LISAT: Language-Instructed Segmentation Assistant for Satellite Imagery

Jerome Quenum* **Wen-Han Hsieh*** **Tsung-Han Wu** **Ritwik Gupta**
Trevor Darrell **David M. Chan**

University of California, Berkeley

Abstract

Segmentation models can recognize a pre-defined set of objects in images. However, segmentation models capable of "reasoning" over complex user queries that implicitly refer to multiple objects of interest remain underexplored, especially in the geospatial domain. Recent advances in "reasoning segmentation"—generating segmentation masks from complex, implicit query text—demonstrate the potential of vision-language models (VLMs) to reason across an open domain of objects. Yet, our experiments reveal that these models struggle when applied to the unique challenges of remote-sensing imagery. To address this gap, we introduce a new dataset which consists of: GRES, a curated geospatial reasoning-segmentation dataset with 27,615 annotations across 9,205 images, and PreGRES, a collection of existing datasets to make up a large-scale multimodal pretraining corpus with over 1M question-answer pairs across 119,279 images. We propose an initial benchmark model, LISAT, a VLM for geospatial analysis that can describe complex remote-sensing scenes, answer detailed queries, and segment objects based on natural-language prompts. LISAT establishes a strong initial geospatial benchmark, outperforming prior foundation models such as RS-GPT4V by 10.04% (BLEU-4) on visual description tasks and surpassing open-domain models on geospatial reasoning segmentation by 143.36% (gIoU). Our model, dataset, and code are available on our project page.

1 Introduction

Segmentation models for remote-sensing have been a staple of geospatial analysis, supporting applications ranging from disaster response, environmental monitoring, and more [56, 47]. These models typically operate within rigid boundaries but struggle to adapt to real-world scenarios in which the ability to segment regions based on flexible, user-defined queries—tasks often referred to as reasoning segmentation—is paramount [24]. For instance, a query such as “identify flood-prone urban areas” or “which regions have observed urban expansion” demands that segmentation models move beyond static object recognition and into contextual, task-specific reasoning. However, progress in this area has been limited, primarily due to the absence of datasets that pair natural language reasoning prompts with pixel-level segmentation in remote-sensing imagery.

Adapting vision-language models to remote-sensing is challenging due to the unique characteristics of satellite imagery—high variability in object scale, spatial context, and clutter [46]. These challenges are compounded by the lack of high-quality datasets that pair natural language queries with fine-grained spatial annotations. While large-scale multimodal datasets exist for natural images, remote-sensing lacks such resources, making it difficult for general-purpose or natural-image-trained models

*Equal contribution; corresponding author: jquenum@berkeley.edu

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

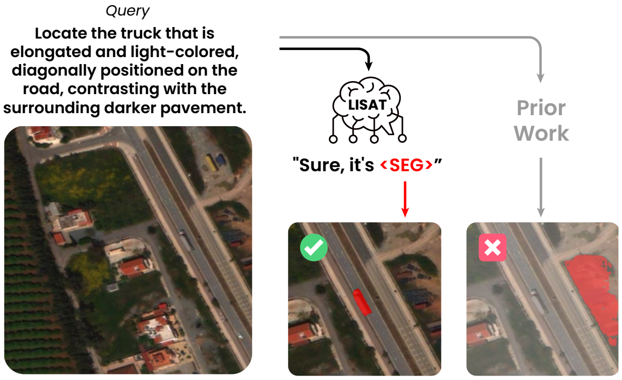

Query
Locate the truck that is elongated and light-colored, diagonally positioned on the road, contrasting with the surrounding darker pavement.
LISAT
"Sure, it's <SEG>"
Prior Work

Figure 1.1: Existing models struggle to generate accurate segmentation masks for complex natural language queries in remote-sensing imagery. LISAT, our open-source, open-data, foundation model for geospatial reasoning segmentation trained on GRES, our new semi-synthetic dataset for remote-sensing reasoning segmentation, helps to bridge the gap between SOTA reasoning segmentation models and remote-sensing domains.

to perform reasoning segmentation effectively [61, 69]. Recent geospatial foundation models excel at tasks like captioning and VQA but are limited to textual outputs and cannot generate segmentation masks. Existing vision-language models that support segmentation [24, 45, 19] struggle in this domain due to the scarcity of spatially grounded language supervision in remote-sensing.

We address these challenges by introducing a new dataset designed to enable multimodal learning for geospatial reasoning segmentation. First, it contains PreGRES, which aggregates over 1 million question-answer pairs from diverse remote-sensing datasets along with 119,279 images and serves as a large-scale dataset for pretraining multimodal large language models on geospatial content. Then, GRES (Geospatial Reasoning Segmentation dataset), which provides a fine-grained supervision through 27,615 pixel-level annotations paired with natural language reasoning segmentation queries across 9,205 satellite images. These two components of the dataset form a complementary training pipeline where models are first pretrained on PreGRES to acquire broad geospatial reasoning capabilities, then fine-tuned on GRES for dense, spatially grounded segmentation.

To demonstrate the utility of this dataset, we train an initial benchmark model LISAT (**L**anguage **I**nstruction **S**egmentation **A**ssistant for Sa**t**ellite Images), an open-source and open-data vision-language model that bridges the gap between reasoning segmentation and remote-sensing foundation models. LISAT achieves significant performance gains over state-of-the-art geospatial and open-domain models. Specifically, LISAT outperforms existing geospatial foundation models, such as RS-GPT4V, by over 10.04% on BLEU-4 on remote-sensing visual description tasks and outperforms state-of-the-art open-domain models on remote-sensing reasoning segmentation by 143.36% on gIoU.

2 Related Work

Semantic segmentation is a core task in remote sensing, supporting applications in urban planning, economic assessment, agriculture, resource management, and environmental protection [56, 15, 16, 7, 14, 47]. However, traditional models are often task-specific and require extensive fine-tuning to adapt across use cases. The rise of vision-language models (VLMs) [42, 32] has driven interest in models that can handle arbitrary natural language queries alongside images. General-purpose models like GPT-4 [1] and LLaVA [31] enable tasks such as visual description and question answering, but they do not perform well in geospatial reasoning segmentation. Developing effective models in this context requires access to large, high-quality datasets that pair natural language with detailed, spatially-grounded annotations, which remains a significant gap in the field that we seek to close.

2.1 Remote-Sensing Datasets for Multimodal Learning

Semantic segmentation in remote sensing has long been constrained by a lack of large-scale datasets that combine fine-grained spatial annotations with multimodal supervision. Well-established benchmarks such as DeepGlobe [11, 51, 25] have advanced geospatial vision tasks by providing imagery annotated for classification, detection, and basic segmentation. However, these datasets do not support the kind of complex, query-driven interaction that modern vision-language models require. To support multimodal tasks, several remote sensing datasets have emerged at the image or region level. Datasets like `Sydney-Captions` [41], `RSICD` [35], `NWPU-Captions` [41], `RSITMD` [65], and `UCM-Captions` [41] enable captioning and image-text retrieval. While useful for high-level semantic understanding, they are individually small in scale, and combining them will help improve text generation. More recent efforts, such as `VRSBench` [27] and `GeoChatInstruct` [23] have expanded multimodal learning to region-level tasks like grounded image captioning, region-specific question answering, and visual grounding. These datasets are built on existing remote sensing datasets (e.g., [66, 54]) and use rule-based or GPT-based methods to automatically generate textual descriptions for objects or regions within images. While they support region-level reasoning through bounding box annotations, they do not include pixel-level ground truth, which is necessary for supervised training in segmentation tasks. Datasets that do provide segmentation supervision, such as FloodPrompt [26] and RefSegRS [64], are often domain-specific or limited in scale, with RefSegRS offering only 4,420 images. The dataset we introduce, PreGRES and GRES, are specifically developed to address these limitations. Together, they provide a unified pipeline of detailed spatial annotations paired with natural language, enabling the training of models that can both understand and segment remote sensing imagery.

2.2 Reasoning Segmentation

Reasoning segmentation refers to generating segmentation masks from complex or implicit natural language queries (Figure 1.1). Two main approaches have emerged for this task. The first is the embedding-as-mask paradigm, introduced by LISA [24], which decodes a `[SEG]` token into a segmentation mask using a SAM-based decoder [22]. PixelLM [45] improved multi-target differentiation with a lightweight pixel decoder and segmentation codebook, while GSVA [58] introduced a `[REJ]` token to handle ambiguous or absent targets. GLaMM [44] addressed granularity through focused training data. The second approach represents visual outputs directly in language, as in Shikra [6], Kosmos-2 [40], and others [71, 62, 31, 32], which align vision and language to predict textual descriptions or coordinates. Despite progress, existing methods underperform in remote sensing due to challenges like varying spatial resolution, fine-grained class differences, and a lack of domain-specific datasets (Table 5.4). Our proposed model, LISAT, extends the embedding-as-mask paradigm to top-down remote-sensing data.

2.3 Geospatial Foundation Models

Recent geospatial foundation models have extended the foundation model paradigm to remote sensing, supporting tasks such as captioning, VQA, and object detection. EarthGPT [69] unifies multi-sensor RS tasks using a large-scale multimodal dataset (see subsection 3.1). TEOChat [21] introduces temporal reasoning for change detection and damage assessment, while GeoChat [23] enables region-specific dialogue and visual grounding. SkyEyeGPT [67] and RS-GPT4V [61] focus on instruction-following and complex scene understanding. Despite this progress, existing models primarily produce natural language outputs and lack support for spatial outputs like segmentation masks. Our proposed model, LISAT, addresses this limitation by directly generating segmentation masks in response to reasoning queries.

3 Geospatial Reasoning Segmentation Dataset

The development of vision-language models (VLMs) for remote sensing has been hindered by the lack of high-quality remote sensing imagery paired with natural language data, a key challenge outlined in our introduction. Unlike natural image datasets, remote-sensing data require fine-grained, context-aware segmentation that accounts for extreme variations in scale, subtle object differences, and the ability to reason across complex spatial relationships. To help alleviate this need, we introduce the Geospatial Reasoning Segmentation Dataset (GRES), a collection of vision and language data designed around remote-sensing applications. **GRES** consists of two core components: **PreGRES**, a dataset consisting of over 1M remote-sensing specific visual instruction-tuning Q/A pairs for pretraining geospatial models, and **GRES**, a semisynthetic dataset specialized for reasoning segmentation

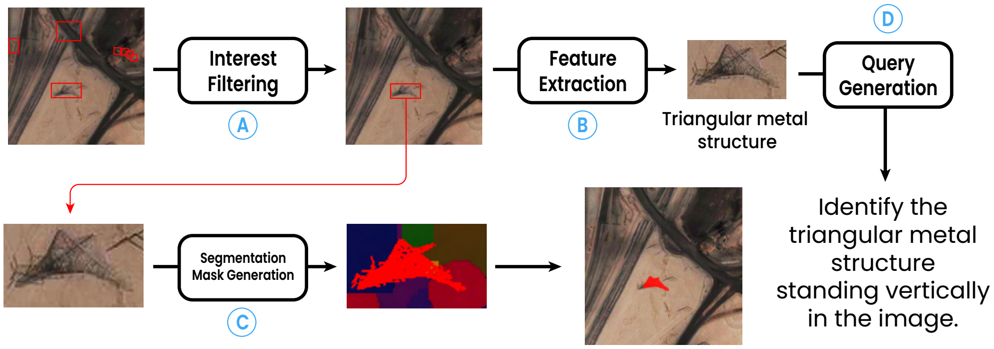

Interest Filtering
A
Feature Extraction
B
Triangular metal structure
D
Query Generation
Identify the triangular metal structure standing vertically in the image.
Segmentation Mask Generation
C

Figure 3.2: To generate synthetic data, we start with a seed detection dataset (`xView`). We then filter detections for those that are both visually interesting and highly distinguishable (A). For those detection, we then generate a natural language description (B), and a pixel-wise segmentation mask (C). Finally, the natural language description is used to generate a localization query (D). Together, the segmentation mask and the query form a ground-truth pair for the LISAT reasoning segmentation fine-tuning.

of remote-sensing data. With this structure of **GRES**, we enable LISAT to overcome both data scarcity and the domain transfer limitations faced by general-purpose models.

3.1 PreGRES

PreGRES is a large-scale structured collection of existing smaller-scale geospatial datasets designed for fine-tuning vision-language models in remote sensing applications. It integrates multiple sources, each contributing to different aspects of geospatial data understanding. The datasets within GRES provide coverage across image captioning, visual question answering, and visual grounding tasks:

1. **Image Captioning:** `NWPU-Captions` [9], `RSICD` [35], `RSITMD` [65], `Sydney-Captions` [41], and `UCM-Captions` [41]. Each contributes paired image-text data, and contains long-form descriptions of top-down imagery across different geospatial environments, increasing the diversity of language supervision.
2. **Visual Question Answering (VQA):** `RSVQA_LR` [33], `RSVQA_HR` [33], `FloodNet` [43], and `RSIVQA` [73]. Each of these datasets consists of structured question-answer pairs and supports reasoning over aerial and satellite images, covering tasks such as object identification, scene understanding, and disaster assessment.
3. **Visual Grounding / Region-Level Captioning:** `DIOR-RSVG` [66] provides paired text-image data for object localization and spatial reference resolution, and `NWPU-RESISC45` [8] supplies scene classification labels.

