position": "in the middle",
  "relations": []
}
```

Computing the semantic reward. Given the extracted attributes $\{o, p, r\}$ for the image, we evaluate whether the current region is semantically compatible with them. For each attribute, we construct a binary QA query and provide VisualRAG with both the global image I_g and the current crop $I(s_t)$. This preserves global positional context while allowing the model to inspect fine-grained local evidence.

Concretely, we ask: (i) whether the target object o appears in the current region; (ii) whether the region's placement within the full scene agrees with the absolute position p; and (iii) for each relational pair $(\text{other}, \text{rel})$, whether the region satisfies the stated relation. VisualRAG answers each query with "yes" or "no". Let N be the total number of questions issued at this state and N_{yes} the number answered positively. The semantic reward is then

$$r_{\text{QA}} = \frac{N_{\text{yes}}}{N}.$$

This normalized score provides a soft, interpretable measure of semantic consistency, enabling the search to favor regions whose visual evidence aligns with the textual description.

6.3. IoU-Reward Design

Motivation. While the semantic QA reward evaluates whether a region is textually compatible with the description, it does not assess whether the region is geometrically plausible as an intermediate search result. The IoU-based reward provides this complementary geometric signal. It evaluates whether the predicted object box B_t, obtained by asking VisualRAG to predict the bounding box of the main object o within the current region $I(s_t)$, is well aligned with the central area of that region. Importantly, this reward serves as a coarse, class-level geometric consistency cue rather than a precise localization objective, which is performed later during the visual grounding stage by the VisualRAG model.

Definition. Following the main paper, we introduce a virtual central box B_c placed at the centre of the current region $I(s_t)$ and compute the geometric reward as

$$r_{\text{IoU}} = \text{IoU}(B_t, B_c).$$

This value lies in $[0, 1]$ and offers a soft assessment of whether the predicted box occupies a geometrically plausible and centrally consistent position within the region.

Why use a central proxy box? To understand what kind of local region serves as the most effective visual cue for Conditional Grounding, we conducted a controlled pre-experiment using ground-truth objects from the DIOR-RSVG dataset. We extracted multiple types of small local patches around each annotated object and used them as additional visual inputs to the grounding model. All variants share the same training protocol: the ViT encoder is frozen, the LLM is trained for the same number of epochs, and only

the choice of local patch varies. As a reference, we also report the performance of directly training Qwen2.5-VL-3B on the standard single-step grounding task without any local visual cue, denoted as the *baseline*. The four candidate patch designs are:

(a) target centred, region size = $1\times$ ground-truth box;
(b) target centred, region size = $2\times$ ground-truth box;
(c) target centred, region size = $3\times$ ground-truth box;
(d) target *not* centred, region size = $2\times$ ground-truth box (random offset).

Table 6 summarizes the grounding performance under these controlled settings. Among all tested configurations, setting (b), which uses a centred region with roughly twice the size of the ground-truth object, achieves the highest and most stable accuracy. These results indicate that a moderately enlarged, centre-focused local patch provides the most informative visual cue for Conditional Grounding, justifying the use of a central proxy region during the search process.

Table 6. Grounding accuracy (Pr@0.5) under different local patch configurations. Baseline denotes single-step grounding without any local patch.

Setting	Target centred?	Region size	Pr@0.5
baseline	-	-	71.2
(a)	yes	$1\times$ GT box	80.3
(b)	yes	$2\times$ GT box	**82.9**
(c)	yes	$3\times$ GT box	82.6
(d)	no	$2\times$ GT box	77.5

Discussion. These results indicate that the "centred + two-times scale" configuration represents an ideal upper bound for what a local visual cue can provide, offering the most reliable geometric signal under controlled conditions. In practice, search-time regions are rarely this perfectly aligned, and the model must operate on patches that vary in scale, offset, and visual completeness. The IoU-based reward is therefore designed to approximate this idealized behaviour: it encourages regions whose geometry is closest to the upper-bound configuration, guiding the search for patches that preserve the most informative spatial structure.