Overall, PreGRES consists of 119,279 images and 1,204,993 question-answer pairs and is used in the first-stage pre-training of the LISAT model enabling general-purpose geospatial question-answering in the final LISAT model. For more details on dataset composition, see Table D.11.

3.2 GRES

GRES is a semi-synthetic dataset designed explicitly for geospatial reasoning segmentation. Each sample in GRES consists of an image, a natural language query referring to a single object in that image, and a pixel-level segmentation mask (See Figure 3.2 for an example of a GRES query/image pair). This task allows us to train the LISAT model to correctly localize images at a pixel level within the scene, even in the case of multiple objects requiring disambiguation.

To build the dataset, we begin with a subset of the `xView` dataset [25] consisting of $26,541$ high-resolution satellite images spanning approximately $1,400$ square kilometers, covering more than 60 classes. `xView` consists of paired images and object detections within the images in bounding box form. To convert `xView` images/annotations to GRES annotations/images, we follow the process overviewed in Figure 3.2.

Given an input image of size 512 × 512, we divide it into 4 quadrants, where the top-left quadrant is defined by $0 \leq x \leq 255, 0 \leq y \leq 255$; the top-right quadrant is defined by $256 \leq x \leq 511, 0 \leq y \leq 255$; the bottom-left quadrant is defined by $0 \leq x \leq 255, 256 \leq y \leq 511$; the bottom-right quadrant is defined by $256 \leq x \leq 511, 256 \leq y \leq 511$.

In the first part of the pipeline, we need to generate a "disambiguating query" that selects for a single object within the scene from the large set of objects. To do so, we first filter the scenes for two key objectives: (1) uniqueness (i.e. can objects be easily disambiguated with a natural language query), and (2) interest (i.e. are the objects visually interesting) (Figure 3.2, A). An object is considered "unique" in an image if it is one of less than 2 detections of its class in its respective quadrant, and an object is considered "visually interesting" if it belongs to a class appearing in less than 50% of the overall subset of `xView` detections. Comprehensive statistics of object categories after filtering are available in Table C.7. To ensure a balanced evaluation, our dataset includes queries with and without explicit spatial references, each with a 50% probability.

After the filtering stage, we convert the object detection to a query using a set of structured queries to a large vision and language model trained on natural images (in our case, GPT-4v [1], Figure 3.2, B). In the first prompting stage, we ask the VLM to identify unique characteristics of the class within the bounding box by asking the model to ``Find visual features (color, shape, size, etc.) that to help find or segment {class_name} in the image.''. We then ask the VLM to come up with a sentence describing the object in the bounding box within the scene using the collected unique characteristics (See the full prompt in Appendix C.1.1). Given these features, we prompt the VLM again with the full image, along with other detections in the image and the position of the bounding box to produce a query(see the full prompt in Appendix C.1.2, Figure 3.2, D).

In the second part of the pipeline (Figure 3.2, C), we need to generate the pixel-based mask from the bounding box. To do this, we leverage a GeoSAM model [48] with a custom high-resolution inference configuration (128 points per side, 0.95 prediction IoU threshold, and 0.95 stability score with an 80-pixel minimum mask region area) to produce a part-wise segmentation of each bounding box. We then add any sub-parts that cover more than $80px$ of the underlying bounding box to the final pixel mask.

We then asked the VLM to rephrase each query two separate ways which added to the initially generated query gives us 3 queries per image. This pipeline overall results in a dataset consisting of 9,205 images and 27,615 natural language queries/answers within those images. From this dataset, we generate train, test, and validation splits consisting of 7,205, 1,500, and 500 images respectively.

4 Training LISAT for Geospatial Reasoning Segmentation

Inspired by LISA [24], LISAT integrates a multimodal large language model (LLM) with a segmentation model. The multimodal LLM processes both textual and visual inputs, leveraging datasets that contain image-text pairs for instruction-following and reasoning [32] while the segmentation model uses a dataset designed for high-quality mask generation [22]. An overview of the architecture is given in Figure 4.3.

4.1 Geospatial Multimodal Language Models

While LISA [24] leverages a pre-trained LLaVA [31, 32] model as a vision and language backbone, we found that leveraging LLaVA alone was insufficient to capture the range of queries and visual variance in remote-sensing applications. To solve this problem, in the first stage of our training process we trained a remote-sensing specific multimodal large language model to serve as the base MLLM for the segmentation backbone. Our architecture generally follows LLaVA [31, 32] with several modifications for remote-sensing applications.

For the base language model, we leverage the default Vicuna-7B [10] without additional pre-training or fine-tuning for remote sensing data to embed a text query $\mathbf{X}_l$. For the visual backbone, LISAT adopts the Remote-CLIP ViT-L/14 encoder [30] to extract visual features from an input image $\mathbf{X}_v$. To align visual representations with the language model's word embedding space, we use a simple linear projection matrix to produce a sequence of visual tokens that match the dimensionality of the word embeddings in the language model. A pre-trained Vicuna base model combined with the vision

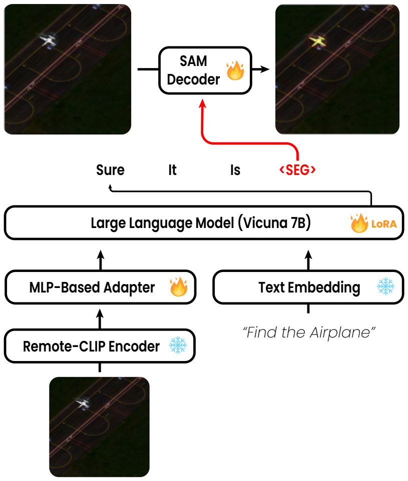


Figure 4.3: LISAT integrates a geospatial multimodal large language model (MLLM) with a segmentation decoder to enable reasoning-based segmentation. LISAT first pre-trains a Remote-CLIP-based MLLM on PreGRES before fine-tuning on GRES. We then expand the LMM vocabulary with a segmentation token (`<SEG>`), whose final-layer embedding is projected into the SAM segmentation query space and combined with image features to produce a segmentation mask.

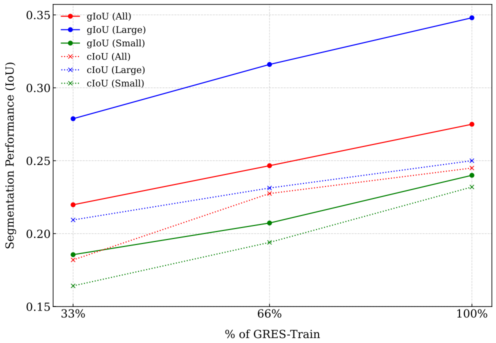


Figure 4.4: Scaling behavior of LISAT on the GRES dataset. While adding additional data is helpful, even with $7K$ *training* images (the full GRES dataset), we observe the beginning of a plateau in performance, particularly on cIoU scores. This suggests that more data alone may not be helpful, and instead, we may need additional data variance outside the `xView` classes.

Segmentation Model	cIoU	gIoU
GeoSAM [48]	$0.220_{\pm 0.019}$	$0.238_{\pm 0.007}$
SAM [22]	$\mathbf{0.245}_{\pm 0.023}$	$\mathbf{0.275}_{\pm 0.009}$

Table 4.1: Comparison of LISAT's performance using GeoSAM vs. SAM for segmentation on the `All` dataset configuration.

encoder is further pre-trained on PreGRES (see section 5) with LoRA [18] prior to being trained on GRES. We refer to this pre-trained variant as LISAT$_{\text{PRE}}$.

4.2 Preliminaries

Existing multimodal LLMs for remote sensing, such as RS-GPT4V [61] and EarthGPT [69], support images and text as input but output only text. To produce segmentation masks, LISAT leverages the "embedding-as-a-mask" paradigm introduced by LISA [24], and expands the LLM vocabulary with a new token, `<SEG>`, which represents segmentation requests. When the model produces an output containing the `<SEG>` token, we extract the final layer embedding of that token, and project it via an MLP layer to the query space of a SAM-based segmentation decoder [22]. The segmentation decoder combines the query-projected final embedding and a set of visual features extracted from the base image to produce a final segmentation mask $\hat{\mathbf{M}}$.

4.3 Training Objectives

LISAT is trained end-to-end with a loss function that combines text generation and segmentation objectives. The total loss $\mathcal{L}$ is the weighted sum of two components:

$$\mathcal{L} = \lambda_{txt}\mathcal{L}_{txt} + \lambda_{mask}\mathcal{L}_{mask}. \tag{4.1}$$

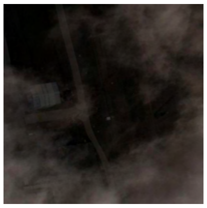
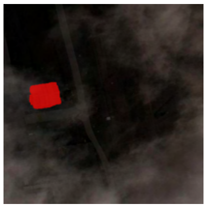
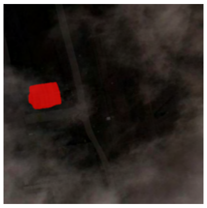
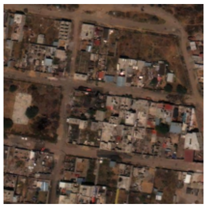
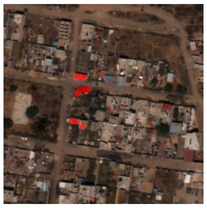
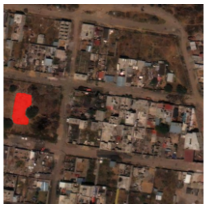
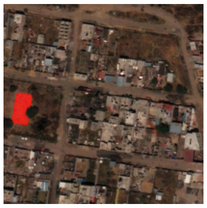
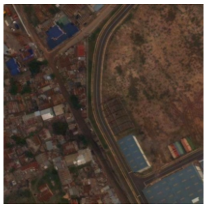
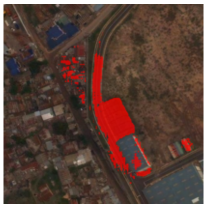
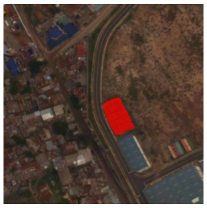
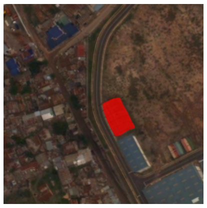
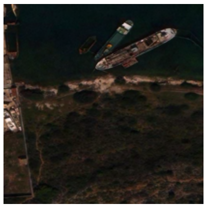
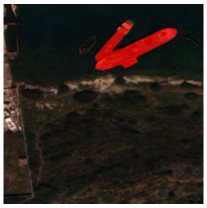
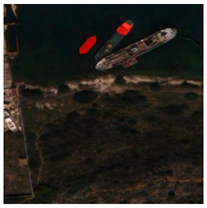
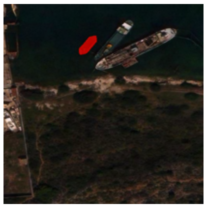
where the text generation loss $\mathcal{L}_{txt}$ is an autoregressive cross-entropy loss:

$$\mathcal{L}_{txt} = \mathbf{CE}(\hat{\mathbf{y}}_{txt}, \mathbf{y}_{txt}). \tag{4.2}$$

and the segmentation loss $\mathcal{L}_{mask}$ consists of a per-pixel binary cross-entropy (BCE) loss and a DICE loss, weighted by λ_{bce} and λ_{dice}:

$$\mathcal{L}_{mask} = \lambda_{bce}\mathbf{BCE}(\hat{\mathbf{M}}, \mathbf{M}) + \lambda_{dice}\mathbf{DICE}(\hat{\mathbf{M}}, \mathbf{M}). \tag{4.3}$$

Table 5.2: Qualitative examples of the segmentations generated by LISAT on the GRES dataset.

Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Locate the building with a large rectangular structure, dark roof, and symmetrical window patterns.				
Identify the facility in the center-left of the image.				
Identify the damaged building in the center of the image.				
Failure Case: Locate the dark, elongated rectangular shape with a red outline against the dark background to identify the barge.				

5 Experimental Results

Implementation Details: LISAT and LISAT$_{\text{PRE}}$ are trained on eight DGX A100 80GB GPUs. In the first stage, we pretrain LISAT$_{\text{PRE}}$ (context length = 2048) using LoRA [18] for 1 epoch on PreGRES (described in subsection 3.1) with next-token prediction cross-entropy loss. We employ the AdamW optimizer [34] with a learning rate of $3e^{-4}$ and a cosine-decay learning rate scheduler, setting the batch size to 2 and gradient accumulation steps to 6.

In the second stage, we train LISAT using GRES, as well as two traditional natural image referring segmentation datasets, FP-Ref-COCO [57] and ReasonSeg [24]. LoRA is applied to LISAT$_{\text{PRE}}$, while the SAM decoder undergoes full fine-tuning. The learning rate is set to $3e^{-4}$, with all other configurations remaining the same. For the loss function, we empirically found that setting the weight for text generation loss (λ_{txt}) and mask loss (λ_{mask}) to 1.0, while the binary cross-entropy loss (BCE) (λ_{bce}) and Dice loss (λ_{dice}) are assigned weights of 2.0 and 0.5, respectively performs better as suggested by [57, 24, 55]. The total training time was approximately 12 hours on eight DGX A100 80GB GPUs.

Evaluation Protocol: We use the GRES test set to evaluate segmentation performance. We focus on two subsets of the GRES test set, Small and Large, to evaluate performance on small and large objects, respectively. We define a threshold of 500 pixels^2 and categorize any object in the test set that covers an area less than the threshold to be `Small` and bigger to be `Large`. We evaluate segmentation performance using generalized Intersection-over-Union (gIoU) and cumulative Intersection-over-Union (cIoU) [24]. To evaluate the performance of our approach on traditional vision and language tasks, we use several existing datasets, including NWPU-Captions [9], UCM-Captions [41], Sydney-Captions [41], and RSICD [35]. Following prior work, we report standard evaluation metrics: BLEU [39], CIDEr [52], and SPICE [2].

Vision Encoder	LLM	BLEU-4	CIDEr	SPICE
CLIP	Llama 2	69.03	328.82	52.21
CLIP336	Llama 2	66.97	324.61	50.46
SAT-CLIP	Llama 2	8.82	30.41	8.15
Geo-CLIP	Llama 2	12.77	44.64	11.67
RemoteCLIP	Llama 2	68.31	330.94	52.17
CLIP	Vicuna	66.68	329.32	52.00
CLIP336	Vicuna	68.28	324.89	51.55
SAT-CLIP	Vicuna	16.87	63.92	15.08
Geo-CLIP	Vicuna	24.56	109.20	21.15
RemoteCLIP	Vicuna	**72.34**	**355.32**	**54.15**

Table 5.3: Ablations of the base language model and visual feature extractor for LISAT$_{\text{PRE}}$ on the UCM-Captions dataset. We found that the combination of RemoteCLIP and Vicuna gives the best performance. This demonstrates the importance of carefully choosing vision encoders and language models for improved multimodal understanding, as doing so can significantly boost captioning metrics compared to generic CLIP variants.

Model	Obj. Size	cIoU	gIoU
LISA-7B	All	$0.122_{\pm 0.014}$	$0.113_{\pm 0.007}$
	Small	$0.104_{\pm 0.022}$	$0.062_{\pm 0.008}$
	Large	$0.157_{\pm 0.017}$	$0.222_{\pm 0.013}$
LISA-13B (llama2)	All	$0.122_{\pm 0.014}$	$0.139_{\pm 0.006}$
	Small	$0.106_{\pm 0.016}$	$0.089_{\pm 0.007}$
	Large	$0.148_{\pm 0.018}$	$0.244_{\pm 0.019}$
PixelLM-7B	All	$0.101_{\pm 0.011}$	$0.142_{\pm 0.006}$
	Small	$0.069_{\pm 0.009}$	$0.094_{\pm 0.006}$
	Large	$0.142_{\pm 0.019}$	$0.243_{\pm 0.014}$
PixelLM-13B	All	$0.145_{\pm 0.013}$	$0.162_{\pm 0.008}$
	Small	$0.102_{\pm 0.015}$	$0.106_{\pm 0.008}$
	Large	$0.204_{\pm 0.028}$	$0.277_{\pm 0.014}$
LISAT (Ours)	All	$\mathbf{0.245}_{\pm 0.023}$	$\mathbf{0.275}_{\pm 0.009}$
	Small	$\mathbf{0.232}_{\pm 0.024}$	$\mathbf{0.240}_{\pm 0.009}$
	Large	$\mathbf{0.250}_{\pm 0.029}$	$\mathbf{0.348}_{\pm 0.015}$

Table 5.4: Performance of LISAT against LISA-7B-v1, LISA-13B-Llama2-v1, PixelLM-7B and PixelLM-13B on GRES across different object sizes. LISAT-7B consistently outperforms the baseline models, particularly in the Small object category.

5.1 Segmentation

Table 5.4 compares LISAT with LISA-7B-v1 and LISA-13B-Llama2-v1 [24] across different dataset configurations (All, Small, Large). LISAT consistently and significantly outperforms both natural-image trained referring segmentation models. Notably, for smaller objects, LISAT has larger relative gains compared to large models, suggesting that LISAT is more effective for capturing fine-grained spatial details, which is important for applications involving dense scenes or small-scale features in remote sensing imagery.

Some qualitative examples are given in Table 5.2. The first three rows represent success cases, where LISAT correctly identifies and localizes objects based on the queries. In the first, LISAT correctly segments the building against a noisy background, and when many of the ground features match the visual features of the target object. In the second and third, LISAT correctly identifies the key object of interest, ignoring other potential distractor objects. In the failure case, LISAT fails to correctly identify the barge alone from the two ships, likely due to the color patterns on the first ship, but still manages to outperform LISA, which only focuses on the larger ship objects.

Figure 4.4 demonstrates the influence of training dataset size on LISAT 's performance. With an increasing number of training images, LISAT demonstrates notable improvements in both cIoU and gIoU scores. These results indicate that LISAT benefits from larger training datasets thereby exhibiting some good scaling properties, as its segmentation performance improves with more data, particularly for small objects.

Table 4.1 compares LISAT's performance using GeoSAM and SAM as base segmentation models on the `All` dataset. While both models yield competitive results, SAM achieves slightly higher cIoU (0.245) and gIoU (0.275) than GeoSAM. This suggests that despite being designed for geospatial tasks, GeoSAM alone without specific language-aligned fine-tuning may be limited by training-specific biases, whereas SAM's broader training on diverse natural images enables more adaptable feature extraction, leading to improved segmentation performance.

5.2 Captioning and Question-Answering

On the UCM-Captions dataset (Table 5.5), LISAT$_{\text{PRE}}$ achieves the highest BLEU-4 (72.34) and CIDEr (355.32) scores, surpassing previous geospatial models such as RS-GPT4V [61] and post-processing methods [20], as well as general-purpose vision-language models such as LLaVA-v1.5 and LLaVA-v1.6 [32, 31]. For NWPU-Captions (Table 5.6), LISAT$_{\text{PRE}}$ achieves the highest BLEU-4 score and matches the best SPICE performance, outperforming prior geospatial captioning models such as MLCA-Net [9] and multimodal attention-based methods [35]. General-purpose vision-language

Method	BLEU-4	CIDEr
SAA [33]	64.77	294.51
SD-RSIC [49]	53.80	213.20
RTRMN (semantic) [53]	35.87	180.25
RTRMN (statistical) [53]	63.93	312.70
SVM-D BOW [17]	51.95	271.42
SVM-D CONC [17]	59.42	292.28
Post-processing [20]	62.62	309.64
LLaVA-v1.5-7b [32]	5.54	32.67
LLaVA-v1.6-7b [31]	5.44	23.86
RS-GPT4V [61]	65.74	333.23
LISA-7B (baseline)	0.00	0.00
LISA-7B (fine-tuned on GRES)	8.73	59.96
LISAt$_{\text{PRE}}$ (Ours)	**72.34**	**355.32**

Table 5.5: Comparison of captioning performance on the UCM-Captions dataset. Results are reported for BLEU-4 and CIDEr metrics.

Method	BLEU-4	SPICE
CSMLF [50]	47.1	26.5
Multimodal [41]	45.5	27.6
Attention (hard) [35]	46.4	28.4
FC-Att [70]	46.9	28.3
MLCA-Net [9]	47.8	28.5
LLaVA-v1.5-7b [32]	4.8	11.1
LLaVA-v1.6-7b [31]	2.9	8.7
EarthGPT[69]	65.5	**32.2**
LISA-7B (baseline)	0.00	0.00
LISA-7B (fine-tuned on GRES)	39.9	19.52
LISAt$_{\text{PRE}}$ (Ours)	**65.8**	**32.2**

Table 5.6: Comparison of captioning performance on the NWPU-Captions dataset. Results are reported for BLEU-4 and SPICE metrics.

models (LLaVA-v1.5 and LLaVA-v1.6) [32, 31] perform significantly worse, highlighting the benefits of domain-specific training. Similar trends are observed on RSICD (Table D.12) and Sydney-Captions (Table D.14).

Table D.13 presents the performance of LISAt$_{\text{PRE}}$ on the RSVQA-LR dataset across Count, Presence, and Comparison categories. The model achieves the highest Presence accuracy (92.36) and Comparison accuracy (92.20), indicating strong performance in these tasks. In contrast, models such as LLaVA-1.5 and InternLM-XC2 report lower scores in Count and Presence. These results suggest that LISAt$_{\text{PRE}}$ effectively handles multimodal reasoning and task-specific fine-tuning, particularly in Presence-based evaluations.

The ablation study in Table 5.3 evaluates different vision encoders and language models for LISAt$_{\text{PRE}}$ on the UCM-Captions dataset. Among the vision encoders, RemoteCLIP (which we use in LISAt) significantly outperforms both Geo-CLIP and Sat-CLIP on all domains, while slightly outperforming the base CLIP models as it is specifically trained to learn geospatial relationships. Models using LLama 2 as a base LLM are notably worse than Vicuna. We found that SatCLIP demonstrates low scores because it is primarily optimized for location encoding rather than fine-grained object recognition. GeoCLIP shows inconsistent performance due to its design focus on large-scale geolocalization, which limits its effectiveness for object-level understanding. These findings highlight that both the vision encoder and the language model play crucial roles, with RemoteCLIP and Vicuna forming the most effective pairing for remote sensing imagery.

5.3 Limitations and Failure Cases

While LISAt outperforms all existing reasoning segmentation models, it is not perfect. Appendix E.2 highlights examples of failure cases in our pipeline. In some instances, LISAt struggles to produce accurate predictions when images are cloudy or when key features are obscured. Other challenges arise when the query is too vague like ‘‘Identify the plane in the bottom-right of the image.‘‘ while there are several planes in the bottom right corner of the image. We hypothesize that training on a larger dataset and refining the query design could help mitigate these issues. Another issue arises from the ground truth masks generated by GeoSAM in the GRES dataset. In some cases, the underlying ground truth mask is incorrect, and LISAt is occasionally penalized even when making correct predictions, as demonstrated in Appendix E.3.

6 Conclusion

In this paper, we introduce PreGRES and GRES, a new open dataset for geospatial reasoning segmentation, along with LISAt, an open-source foundation model for referring segmentation in remote-sensing. Together, they represent a first step toward models capable of generating task-specific outputs such as masks when reasoning about the geospatial world using natural language. With promising future directions in scaling, model fusion, and multimodal learning, we hope this effort lays a strong foundation for advancing geospatial AI through the fusion of vision and language.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021. URL `https://arxiv.org/abs/2103.15691`.

[4] Yakoub Bazi, Mohamad Mahmoud Al Rahhal, Mohamed Lamine Mekhalfi, Mansour Abdulaziz Al Zuair, and Farid Melgani. Bi-modal transformer-based approach for visual question answering in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021. URL `https://arxiv.org/abs/2102.05095`.

[6] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[7] Long Chen, Zhiyuan Qu, Yao Zhang, Jingyang Liu, Ruwen Wang, and Dezheng Zhang. Edge enhanced gciffnet: A multiclass semantic segmentation network based on edge enhancement and multiscale attention mechanism. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[8] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[9] Qimin Cheng, Haiyan Huang, Yuan Xu, Yuzhuo Zhou, Huanying Li, and Zhongyuan Wang. Nwpu-captions dataset and mlca-net for remote sensing image captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[11] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 172–181, 2018.

[12] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.

[13] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series, 2019. URL `https://arxiv.org/abs/1807.01232`.

[14] Johannes Flotzinger, Philipp J Rösch, Christian Benz, Muneer Ahmad, Murat Cankaya, Helmut Mayer, Volker Rodehorst, Norbert Oswald, and Thomas Braml. dacl-challenge: Semantic segmentation during visual bridge inspections. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 716–725, 2024.

[15] Tianyi Gao, Wei Ao, Xing-Ao Wang, Yuanhao Zhao, Ping Ma, Mengjie Xie, Hang Fu, Jinchang Ren, and Zhi Gao. Enrich distill and fuse: Generalized few-shot semantic segmentation in remote sensing leveraging foundation model's assistance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2771–2780, 2024.