6.4. Additional Ablation Study

To further examine the effect of the maximum search depth and the maximum number of MCTS simulations, we conduct an ablation study on DIOR-RSVG using a set of controlled depth–simulation configurations. These two hyperparameters interact closely: deeper trees provide finer spatial resolution but expand exponentially, whereas too few simulations lead to under-exploration and noisy value estimates. A balanced combination is therefore required to achieve stable and efficient search.

Table 7. Ablation of search depth (Sim=10) and simulation count (Depth=5) on DIOR-RSVG using Pr@0.5.

Depth Comparison		**Simulation Comparison**	
Max depth	Pr@0.5	Max Sim.	Pr@0.5
3	77.7	5	77.8
5	79.8	10	79.8
7	78.5	15	80.0

Table 7 reports the results. When fixing the number of simulations to 10, shallow searches (Depth = 3) fail to provide sufficient granularity, while overly deep searches (Depth = 7) become unstable under the same simulation budget. With Depth = 5, the search is both sufficiently fine-grained and reliably explored.

When varying the simulation count under Depth = 5, we observe a similar trend. Increasing the number of simulations from 5 to 10 leads to a clear improvement, but raising it further to 15 provides only marginal gains while notably increasing inference cost. This diminishing return suggests that Depth = 5 and Sim = 10 offer the most favourable trade-off between accuracy and efficiency.

Overall, the configuration used in our main experiments (Depth = 5, Sim = 10) strikes a practical balance between spatial discrimination, exploration stability, and inference time, and we adopt it as the default setting throughout our evaluations.

7. Atomic Operation Dataset Construct

To train the VisualRAG model with fine-grained control over its four core capabilities (conditional grounding, zoom-in action selection, semantic QA reward evaluation, and IoU-based geometric assessment), we construct a set of atomic operation datasets derived systematically from the training splits of publicly available remote sensing benchmarks. These datasets transform the raw grounding annotations into task-specific supervision for each atomic operation, enabling VisualRAG to learn localized reasoning patterns, region-selection behaviours, semantic verification skills, and coarse geometric alignment signals in a unified manner. The following subsections describe how each dataset is generated and paired with the corresponding supervision objective. The final subsection summarizes the overall dataset sizes and data sources used during training.

7.1. Conditional Grounding Datasets Construction

During the construction of the **Conditional Grounding** dataset, we utilize "large image, description, and ground-truth box" tuples to create supervision signals. For the pos-

itive pairs, we crop a local image aligned with the ground-truth box (b_{gt}) to serve as the precise visual cue.

To better simulate the imperfect local observations produced during the multi-step search process, we also synthesize spatially perturbed regions. We explicitly select a perturbed region R_{pert} by applying spatial shifts and scaling based on b_{gt}, and crop the corresponding image content. If the Intersection over Union (IoU) between this perturbed region and the ground truth falls below a specific threshold, the cropped view is labelled as a perturbed region.

Algorithm 1 summarizes the complete data generation procedure.

Algorithm 1: Conditional Grounding Data Generation

Input : Annotation set $\mathcal{A}$, crop margin m, perturbation count K, IoU threshold τ_{iou}
Output: Training dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$;
for *each* $(I, T, b_{gt}) \in \mathcal{A}$ **do**
  $I_{\text{loc}} \leftarrow \text{Crop}(I, b_{\text{gt}}, m)$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I, I_{\text{loc}}, T), 1)\}$;
  for $k \leftarrow 1$ **to** K **do**
    $R_{\text{pert}} \leftarrow \text{GetPerturbedRegion}(b_{\text{gt}})$;
    if $IoU(b_{gt}, R_{pert}) < \tau_{iou}$ **then**
      $I_{\text{pert}} \leftarrow \text{Crop}(I, R_{\text{pert}}, m)$;
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I, I_{\text{pert}}, T), 0)\}$;
return $\mathcal{D}$;

Nomenclature:

- $\mathcal{A}$: Input annotation set containing (Image, Text, GT Box).