[16] Lasse H Hansen, Simon B Jensen, Mark P Philipsen, Andreas Møgelmose, Lars Bodum, and Thomas B Moeslund. Opentrench3d: A photogrammetric 3d point cloud dataset for semantic segmentation of underground utilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7646–7655, 2024.

[17] Genc Hoxha et al. A novel svm-based decoder for remote sensing image captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[19] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 108–124. Springer, 2016.

[20] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36: 72096–72109, 2023.

[21] Jeremy Andrew Irvin, Emily Ruoyu Liu, Joyce Chuyi Chen, Ines Dormoy, Jinyoung Kim, Samar Khanna, Zhuo Zheng, and Stefano Ermon. Teochat: A large vision-language assistant for temporal earth observation data. *arXiv preprint arXiv:2410.06234*, 2024.

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[23] 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.

[24] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[25] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[26] Lei Li. Cpseg: Finer-grained image semantic segmentation via chain-of-thought language prompting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 513–522, 2024.

[27] Xiang Li, Jian Ding, and Mohamed Elhoseiny. Vrsbench: A versatile vision-language benchmark dataset for remote sensing image understanding. *arXiv preprint arXiv:2406.12384*, 2024.

[28] Xuelong Li, Aihong Yuan, and Xiaoqiang Lu. Multi-modal gated recurrent units for image description. *Multimedia Tools and Applications*, 77(22):29847–29869, 2018.

[29] Chenyang Liu, Rui Zhao, and Zhenwei Shi. Remote-sensing image captioning based on multilayer aggregated transformer. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[30] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[33] Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12): 8555–8566, 2020.

[34] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[35] Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, and Xuelong Li. Exploring models and data for remote sensing image caption generation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2183–2195, 2017.

[36] T. Nathan Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning, 2016. URL `https://arxiv.org/abs/1609.04453`.

[37] OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry

Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

[38] OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason

Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL `https://arxiv.org/abs/2412.16720`.

[39] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[40] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *The Twelfth International Conference on Learning Representations*, 2024.

[41] Bo Qu, Xuelong Li, Dacheng Tao, and Xiaoqiang Lu. Deep semantic understanding of high resolution remote sensing image. In *2016 International conference on computer, information and telecommunication systems (Cits)*, pages 1–5. IEEE, 2016.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[43] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

[44] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024.

[45] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[46] Esther Rolf, Konstantin Klemmer, Caleb Robinson, and Hannah Kerner. Position: Mission critical–satellite data is a distinct modality in machine learning. In *Forty-first International Conference on Machine Learning*, 2024.

[47] Subhashree Subudhi, Ram Narayan Patro, Pradyut Kumar Biswal, and Fabio Dell'Acqua. A survey on superpixel segmentation as a preprocessing step in hyperspectral image analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:5015–5035, 2021.

[48] Rafi Ibn Sultana, Chengyin Lia, Hui Zhua, Prashant Khanduria, Marco Brocanellib, and Dongxiao Zhua. Geosam: Fine-tuning sam with multi-modal prompts for mobility infrastructure segmentation. *arXiv preprint arXiv:2311.11319*, 2023.

[49] Gencer Sumbul, Sonali Nayak, and Begüm Demir. Sd-rsic: Summarization-driven deep remote sensing image captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 59(8): 6922–6934, 2020.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: open and efficient foundation language models. arxiv. *arXiv preprint arXiv:2302.13971*, 2023.

[51] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018.

[52] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[53] Binqiang Wang, Xiangtao Zheng, Bo Qu, and Xiaoqiang Lu. Retrieval topic recurrent memory network for remote sensing image captioning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:256–270, 2020.

[54] Di Wang, Jing Zhang, Bo Du, Minqiang Xu, Lin Liu, Dacheng Tao, and Liangpei Zhang. Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. *Advances in Neural Information Processing Systems*, 36:8815–8827, 2023.

[55] XuDong Wang, Shaolun Zhang, Shufan Li, Konstantinos Kallidromitis, Kehan Li, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Segllm: Multi-round reasoning segmentation. *arXiv preprint arXiv:2410.18923*, 2024.

[56] Marie Weiss, Frédéric Jacob, and Grgory Duveiller. Remote sensing for agricultural applications: A meta-review. *Remote sensing of environment*, 236:111402, 2020.

[57] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See say and segment: Teaching lmms to overcome false premises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13459–13469, 2024.

[58] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3858–3869, 2024.

[59] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding, 2021. URL `https://arxiv.org/abs/2109.14084`.

[60] Kelvin Xu. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[61] Linrui Xu, Ling Zhao, Wang Guo, Qiujun Li, Kewang Long, Kaiqi Zou, Yuhan Wang, and Haifeng Li. Rs-gpt4v: A unified multimodal instruction-following dataset for remote sensing image understanding. *arXiv preprint arXiv:2406.12479*, 2024.

[62] Shiyu Xuan, Qingpei Guo, Ming Yang, and Shiliang Zhang. Pink: Unveiling the power of referential comprehension for multi-modal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13838–13848, 2024.

[63] Zhenghang Yuan, Lichao Mou, Qi Wang, and Xiao Xiang Zhu. From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data. *IEEE transactions on geoscience and remote sensing*, 60:1–11, 2022.

[64] Zhenghang Yuan, Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. Rrsis: Referring remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[65] Zhiqiang Yuan, Wenkai Zhang, Kun Fu, Xuan Li, Chubo Deng, Hongqi Wang, and Xian Sun. Exploring a fine-grained multiscale method for cross-modal remote sensing image retrieval. *arXiv preprint arXiv:2204.09868*, 2022.

[66] 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.

[67] Yang Zhan, Zhitong Xiong, and Yuan Yuan. Skyeyegpt: Unifying remote sensing vision-language tasks via instruction tuning with large language model. *arXiv preprint arXiv:2401.09712*, 2024.

[68] Meimei Zhang, Fang Chen, and Bin Li. Multi-step question-driven visual question answering for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[69] Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, and Xuerui Mao. Earthgpt: A universal multi-modal large language model for multi-sensor image comprehension in remote sensing domain. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[70] Xiangrong Zhang, Xin Wang, Xu Tang, Huiyu Zhou, and Chen Li. Description generation for remote sensing images using attribute attention mechanism. *Remote Sensing*, 11(6):612, 2019.

[71] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14227–14238, 2024.

[72] Zixiao Zhang, Licheng Jiao, Lingling Li, Xu Liu, Puhua Chen, Fang Liu, Yuxuan Li, and Zhicheng Guo. A spatial hierarchical reasoning network for remote sensing visual question answering. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023.

[73] Xiangtao Zheng, Binqiang Wang, Xingqian Du, and Xiaoqiang Lu. Mutual attention inception network for remote sensing visual question answering. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

Appendix

In this appendix, we include several additional discussions:

- Appendix A expands on the societal impact of this project.
- Appendix B details the code release, including links to the codebases and datasets used in this project.
- Appendix C outlines the prompt structure used for engineering the GRES dataset for LISAT, provides further details on its class distribution as well as its quality verification, and discusses additional experiments.
- Appendix D presents additional details on the PreGRES dataset used to fine-tune LISAT$_{\text{PRE}}$, discussing its composition and further evaluations.
- Appendix E showcases qualitative results, highlighting both successful and failure cases, as well as instances where LISAT was penalized due to incomplete Ground Truth annotations generated by GeoSAM (GT).
- Appendix F provides some additional directions for future work.

Acknowledgements

Authors, as part of their affiliation with UC Berkeley, were supported in part by the National Defense Science and Engineering Graduate Fellowship program, the National Science Foundation, the U.S. Department of Defense, the Department of Defense's High Performance Computing Modernization Program (HPCMP) and/or the Berkeley Artificial Intelligence Research (BAIR) industrial alliance program. Sky Computing Lab is supported by gifts from Accenture, AMD, Anyscale, Cisco, Google, IBM, Intel, Intesa Sanpaolo, Lambda, Microsoft, NVIDIA, Samsung SDS, SAP, and VMware. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of any sponsor, the Department of Defense, or the U.S. Government.

A Impact, Limitations and Ethical Considerations

This paper presents advancements in reasoning segmentation for remote sensing tasks. LISAT is a method that is able to reason over arbitrary remote sensing images and output both explanations and segmentation masks for objects of interest. These kinds of workflows are extremely common across multiple fields. For example, disaster management personnel may want to know which roads leading to an airport are undamaged, and why. LISAT is the first such model that can simultaneously answer both components of such questions.

Broadly, LISAT has impacts in numerous domains such as environmental monitoring, urban planning, and search and rescue. However, one of the biggest uses of satellite imaging is surveillance. Being cognizant of this, our work is primarily based on datasets that have been widely adopted by the remote sensing community over interesting, cluttered scenes that do not capture any individual entity.

In other words, the challenge of surveillance is inherent to many imaging technologies, including satellite data, and is not unique to our method. In that sense, the goal of LISAT is to advance geospatial analysis for a wide range of applications, such as environmental monitoring and disaster response. We train on a dataset that the community has widely accepted to be benign (`xView` [25]).

This research is intended solely for scientific and humanitarian applications such as disaster response, environmental monitoring, and urban planning. While the models and datasets developed here demonstrate strong geospatial reasoning capabilities, they should not be used for surveillance, targeting, or any activity that infringes upon individual privacy or human rights. All data sources used are publicly available and non-sensitive, and we release our work under a research-oriented license to encourage transparent, responsible innovation. We also advocate that any derivative work or downstream application of this model align with ethical AI development and use guidelines. We believe that the most effective way to mitigate the risks of misuse is to foster open and transparent research in this area, and we encourage continued discourse on the implications of geospatial AI in real-world applications.

B Code Release

The project page for this paper is available here. Our code for LISAT, derived from the Apache 2.0-licensed LISA codebase [24], as well as the curated datasets are publicly released under the MIT license (or their respective licenses) and could also be found on the same page.

C More on GRES

C.1 Prompt Engineering

As outlined in Section 3, we used GPT-4o to generate the final prompt in two stages, detailed below.

C.1.1 Promt Engineering Stage 1

In the first stage, we input a 512×512 chip into the model and prompt it, following the template below, to generate a sentence that accurately describes the item within the bounding box provided, as specified by the Ground Truth from `xView`

```
The size of the original image is (512,512).
This original image, where the image’s origin is at the top left corner,
      contains the following objects: {classes_list_str}.
Only focus on {class_name} in the image.
If {class_name} contains the word ’Other’, remove the word ’Other’ and
     use only the second word in {class_name} describing the class. In
     that case, make sure that second word in {class_name} starts with a
     lowercase letter.
The following are the bounding boxes [x, y, width, height] of objects of
      class {class_name}, where (x,y) represents the top left corner of
     the bounding box, and ’width’ represents the bounding box’s width,
     and ’height’ represents the bounding box’s height.
The bounding box of the {class_name} is at coordinates {bbox}.
Find visual features (color, shape, size, etc.) that can help find or
     segment {class_name} in the image.
Generate a sentence (not a question) that can uniquely segment or
     identify or find or locate {class_name} in this image, be concise
     and clear.
```

Where `{classes_list_str}`, `{class_name}`, and `{bbox}` are the ground truth list of classes, the object class name or category, and the bounding box of the object from the `xView` dataset bounding box and class annotations.

The model outputs a descriptive sentence in the variable `{unique_characteristics.query}`, which is then used to query the model again in the second stage, as shown below.

C.1.2 Prompt Engineering stage 2

Once the uniquely descriptive sentence is generated, we asked the model using the template below to come with a question to which the given sentence in `{unique_characteristics.query}` will be the answer.

```
The size of the original image is (512,512).
Only focus on {class_name} in the image.
In the original image, where the image’s origin is at the top left
     corner, the object is a {class_name} located at bounding box
     coordinates {bbox}.
The following are the bounding boxes [x, y, width, height] of objects of
      class {class_name}, where (x,y) represents the top left corner of
     the bounding box, and ’width’ represents the bounding box’s width,
     and ’height’ represents the bounding box’s height:
This original image, where the image’s origin is at the top left corner,
      contains the following objects: {classes_list_str}.
If {class_name} contains the word ’Other’, remove the word ’Other’ and
     use only the second word in {class_name} describing the class. In
     that case, make sure that second word in {class_name} starts with a
     lowercase letter.
{ ’ located at bounding box coordinates {bbox}.’ if include_bbox else
     ’.’}
Please generate a query that would help locate this {class_name} in the
     original image.
Your query will be the question to the answer provided by {
     unique_characteristics.query}.
For example, if the value contained in {unique_characteristics.query} is
      ’Look for a long rectangular shape with distinct wheels, typically
     metallic or painted in color, connected to a truck cab at the front
     ’, your query should be:
’Segment the blue car in the bottom right of the image with a long
     rectangular shape with distinct wheels, typically metallic or
     painted in color, connected to a truck cab at the front’
’Identify the blue car in the bottom right of the image with a long
     rectangular shape with distinct wheels, typically metallic or
     painted in color, connected to a truck cab at the front’
’Find the blue car in the bottom right of the image with a long
     rectangular shape with distinct wheels, typically metallic or
     painted in color, connected to a truck cab at the front’
’Locate the blue car in the bottom right of the image with a long
     rectangular shape with distinct wheels, typically metallic or
     painted in color, connected to a truck cab at the front’
’Show the blue car in the bottom right of the image with a long
     rectangular shape with distinct wheels, typically metallic or
     painted in color, connected to a truck cab at the front’.
Generate the query considering the sentence: {unique_characteristics.
     query}
{ ’and the location described by the bounding box.’ if include_bbox else
      ’.’}
Make sure to vary the start of your queries with key words such as ’
     Segment, Find, Locate, Show, Identify’ and similar synonyms. Do not
     overuse one over the others.
Rephrase the generated query to make it sound better.
{ ’Do not mention or use any location-related info such as: top, near
     the center in your query.’ if not include_bbox else ’’}
Do not output the exact bounding box coordinates, instead, output the
     locations such as: bottom-left, top-right, top-left, bottom-right,
     center, etc.
The response to the generated queries should be a JSON object in the
     following format and contain nothing else:
The response to the generated query should be a sentence, not a question
     .
Be concise and clear, start the sentence with: Locate, Segment, or
     Identify.
{"query": "<your_query_here>"}
```

Where {class_name}, {bbox}, {unique_characteristics.query}, and {class_name} are the ground truth class name or category of the object class name or category, its bounding box and the unique characteristics obtained from GPT-4 [1] in the first stage.

The final query is then treated as the principal query. To enhance query diversity, we ask GPT to rephrase the principal query into two additional variants, resulting in three distinct queries per image. We then use GeoSAM to generate corresponding masks, forming image-queries-mask tuples.

C.2 Dataset Quality Assurance

We use RGB images from the xView dataset [25], as referenced in our manuscript. Although the dataset covers regions in South America, Africa, Europe, Asia, and Australia, we agree that LISAT and GRES would benefit from additional datasets from around the world, as shown in Figure 4.4.

For the classes of the targeted object referenced in the GRES natural language queries associated with each image, we inherited them from the Quality Control and Gold Standards method used in the xView paper [25]. In their paper, the authors outline a three-tier quality assurance process: worker, supervisory, and expert stage. In the first stage, labelers reviewed each other's annotations in a rotating manner. During the supervisory phase, the process included checks for duplicate or incorrect labels, geometry errors, incomplete annotation coverage, misaligned features, and empty image tiles. In the final expert stage, annotations were compared against a gold standard dataset. This reference dataset was developed by the paper's co-authors and professional image analysts. It involves a manual labeling of six 1 km^2 image chips per batch. To meet the expert-level quality standard at this 3rd stage, annotation batches were required to achieve a minimum of 0.75 precision and 0.95 recall at a 0.5 Intersection over Union (IoU) threshold when evaluated against the gold standard.

For natural language queries in GRES, we generated three variations per RGB image and used cosine similarity to ensure they conveyed semantically equivalent information. Only those with a similarity/alignment score of 0.9 or higher were retained.

Regarding ground truth segmentation masks produced via GeoSAM, we used only the cropped RGB regions defined by the bounding boxes in the xView dataset [25]. Random batches of GeoSAM outputs were inspected by co-authors in a rotating manner. Their task was to validate or reject the generated masks. A randomly selected data point from a random batch was retained only if all participants unanimously agreed on its accuracy.

For the obtained data, we employ *Human Verification*, where multiple team members manually inspect randomly selected subsets of the dataset to verify the accuracy of the query-image-annotation triplets.

C.3 GRES Dataset Summary

Table C.7 below shows the LISAT dataset distribution per class. We have also provided bar charts for the dataset distributions in Figures C.5 through C.9.

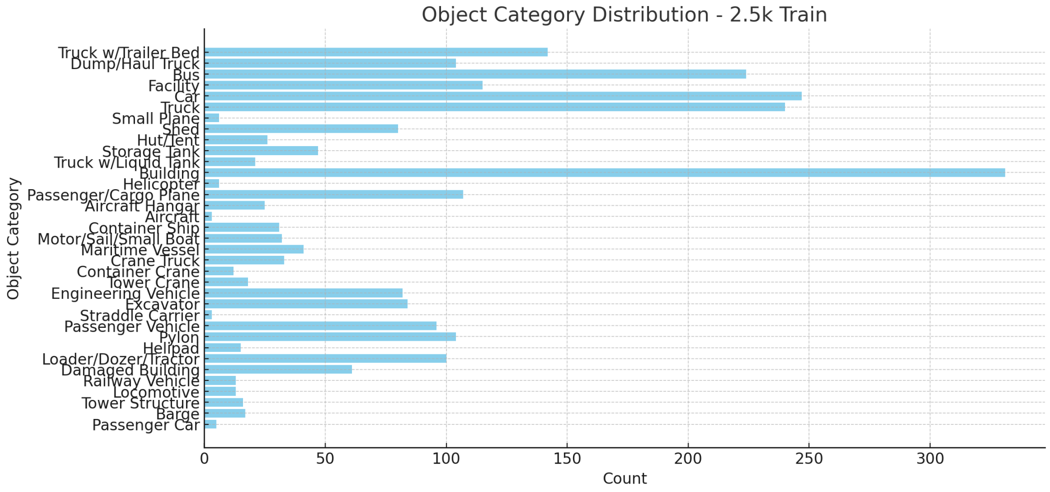


Figure C.5: Class Distribution of 33% Training Set

C.4 Additional Experiments

C.4.1 GPT vs. Human vs. Template style Queries

We start by asking whether there is a difference in language style and complexity between queries generated by Large Language Models and real analysts. Though using GPT-generated queries is an effective strategy, it is important to consider the potential differences between the two.

To preliminarily investigate this, we conducted a small-scale comparative analysis using 10 test examples. We created two additional query variants: (1) human-like rewordings and (2) template-

Table C.7: Summary of Object Categories Across Train, Validation, and Test Sets

Object Category	**Train (2.5k)**	**Train (4.5k)**	**Train (7.2k)**	**Val (0.5k)**	**Test (1.5k)**	**Test-L (0.5k)**	**Test-S (1k)**
Truck w/Trailer Bed	142	298	469	25	100	34	66
Dump/Haul Truck	104	148	208	16	50	18	32
Bus	224	417	671	61	139	8	131
Facility	115	197	370	28	66	44	22
Car	247	546	914	65	182	2	180
Truck	240	518	932	75	173	12	161
Small Plane	6	18	39	2	7	3	4
Shed	80	152	249	7	51	16	35
Hut/Tent	26	46	82	9	16	8	8
Storage Tank	47	74	120	8	25	14	11
Truck w/Liquid Tank	21	29	45	3	10	4	6
Building	331	548	937	69	183	102	81
Helicopter	6	12	19	2	4	1	3
Passenger/Cargo Plane	107	135	198	11	45	25	20
Aircraft Hangar	25	39	73	6	13	9	4
Aircraft	3	15	29	0	5	3	2
Container Ship	31	72	102	5	24	11	13
Motor/Sail/Small Boat	32	58	87	7	20	2	18
Maritime Vessel	41	92	134	12	31	21	10
Crane Truck	33	48	70	2	16	5	11
Container Crane	12	25	38	4	9	0	9
Tower Crane	18	42	57	6	14	7	7
Engineering Vehicle	82	115	166	15	39	11	28
Excavator	84	115	161	12	39	10	29
Straddle Carrier	3	7	14	2	3	2	1
Passenger Vehicle	96	145	215	15	49	0	49
Pylon	104	140	177	6	47	34	13
Helipad	15	21	32	2	8	6	2
Loader/Dozer/Tractor	100	137	186	7	46	7	39
Damaged Building	61	151	226	8	51	37	14
Railway Vehicle	13	22	26	1	8	8	0
Locomotive	13	21	32	3	8	4	4
Tower Structure	16	30	41	1	11	6	5
Barge	17	42	59	5	14	13	1
Passenger Car	5	14	27	1	5	1	4
Total	2500	4489	7205	500	1500	488	1023

based queries referencing specific image regions. Our evaluation showed in Table C.8 revealed that while GPT-style queries achieved slightly higher average performance on segmentation metrics, the differences were accompanied by relatively high variance, likely due to the tiny sample size.

This initial result suggests that GPT-generated queries are a reasonable proxy for human queries in the current setting, supporting the effectiveness of our dataset construction approach. However, we agree that a larger-scale collection of real human queries would provide a stronger validation and potentially improve the dataset further if augmented with such a collection.

Table C.8: Performance comparison across different query types on 10 test examples.

Type of Queries (10 test examples)	**cIoU (±)**	**gIoU (±)**
LISAt on Template-style queries	0.025 ± 0.014	0.045 ± 0.021
LISAt on Human-entered queries	0.037 ± 0.022	0.063 ± 0.040
LISAt on GPT-style queries (GRES Data)	0.050 ± 0.036	0.099 ± 0.045

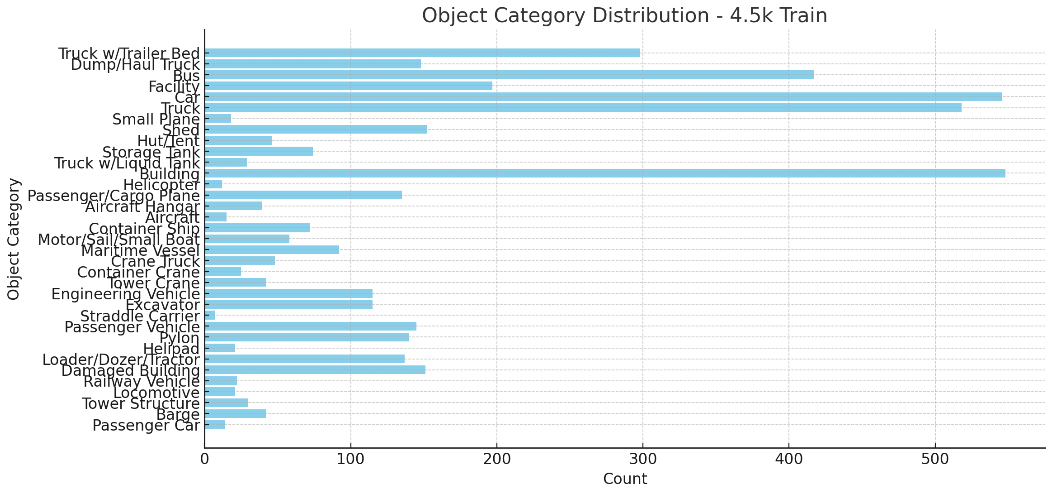


Figure C.6: Class Distribution of 66% Training Set

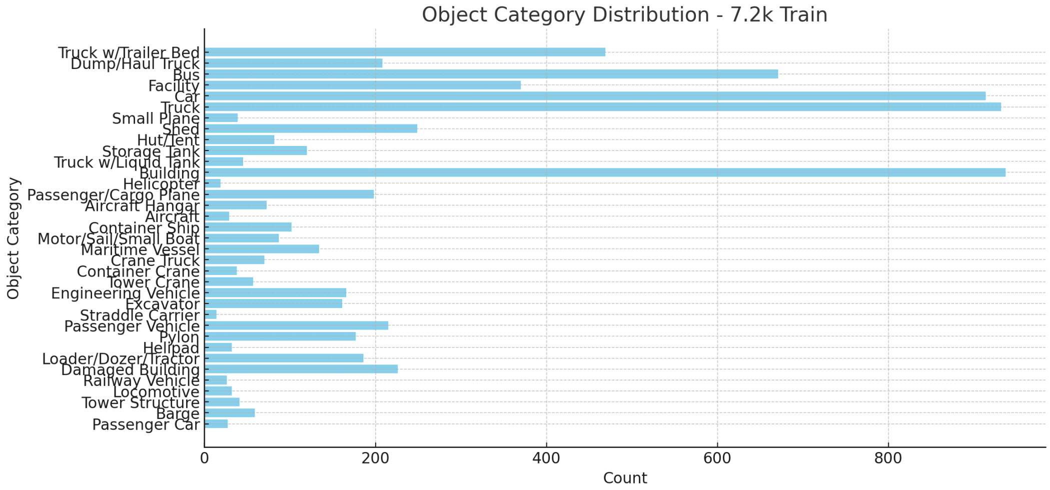


Figure C.7: Class Distribution of 100% Training Set

C.4.2 LISAT vs. LISA on Natural Images

We also evaluated and reported in Table C.9 the performance of LISAT on the natural images test set from the LISA benchmark [24]. `LISA-7B on LISA Natural Images Data` is the baseline model reported in the original paper. `LISA-7B (ft) on LISA Natural Images Data` refers to the fine-tuned version, where [24] note that performance improves after fine-tuning on 239 complex-reasoning samples. `LISAT on LISA Natural Images Data` represents our LISAT model evaluated on the same test set, while `LISAT on GRES Data` shows its performance on the GRES dataset.