- I: The high-resolution source image (Global Context).
- I_{loc}: The cropped local image aligned with the target (Positive Visual Cue).
- I_{pert}: The cropped local image from a perturbed region (Negative Visual Cue).
- T: The natural language query describing the target.
- $\mathcal{D}$: The resulting dataset for training.
- R_{pert}: A spatially perturbed region generated based on b_{gt}.
- y: The binary supervision label (1 for aligned, 0 for perturbed).
- K: Number of perturbed regions to sample per positive annotation.
- m: Padding margin applied when cropping local images.
- τ_{iou}: IoU threshold used to define valid misalignment.
- $\text{Crop}(\cdot)$: Extracts the local image patch from I given a region and margin.
- $\text{GetPerturbedRegion}(\cdot)$: Generates a candidate region by applying random offsets/scaling to b_{gt}.

7.2. Zoom-in Action Dataset Construction

This phase converts the (Large Image, Description, Ground-truth Box) input into a 3×3 grid prediction task to supervise the **Zoom-in Action**. First, we apply a 3x3 grid to the entire large image, identify the cell k_{full} with the maximum overlap with b_{true}, and create a single-image sample: (Large Image + Description $\rightarrow k_{\text{full}}$). Next, we simulate a hierarchical search process: starting from the full region, we iteratively divide the current region into 3×3 blocks and identify the target cell k. We then crop the region corresponding to cell k from the original large image to create a local image I_{sub}, optionally expanding it randomly to retain geospatial context. This process generates a series of dual-image samples: (Large Image + Small Image + Description $\rightarrow k$). Finally, each annotation is expanded into one global sample and multiple chained local samples to train the zoom-in policy.

Algorithm 2 summarizes the complete data generation procedure.

Algorithm 2: Zoom-in Action Data Generation

Input : Annotation set $\mathcal{A}$, max depth $D_{\max}$, min area $A_{\min}$, zoom factor f_{zoom}
Output: Training dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$;
foreach $(I, T, b_{gt}) \in \mathcal{A}$ **do**
  $R_{\text{curr}} \leftarrow \text{GetFullExtent}(I)$;
  $y \leftarrow \text{GetGridIndex}(R_{\text{curr}}, b_{\text{gt}})$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\, ((I, T), y) \,\}$;
  for $d \leftarrow 1$ **to** $D_{\max}$ **do**
    if $Area(R_{curr}) < A_{\min}$ **then**
      Break;
    $y \leftarrow \text{GetGridIndex}(R_{\text{curr}}, b_{\text{gt}})$;
    $R_{\text{sub}} \leftarrow \text{GetSubRegion}(R_{\text{curr}}, y)$;
    $I_{\text{sub}} \leftarrow \text{Crop}(I, R_{\text{sub}})$;
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\, ((I, I_{\text{sub}}, T), y) \,\}$;
    $R_{\text{curr}} \leftarrow \text{Expand}(R_{\text{sub}}, f_{\text{zoom}})$;
return $\mathcal{D}$;

Nomenclature:

- T: The search query or structured instruction string.
- R_{curr}: The current active region of interest (initialized to full image).
- R_{sub}: The selected sub-region based on the grid action.
- y: The target class index (representing the optimal zoom-in action, e.g., 1-9).
- $D_{\max}$: Maximum depth allowed for the hierarchical search chain.
- $A_{\min}$: Minimum area threshold to terminate the zoom chain.
- f_{zoom}: Expansion factor used to include context around a selected sub-region.
- $\text{GetFullExtent}(\cdot)$: Returns the coordinate geometry of the entire image.
- $\text{GetGridIndex}(\cdot)$: Calculates which grid cell contains the target center (the supervision label).

- GetSubRegion($\cdot$): Computes the coordinates of the specific grid cell y within R_{curr}.
- Expand($\cdot$): Enlarges a region by factor f_{zoom} to maintain context overlap.
- Area($\cdot$): Computes the pixel area of a region.