Table C.9: Performance comparison across different models and datasets.

Type of Model and Data	cIoU	gIoU
LISA-7B on LISA Natural Images Data	0.341	0.368
LISA-7B (ft) on LISA Natural Images Data	0.484	0.473
LISAT on LISA Natural Images Data	0.326	0.341
LISAT (Ours) on GRES Data	0.245	0.275

The results shown in Table C.9 indicate that LISAT does perform slightly worse than the original `LISA-7B` model and its fine-tuned version on this domain. Specifically, LISAT achieved a cIoU of

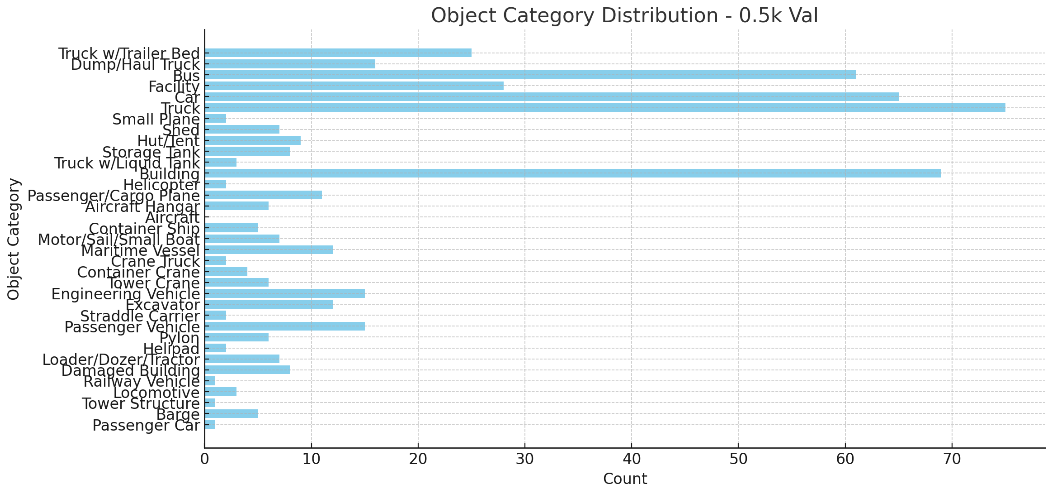


Figure C.8: Class Distribution of Validation Set

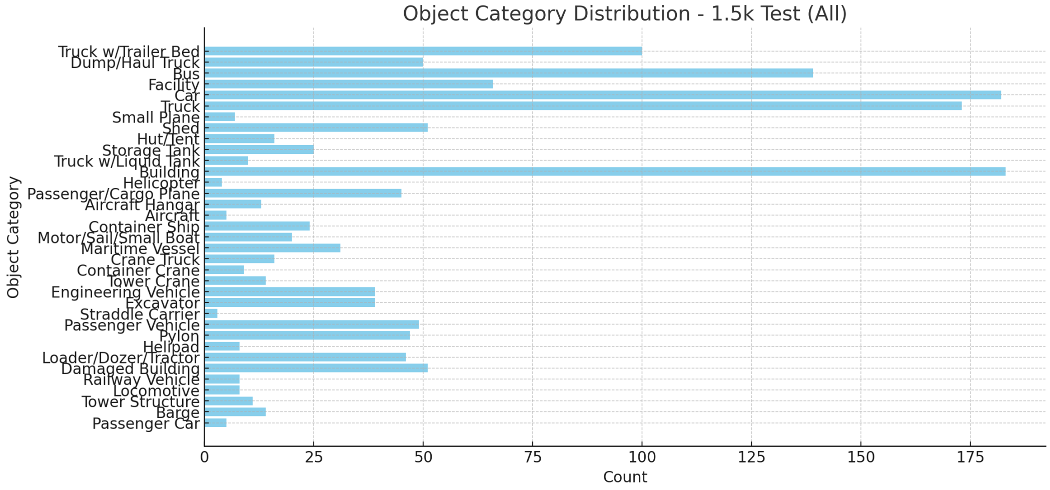


Figure C.9: Class Distribution of Testing Set

0.326 and gIoU of 0.341, compared to 0.341/0.368 for LISA-7B and 0.484/0.473 for the fine-tuned LISA-7B (ft).

While LISAT is not optimized for natural image reasoning tasks, its performance is still in a comparable range to the baseline LISA-7B model. The difference is expected, as LISAT is designed for generalization across geospatial and abstract reasoning segmentation tasks, and has not been fine-tuned on the LISA dataset. Thus, while it does not outperform models specialized or fine-tuned on natural image tasks, it remains competitive and shows promise as a more generalizable model.

C.4.3 LISAT's Latency Across Image resolutions

Because assessing inference speed and computational requirements is important for evaluating practical deployment feasibility, we have included an analysis in Table C.10, which reports the average inference time per image-query pair on a single NVIDIA A100 GPU across different image resolutions.

We found that at a standard resolution of 512×512, the model achieves 0.244 seconds per query, while maintaining competitive accuracy (cIoU: 0.245 ± 0.023, gIoU: 0.275 ± 0.009). As expected, inference becomes slower and less accurate at very low resolutions (e.g., 32×32), where performance drops (gIoU: 0.042 ± 0.004) and latency slightly increases (0.468s).

Table C.10: Effect of input image size on performance and inference time (All object sizes). Measured on a single NVIDIA A100.

Image Size	Object Size	cIoU (±)	gIoU (±)	Avg. Inference Time (s/image-query pair)
512 × 512	All	0.245 ± 0.023	0.275 ± 0.009	0.244
256 × 256	All	0.237 ± 0.029	0.207 ± 0.007	0.262
128 × 128	All	0.158 ± 0.019	0.130 ± 0.007	0.391
64 × 64	All	0.102 ± 0.010	0.061 ± 0.003	0.454
32 × 32	All	0.081 ± 0.007	0.042 ± 0.004	0.468

We believe the inference time increases at lower resolutions because the frozen vision encoder still processes inputs at a fixed size of 512×512 in our case. This requires lower-resolution input images to the resized back to 512x512, which adds an overhead. These resized images also contain fewer details, which makes it harder for the model to perform well since our pipeline resizes all inputs to the fixed resolution required by the encoder before inference.

For very small images (e.g., 32×32), the additional overhead from resizing operations and suboptimal GPU utilization can slightly increase inference time, as shown in Table C.10.

This indicates that reducing input resolution significantly degrades visual quality without providing meaningful speed benefits, which supports the use of higher resolutions (e.g., 512×512) in deployment settings.

C.4.4 LISAt$_{\text{PRE}}$ vs. GPT-4o vs. GPT-o1

At the time of this work, we note that since GPT-4o [37] and GPT-o1 [38] do not explicitly output segmentation masks, they cannot be fairly compared with LISAt. A specialized prompt must be engineered to extract the coordinates of points along the contour lines for the target object. Instead, we compare them to LISAt$_{\text{PRE}}$.

We found that GPT-4o [37] and GPT-o1 [38] yielded identical scores across all metrics and benchmark datasets, while significantly underperforming compared to LISAt$_{\text{PRE}}$ on the PreGRES test data. We verified that this results from both models returning generic or irrelevant outputs (e.g., hallucinated captions, answers unrelated to the query, or blank responses), likely due to their lack of grounding in geospatial semantics and structured output generation.

While these models represent the state of the art in general-purpose multimodal reasoning, they often require carefully crafted prompts to perform meaningfully on domain-specific tasks such as geospatial captioning or other domain-related VQA. This highlights the need for specialized VLMs like LISAt, which natively support geospatial semantics and reasoning.

D Additional Information on PreGRES

We conducted additional evaluations of LISAt$_{\text{PRE}}$. We show evaluation results on the NWPU Caption in Table 5.6, RSICD in Table D.12, Sidney-Caption in Table D.14. We also ran Count, Presence, Comparisaon and Area evaluation as was done in [61] in Table D.16.

Table D.11: Overview of Task Data Sources and Statistics

Task	Data Source	Train Images	Train QA Pairs	Test Images	Test QA Pairs
Image Captioning	NWPU-Captions	25200	125894	3150	1093
	RSICD	8734	17813	1093	1093
	RSITMD	4291	20096	-	-
	Sydney-Captions	497	2294	58	58
	UCM-Captions	1680	7999	210	210
Visual Question Answering	RSVQA-LR	572	57223	100	10004
	RSVQA-HR	6251	625340	2226	222684
	FloodNet	1448	4511	-	-
	RSIVQA	5401	19218	-	-
Visual Grounding	DIOR-RSVG	9466	19643	7936	18677
Region-level Captioning	DIOR-RSVG	9466	19643	-	-
Scene Classification	NWPU-RESISC45	31500	31500	-	-
Total	-	**104506**	**951174**	**14773**	**253819**

Table D.12: Comparison of Various Models for LISAt$_{\text{PRE}}$ on RSICD

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE L	CIDEr
VLAD + RNN [35]	49.38	30.91	22.09	16.77	19.96	42.42	103.92
VLAD + LSTM [35]	50.04	31.95	23.19	17.78	20.46	43.34	118.01
mRNN [41]	45.58	28.25	18.09	12.13	15.69	31.26	19.15
mLSTM [41]	50.57	32.42	23.29	17.46	17.84	35.02	31.61
mGRU [28]	42.56	29.99	22.91	17.98	19.41	37.97	124.82
mGRU embedword [28]	60.94	46.24	36.80	29.81	26.14	48.20	**159.54**
CSMLF [50]	57.59	38.59	28.32	22.17	21.28	44.55	52.97
SAA [33]	59.35	45.11	35.29	28.08	26.11	49.57	132.35
Soft-attention [60]	65.13	49.04	39.00	32.30	26.39	49.69	90.58
SD-RSIC [49]	64.50	47.10	36.40	29.40	24.90	51.90	77.50
RTRMN (semantic) [53]	62.01	46.23	36.44	29.71	28.29	**55.39**	151.46
RTRMN (statistical) [53]	61.02	45.14	35.35	28.59	27.51	54.52	148.20
SVM-D BOW [17]	61.12	42.77	31.53	24.11	23.03	45.88	68.25
SVM-D CONC [17]	59.99	43.47	33.55	26.89	22.99	45.57	68.54
MLAT [29]	66.90	51.13	41.14	34.21	27.31	50.57	94.27
Post-processing [20]	62.90	45.99	35.68	28.68	25.30	47.34	75.56
RS-GPT4V [61]	70.32	54.23	**44.02**	**36.83**	30.10	53.34	102.94
LLaVA-v1.5-7b [32]	38.36	18.27	8.46	3.57	14.64	27.36	16.96
LLaVA-v1.6-7b [31]	29.31	13.40	6.00	2.44	13.11	24.40	10.69
LISAt$_{\text{PRE}}$ (Ours)	**72.51**	**54.98**	43.77	36.10	**30.28**	53.80	118.39

Table D.13: Performance on RSVQA-LR (% accuracy).

Model	Count	Presence	Comparison
RSVQA [33]	67.01	87.46	81.50
EasyToHard [63]	69.22	90.66	87.49
Bi-Modal [4]	**72.22**	91.06	91.16
SHRNet [72]	73.87	91.03	90.48
LLaVA-1.5 [32]	26.81	54.72	66.22
InternLM-XC2 [12]	26.91	55.74	64.89
RS-GPT4V [61]	-	91.17	91.70
GeoChat [68]	-	91.09	90.33
Full-FT [61]	70.48	91.10	92.23
RS-GPT4V-LoRA-FT [61]	70.34	92.24	92.10
RS-GPT4V-MoE-LoRA-FT [61]	71.06	91.10	**92.55**
LLaVA-v1.5-7b [32]	18.66	53.98	66.22
LLaVA-v1.6-7b [31]	19.65	57.53	62.32
LISA-7B (baseline)	0.00	0.00	0.00
LISA-7B (fine-tuned on GRES)	25.86	79.80	84.41
LISAt$_{\text{PRE}}$ (Ours)	70.24	**92.36**	92.20

E Qualitative Analysis

In this section, we present a qualitative analysis of the model's performance, showcasing a range of success cases Appendix E.1, failure cases Appendix E.2, and instances where the ground truth (GT) was erroneous Appendix E.3. Success cases shown in Table E.17, Table E.18, Table E.19, Table E.20, and Table E.21 highlight scenarios where the model successfully aligns with the expected outcomes, demonstrating its ability to handle complex tasks accurately. Failure cases shown in Table E.22, however, indicate situations where the model struggles due to challenges such as occlusion, poor lighting, or ambiguous object representations, leading to incorrect predictions or missed detections. These cases reveal areas where model improvements are needed, particularly in dynamic environments or with less structured input data. Finally, GT mistake cases, as shown in Table E.23, refer to instances where the GT was erroneous but the model aligns with the expected ground truth annotations. The model is penalized here due to inherent inconsistencies in the dataset from the mask labeling with GeoSAM. These cases reveal the challenges posed by noisy or ambiguous ground truth data, highlighting the importance of dataset refinement and improved model calibration to reduce such errors. Together, these cases provide valuable insights into the model's performance, guiding future research and optimizations.