7.3. QA Reward Dataset Construction

In this phase, we construct the **QA Reward** dataset to enable semantic verification. Based on the large remote sensing image L and its b_{true}, we crop multi-scale positive local images I_{pos} (containing the target) and negative local images I_{neg} (excluding the target). Concurrently, using the target, absolute_position, and relations attributes extracted in the previous stage, we automatically generate a set of attribute-based questions $\mathcal{Q}$ (regarding absolute and relative positions). Finally, for each local crop (I_{pos} or I_{neg}), we create a set of QA pairs, including a primary presence question ("Is there a `<target>`?") followed by all questions in $\mathcal{Q}$. The answers are "Yes" for pairs with I_{pos} and "No" for pairs with I_{neg}.

Algorithm 3 summarizes the complete data generation procedure.

Algorithm 3: QA Reward Data Generation

Input : Annotation set $\mathcal{A}$, Attribute set $\mathcal{M}$
Output: QA Dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$;
for *each* $(I, b_{gt}) \in \mathcal{A}$ *and corresponding* $M \in \mathcal{M}$ **do**
  $\mathcal{I}_{\text{pos}} \leftarrow \text{GenPosCrops}(I, b_{\text{gt}})$;
  $\mathcal{I}_{\text{neg}} \leftarrow \text{GenNegCrops}(I, b_{\text{gt}})$;
  $\mathcal{Q} \leftarrow \{\text{GenExistQ}(M.\text{tgt})\}$;
  if $M.abs \neq \emptyset$ **then**
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\text{GenAbsPosQ}(M.\text{tgt}, M.\text{abs})\}$;
  for *each* $(obj, rel) \in M.rel$ **do**
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\text{GenRelPosQ}(M.\text{tgt}, obj, rel)\}$;
  for *each* $I_{crop} \in \mathcal{I}_{pos}$ **do**
    for *each* $q \in \mathcal{Q}$ **do**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I, I_{\text{crop}}, q), \text{“Yes”})\}$;
  for *each* $I_{crop} \in \mathcal{I}_{neg}$ **do**
    for *each* $q \in \mathcal{Q}$ **do**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I, I_{\text{crop}}, q), \text{“No”})\}$;
return $\mathcal{D}$;

Nomenclature:

- $\mathcal{M}$: Structured attributes set containing target names and spatial relations.
- I_{crop}: A specific cropped local image from either the positive or negative set.
- q: A single generated question string.
- M.tgt: The name of the target object.
- M.abs: Absolute position attribute.
- M.rel: List of relative spatial relations .
- GenPosCrops($\cdot$)/GenNegCrops($\cdot$): Functions to sample crops based on overlap with b_{gt}.
- GenExistQ($\cdot$): Generates existence questions .
- GenAbsPosQ($\cdot$): Generates questions verifying absolute position.
- GenRelPosQ($\cdot$): Generates questions verifying relative position to other objects.

7.4. IoU Reward Dataset Construction

This phase builds the **IoU Reward** dataset based on the extracted (target phrase, bounding box) pairs to provide explicit spatial supervision. First, we normalize target phrases into semantic categories (e.g., keeping only nouns) and merge all bounding boxes for the same category within an image. Then, for each (image, category) pair, we sample local crops from the large image. We constrain some crops to contain one or more instances of the category (positive samples) and others to contain none (negative samples), maintaining a balanced distribution. Finally, we generate a uniform detection question for each crop ("Please provide the coordinates for the <category> targets"). The answer is a list of coordinates within the crop for positive samples, or an "is not present" message for negative samples.

Algorithm 4 summarizes the complete data generation procedure.