Table D.14: Comparison of Various Models for LISAt$_{\text{PRE}}$ on Sydney-Captions

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE L	CIDEr
VLAD + RNN [35]	56.58	45.14	38.07	32.79	26.72	52.71	93.72
VLAD + LSTM [35]	49.13	34.12	27.60	23.14	19.30	42.01	91.64
mRNN [41]	51.30	37.50	20.40	19.30	18.50	-	161.00
mLSTM[41]	54.60	39.50	22.30	21.20	20.50	-	186.00
mGRU [28]	69.64	60.92	52.39	44.21	31.12	59.17	171.55
mGRU embedword [28]	68.85	60.03	51.81	44.29	30.36	57.47	168.94
CSMLF [50]	59.98	45.83	38.69	34.33	24.75	50.18	75.55
SAA [33]	68.82	60.73	52.94	45.39	30.49	58.20	170.52
Soft-attention [60]	73.22	66.74	62.23	58.20	39.42	71.27	249.93
Hard-attention [60]	75.91	66.10	58.89	52.58	38.98	71.89	218.19
SD-RSIC [49]	72.40	62.10	53.20	45.10	34.20	63.60	139.50
SVM-D BOW [17]	77.87	68.35	60.23	53.05	37.97	69.92	227.22
SVM-D CONC [17]	75.47	67.11	59.70	53.08	36.43	67.46	222.22
Post-processing [20]	78.37	69.85	63.22	57.17	39.49	71.06	255.53
LLaVA-v1.5-7b [32]	41.04	19.62	10.80	4.69	13.71	31.38	10.89
LLaVA-v1.6-7b [31]	32.25	17.15	9.98	5.92	14.11	29.17	12.20
RS-GPT4V [61]	**82.26**	**75.28**	**68.57**	**62.23**	**41.37**	**74.77**	**273.08**
LISAt$_{\text{PRE}}$ (Ours)	77.92	68.30	60.75	54.24	38.50	69.92	216.36

E.1 Success Cases of LISAt

In this subsection, we present a selection of successful cases where LISAt accurately predicted object categories and configurations. These examples highlight the model's ability to generalize and perform well under varied conditions, demonstrating its effectiveness in real-world applications.

E.2 Failure Cases of LISAt

We examined failure cases where LISAt struggled to make accurate predictions in subsection 5.3. Some of these instances, where the model's performance could be improved, highlight the challenges it faces under complex conditions, such as cloudy or ambiguous scenes as shown in Table E.22.

E.3 Ground Truth Error Cases

Table E.23 displays cases where the model's predictions are affected by errors in the ground truth data. These errors highlight discrepancies between the model's output and the labeled data, shedding light on limitations within the dataset and the potential impact on evaluation metrics.

F Additional Future Work

Building on the promising performance of LISAt, we outline several directions for future work to enhance both the model and the GRES dataset:

1. **Incorporation of Temporal Data**
   It will be interesting to extend LISAt's capabilities by incorporating temporal geospatial data, enabling the model to reason over frame sequences. This includes investigating the effects of frame-rate downsampling and adapting architectures that leverage temporal vision encoders such as TimeSformer [5] and ViViT [3], as well as contrastive video-text pretraining approaches like VideoCLIP [59], in combination with different language encoders. These explorations aim to identify the most effective architectural combinations for spatiotemporal grounding and reasoning in remote sensing contexts.
2. **Expansion to Additional Modalities**
   To enhance generalization and robustness, it will be interesting to integrate additional modalities such as synthetic aperture radar (SAR), LiDAR, aerial imagery, and elevation data (e.g., digital surface models, DSM), in both static and temporal settings. Once collected and processed, these modalities will broaden the applicability of the model and enable it to handle more dynamic and realistic geospatial scenarios. Incorporating these diverse inputs will also help evaluate LISAt's cross-modal generalization capabilities.
3. **Dataset Enrichment and Potential Bias Mitigation**
   For future iterations of the GRES dataset, it will be valuable to augment it with additional

publicly available datasets such as SpaceNet [13] and COWC [36]. This enrichment will help address existing dataset potential biases and improve the robustness and fairness of LISAt across a wider range of environmental and sensor conditions. To further enhance data quality, a promising direction includes implementing an active-learning–driven `GeoSAM` correction pipeline, where low-confidence or inconsistent masks are automatically identified and sent for human review, yielding progressively refined annotations. In parallel, incorporating large-scale human-query evaluations will complement GPT-generated queries, allowing us to better assess how real-world and GPT-style linguistic diversity and ambiguity affect visual grounding performance and overall model generalization.

4. **Efficient Model Variants for Deployment**
   For deployment in resource-constrained environments, future avenues include exploring model compression techniques such as knowledge distillation, quantization, and pruning. These approaches will enable us to reduce model size and improve inference efficiency while maintaining competitive performance, thereby supporting broader accessibility and real-time applications of LISAt.

Through these efforts, we believe LISAt will turn into an even more comprehensive and generalizable foundation model for geospatial-language understanding, capable of reasoning across modalities and time with increased accuracy and efficiency.

Table D.15: Comparison of Vision and Language Encoders for LISAT$_{\text{PRE}}$ on UCM-Captions, NWPU-Captions, RSICD, and Sydney-Captions

Vision Encoder	Language Encoder	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE L	CIDEr	SPICE
UCM-Captions									
CLIP	Llama 2	85.57	79.02	73.81	69.03	45.49	80.10	328.82	52.21
CLIP336	Llama 2	84.86	77.81	72.06	66.97	44.70	78.97	324.61	50.46
SAT-CLIP	Llama 2	41.24	32.38	12.74	8.82	13.90	28.30	30.41	8.15
Geo-CLIP	Llama 2	44.57	26.22	17.37	12.77	15.61	32.22	44.64	11.67
RemoteCLIP	Llama 2	85.95	79.00	73.38	68.31	45.80	79.99	330.94	52.17
CLIP	Vicuna	84.93	77.80	72.05	66.68	45.62	80.04	329.32	52.00
CLIP336	Vicuna	85.40	78.81	73.34	68.28	45.84	79.66	324.89	51.55
SAT-CLIP	Vicuna	47.21	29.55	21.21	16.87	17.35	34.20	63.92	15.08
Geo-CLIP	Vicuna	53.77	37.86	29.50	24.56	21.77	42.86	109.20	21.15
RemoteCLIP	Vicuna	**88.23**	**82.07**	**77.08**	**72.34**	**47.78**	**83.13**	**355.32**	**54.15**
NWPU-Captions									
CLIP	Llama 2	87.25	77.53	69.89	63.53	43.33	76.59	180.81	31.38
CLIP336	Llama 2	86.70	76.38	68.43	62.04	42.55	75.56	176.54	30.75
SAT-CLIP	Llama 2	69.51	50.90	39.28	31.48	25.83	52.34	68.94	16.15
Geo-CLIP	Llama 2	74.36	58.26	47.59	39.96	30.12	58.48	97.67	19.78
RemoteCLIP	Llama 2	87.25	77.67	70.06	63.76	43.44	76.64	181.01	31.48
CLIP	Vicuna	86.62	76.76	69.03	62.59	43.09	76.18	179.96	31.09
CLIP336	Vicuna	87.47	77.79	70.13	63.78	43.47	76.42	181.94	31.15
SAT-CLIP	Vicuna	75.87	60.51	50.35	43.00	31.78	60.51	105.53	21.25
Geo-CLIP	Vicuna	77.98	63.94	54.24	46.96	34.10	63.87	121.54	23.44
RemoteCLIP	Vicuna	**88.5**	**79.3**	**72.0**	**65.8**	**44.4**	**77.5**	**185.7**	**32.2**
RSICD									
CLIP	Llama 2	60.51	43.02	32.34	25.60	25.41	46.20	76.30	25.46
CLIP336	Llama 2	70.05	51.72	40.06	32.36	28.19	50.45	107.18	28.35
Geo-CLIP	Llama 2	45.86	21.97	11.90	7.16	15.73	28.46	22.60	12.88
RemoteCLIP	Llama 2	68.19	49.65	38.25	30.74	27.64	49.91	101.05	27.94
SAT-CLIP	Llama 2	43.68	19.17	9.82	5.60	14.61	26.65	16.71	10.95
CLIP	Vicuna	62.46	44.31	33.36	26.47	25.65	47.08	81.22	25.56
CLIP336	Vicuna	70.40	52.09	40.44	32.79	28.50	50.74	108.37	28.82
SAT-CLIP	Vicuna	45.95	22.68	13.00	8.23	16.02	29.02	24.77	12.76
Geo-CLIP	Vicuna	49.19	26.41	16.17	10.80	17.67	31.74	31.44	15.14
RemoteCLIP	Vicuna	**72.51**	**54.98**	**43.77**	**36.10**	**30.28**	**53.80**	**118.39**	**30.54**
Sydney-Captions									
CLIP	Llama 2	78.59	69.78	62.50	56.35	39.09	70.00	220.50	45.20
CLIP336	Llama 2	78.48	69.35	62.25	56.14	38.54	68.98	211.95	43.25
SAT-CLIP	Llama 2	58.40	45.47	37.93	32.27	25.75	46.95	85.67	22.96
Geo-CLIP	Llama 2	68.91	56.44	48.91	43.09	30.75	55.86	153.84	31.33
RemoteCLIP	Llama 2	76.19	66.31	58.57	52.27	37.30	68.12	201.64	43.99
CLIP	Vicuna	76.42	67.79	60.49	53.95	38.02	68.78	204.76	**44.07**
CLIP336	Vicuna	77.68	**68.70**	61.37	**55.35**	38.41	**70.37**	213.92	45.22
SAT-CLIP	Vicuna	69.58	58.86	51.80	46.45	32.24	58.43	182.93	34.65
Geo-CLIP	Vicuna	71.68	60.83	53.67	47.76	34.33	61.49	177.19	36.60
RemoteCLIP	Vicuna	**77.92**	68.30	**60.75**	54.24	**38.50**	69.92	**216.36**	44.04

Table D.16: Performance Metrics for LISAt$_{\text{PRE}}$ on the RSVQA_LR

Model	**Count**	**Presence**	**Comparison**	**Area**
RSVQA [33]	67.01	87.46	81.50	85.24
EasyToHard [63]	69.22	90.66	87.49	85.92
Bi-Modal [4]	**72.22**	91.06	91.16	86.27
SHRNet [72]	73.87	91.03	90.48	**86.35**
LLaVA-1.5 [32]	26.81	54.72	66.22	1.45
InternLM-XC2 [12]	26.91	55.74	64.89	5.94
RS-GPT4V [61]	-	91.17	91.70	-
GeoChat [68]	-	91.09	90.33	-
Full-FT [61]	70.48	91.10	92.23	86.00
LoRA [61]	70.34	92.24	92.10	85.84
MoE LoRA [61]	71.06	91.10	**92.55**	85.82
LLaVA-v1.5-7b [32]	18.66	53.98	66.22	58.00
LLaVA-v1.6-7b [31]	19.65	57.53	62.32	62.00
LISAt$_{\text{PRE}}$ (Ours)	70.24	**92.36**	92.20	61.43