Algorithm 4: IoU Reward Data Generation

Input : Annotation set $\mathcal{A}$, Counts N_{pos} and N_{neg}
Output: IoU Reward Dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$;
for *each* $(I, \mathcal{O}_{list}) \in \mathcal{A}$ **do**
  $\mathcal{C} \leftarrow \text{GroupByCategory}(\mathcal{O}_{\text{list}})$;
  for *each* $(c, \mathcal{B}_c) \in \mathcal{C}$ **do**
    $T \leftarrow \text{GenPrompt}(c)$;
    for $i \leftarrow 1$ **to** N_{pos} **do**
      $I_{\text{crop}} \leftarrow \text{SamplePosCrop}(I, \mathcal{B}_c)$;
      $\mathcal{B}_{\text{vis}} \leftarrow \text{ProjectBoxes}(\mathcal{B}_c, I_{\text{crop}})$;
      if $\mathcal{B}_{vis} \neq \emptyset$ **then**
        $y \leftarrow \text{EncodeCoords}(\mathcal{B}_{\text{vis}})$;
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I_{\text{crop}}, T), y)\}$;
    for $i \leftarrow 1$ **to** N_{neg} **do**
      $I_{\text{crop}} \leftarrow \text{SampleNegCrop}(I, \mathcal{B}_c)$;
      $y \leftarrow$ “None”;
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{((I_{\text{crop}}, T), y)\}$;
return $\mathcal{D}$;

Nomenclature:

- $\mathcal{O}_{\text{list}}$: List of all annotated objects in image I.
- $\mathcal{C}$: Map grouping objects by category name $c \rightarrow$ box set $\mathcal{B}_c$.
- c: The category name of the target objects (e.g., "car").

Table 8. Statistics of the constructed training data. "Training Sets" denotes the raw number of samples in the source datasets. The subsequent columns detail the number of samples generated for each of our four atomic operations. (Unit: k samples)

Source Dataset	Training Sets	Cond. Grounding	Zoom-in Action	QA Reward	IoU Reward	Total
DIOR-RSVG [11]	27.0k	81.0k	58.0k	146.0k	45.6k	330.5k
RSVG-HR [6]	2.2k	6.5k	4.6k	18.7k	3.9k	33.6k
OPT-RSVG [8]	19.6k	58.7k	41.6k	68.8k	36.5k	205.7k
VRSBench [9]	36.3k	108.9k	75.6k	226.5k	59.3k	470.3k
GeoChat [4]	66.6k	198.9k	-	-	-	198.9k
Total	**151.7k**	**454.0k**	**179.7k**	**460.1k**	**145.2k**	**1239.0k**

- $\mathcal{B}_c$: The set of all ground truth boxes belonging to category c in the image.
- $\mathcal{B}_{\text{vis}}$: The subset of boxes visible within the current crop I_{crop}, projected to local coordinates.
- T: The generated text prompt or query for the specific category.
- y: The target label (coordinate string for positives, "None" for negatives).
- $N_{\text{pos}}, N_{\text{neg}}$: Number of positive/negative crops to sample per category.
- GroupByCategory$(\cdot)$: Function organizing objects into category-specific groups.
- GenPrompt$(\cdot)$: Generates the detection query based on category name.
- SamplePosCrop$(\cdot)$: Samples a crop that overlaps with at least one box in $\mathcal{B}_c$.
- SampleNegCrop$(\cdot)$: Samples a crop that does not overlap with any box in $\mathcal{B}_c$.
- ProjectBoxes$(\cdot)$: Converts global box coordinates to relative coordinates within the crop.
- EncodeCoords$(\cdot)$: Serializes the list of boxes into the output text format.

7.5. Data Sources and Statistics

The four atomic operation datasets described in the previous sections—Conditional Grounding, Zoom-in Action, QA Reward, and IoU Reward—are derived from annotations provided by established remote sensing benchmarks: DIOR-RSVG [11], RSVG-HR [6], OPT-RSVG [8], VRSBench [9], and GeoChat [4]. Notably, regarding GeoChat [4], we restricted our data construction exclusively to the Conditional Grounding task. This decision was driven by the fact that the aggregated data from the other sources for the remaining atomic operations already offers ample volume and diversity, thereby ensuring satisfactory generalization capability.

We follow the official data splits of all source datasets. For our atomic-operation supervision, we use only the training split and do not include any validation or test data. Detailed statistics of the datasets are presented in Table 8. Since these benchmarks are collected independently and we rigidly respect the split protocols, this ensures zero data leakage into the evaluation sets.

8. Generalization Ability

GeoViS is built upon a multimodal large language model rather than a task-specific model. Consequently, its role extends beyond locating objects in remote sensing scenes: our model also retains broad vision–language capabilities, such as open-ended question answering, global scene understanding, and instruction following. This distinguishes our framework from existing remote sensing visual grounding methods, which optimize only for a specific task but lack general-purpose reasoning abilities.

In addition to the geospatial atomic-operation data, we incorporate a substantial amount of general-domain visual instruction data to preserve the model's broad multimodal capability. Specifically, we sample approximately 400K diverse instruction-following examples from LLaVA-UHD v1 [3] and v2 [12] and interleave them with our training mixture. This strategy ensures that GeoViS retains strong general-purpose vision–language skills while simultaneously learning the specialized reasoning required for remote sensing visual grounding.

To verify that the model indeed preserves its general knowledge, we evaluate GeoViS on three widely used multimodal benchmarks: MMBench-V1.1 [10], SEED-Bench [7], and MMStar [2]. The results in Table 9 show that GeoViS remains highly comparable to the base model Qwen2.5-VL-3B-Instruct [1], with only marginal fluctuations on MMBench-V1.1 and SEED-Bench and a slight improvement on MMStar. This demonstrates that our geospatial fine-tuning does not compromise the model's general-purpose capabilities and may even enhance robustness in complex reasoning tasks.

9. Limitation and Future Work

Although GeoViS substantially improves localization accuracy for small objects in large-scale remote sensing imagery, the framework introduces an inherent efficiency limitation. Unlike conventional single-step MLLM inference,

Table 9. Comparison with the baseline model on general multimodal benchmarks. The metric used is Accuracy (Acc).

Benchmark	Qwen2.5-VL-3B [1]	Ours
MMBench-V1.1 [10]	76.9	76.3
SEED-Bench [7]	73.8	73.3
MMStar [2]	55.9	56.1

our method relies on a multi-step search procedure, which increases inference latency. To reduce this overhead, we adopt a lightweight 3B backbone and deploy inference through the vLLM framework [5], which significantly reduces per-step runtime.

Despite this additional cost, iterative visual search is essential in scenarios where objects are extremely small relative to the global scene and cannot be reliably localized with a single global encoding. In such cases, the accuracy improvements obtained through progressive search justify the modest extra runtime and make the framework well-suited to high-resolution, wide-area remote sensing analysis.

In future work, we plan to explore more efficient search strategies to further reduce inference cost. We also seek to extend GeoViS toward broader operational settings and investigate its integration with large-scale remote sensing workflows, enabling even more flexible and scalable geospatial reasoning capabilities.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[3] Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pages 390–406. Springer, 2024.

[4] Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27831–27840, 2024.

[5] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[6] Meng Lan, Fu Rong, Hongzan Jiao, Zhi Gao, and Lefei Zhang. Language query-based transformer with multiscale cross-modal alignment for visual grounding on remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–13, 2024.

[7] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13310, 2024.

[8] Ke Li, Di Wang, Haojie Xu, Haodi Zhong, and Cong Wang. Language-guided progressive attention for visual grounding in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–13, 2024.

[9] Xiang Li, Jian Ding, and Mohamed Elhoseiny. Vrsbench: A versatile vision-language benchmark dataset for remote sensing image understanding. *Advances in Neural Information Processing Systems*, 37:3229–3242, 2024.

[10] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, 2024.

[11] Yang Zhan, Zhitong Xiong, and Yuan Yuan. Rsvg: Exploring data and models for visual grounding on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–13, 2023.

[12] Yipeng Zhang, Yifan Liu, Zonghao Guo, Yidan Zhang, Xuesong Yang, Xiaoying Zhang, Chi Chen, Jun Song, Bo Zheng, Yuan Yao, et al. Llava-uhd v2: an mllm integrating high-resolution semantic pyramid via hierarchical window transformer. *arXiv preprint arXiv:2412.13871*, 2024.