Table E.17: Comparison of Predictions and Ground Truth Across Models

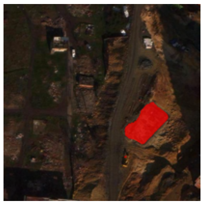
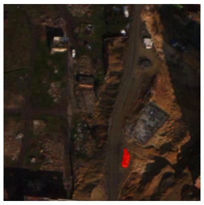
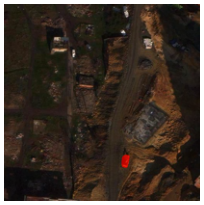
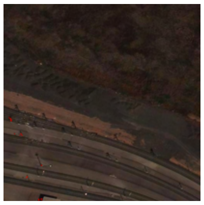
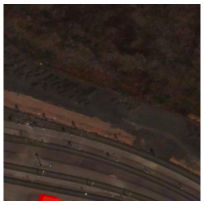
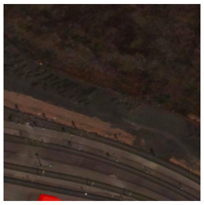
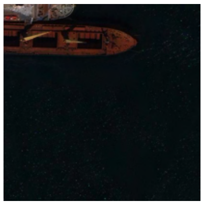
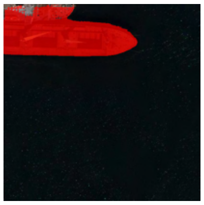
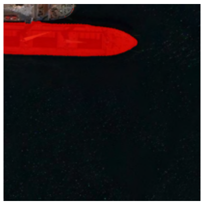
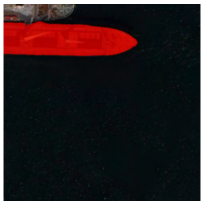
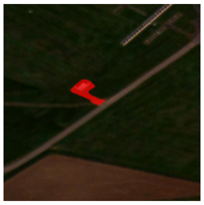
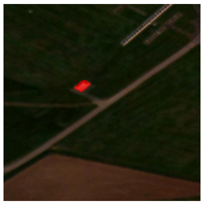
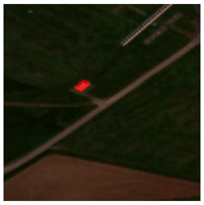
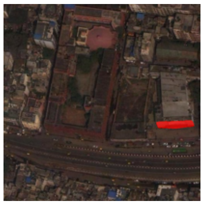
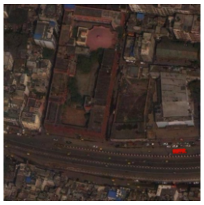
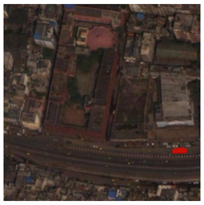
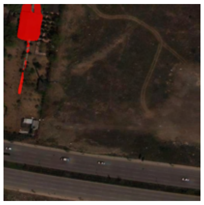
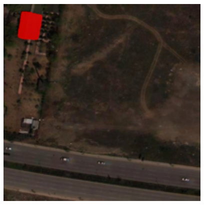
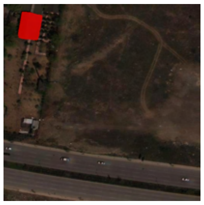
Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Identify the excavator by locating the bright yellow arm and bucket against the darker background.				
Locate the building with a beige facade and a dark brown roof in the image.				
Locate the large, elongated structure with stacked rectangular containers and a reddish-brown deck, characteristic of a container ship, against the dark water background.				
Locate the building in the center-left of the image.				
Locate the long, green vehicle with rectangular windows and wheels, positioned horizontally across the image.				
Locate the building in the top-left of the image.				

Table E.18: Comparison of Predictions and Ground Truth Across Models (Cont.)

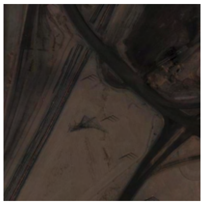
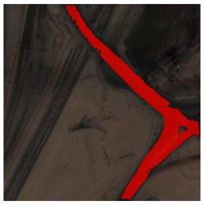
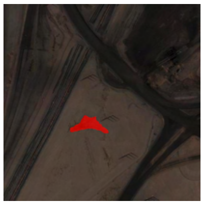
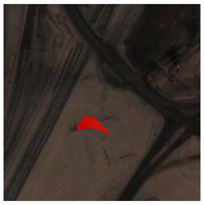
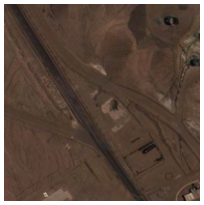
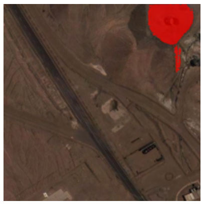
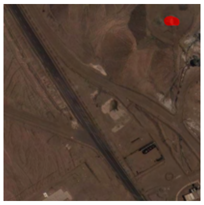
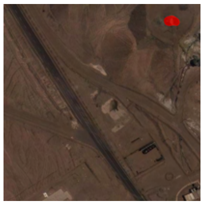
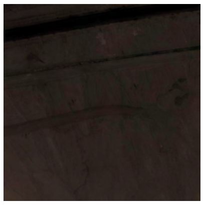
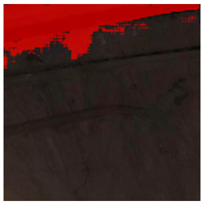
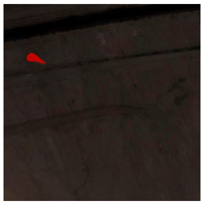
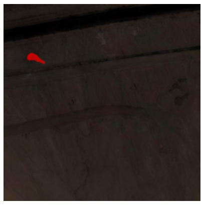
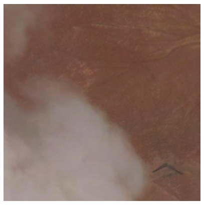
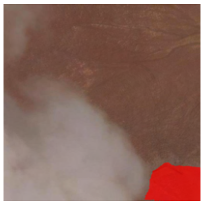
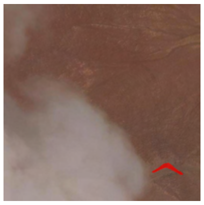
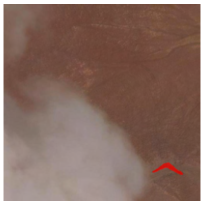
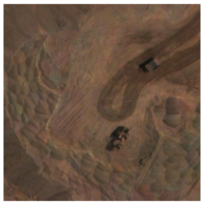
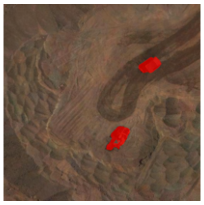
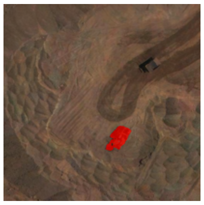
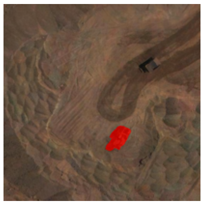
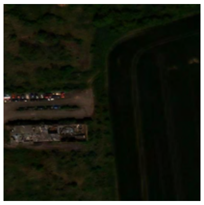
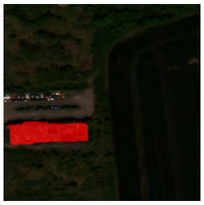
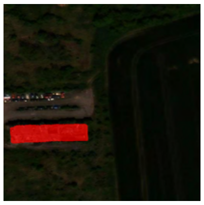
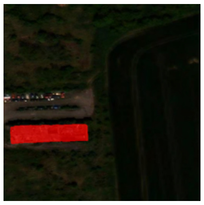
Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Identify the triangular metal structure with intersecting lines, standing vertically in the image.				
Identify the circular structure with a metallic appearance and distinct shadow, contrasting against the surrounding terrain.				
Identify the pylon in the top-left area of the image.				
Identify the pylon located in the bottom-right of the image.				
Identify the engineering vehicle with a metallic appearance and distinct geometric shapes against the brown background.				
Identify the damaged building with an irregular, fragmented roof structure and scattered debris contrasting with surrounding vegetation.				

Table E.19: Comparison of Predictions and Ground Truth Across Models (Cont.)

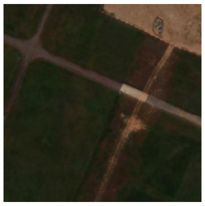
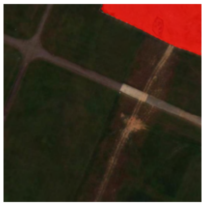
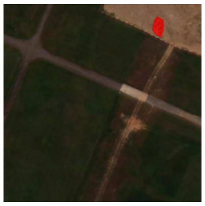
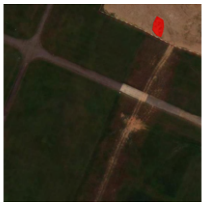
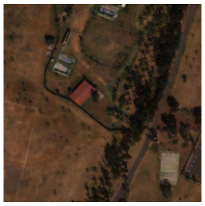
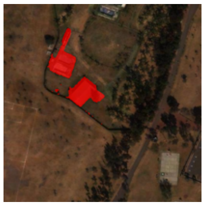
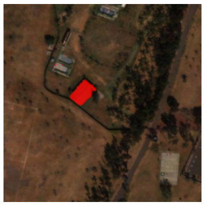
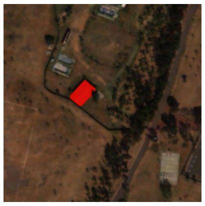
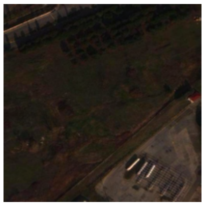
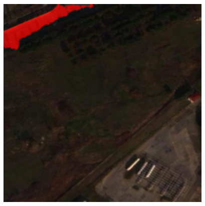
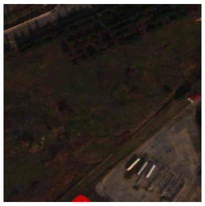
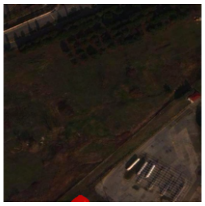
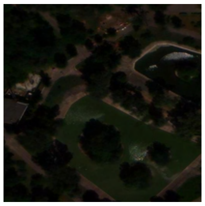
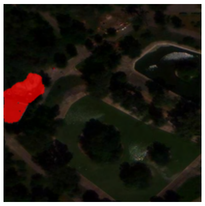
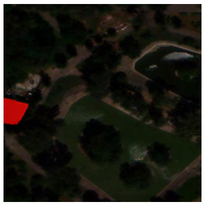
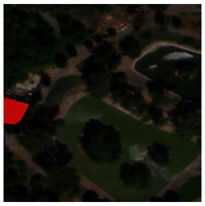
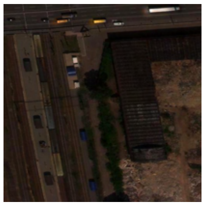
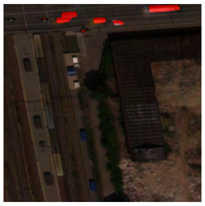
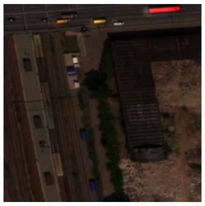
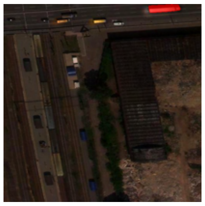
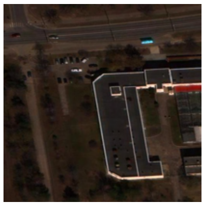
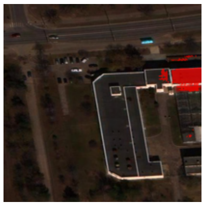
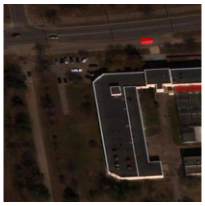
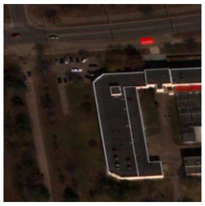
Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Segment the damaged building located in the top-right of the image.				
Identify the building in the center-left of the image.				
Identify the building with a unique vertical dark brown structure with a slight curvature on the edge.				
Identify the large, rectangular building with a dark roof and multiple visible roof fixtures.				
Locate the trailer bed in the top-right of the image, characterized by a long rectangular shape with distinct wheels, typically metallic or painted in color, attached to a truck cab.				
Identify the liquid tank in the top-right of the image with a long rectangular shape connected to a truck cab at the front.				

Table E.20: Comparison of Predictions and Ground Truth Across Models (Cont.)

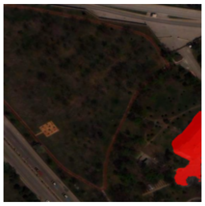
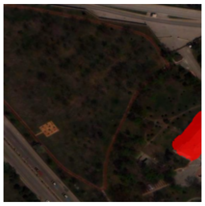
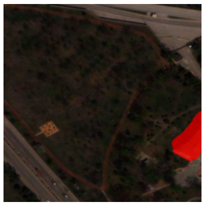
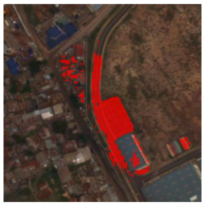
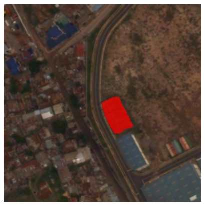
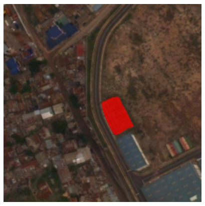
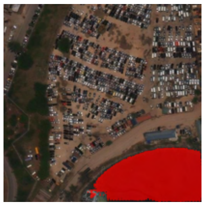
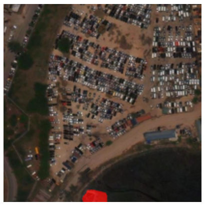
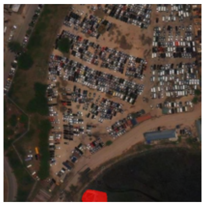
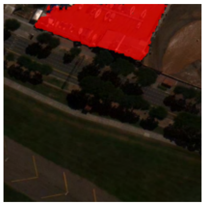
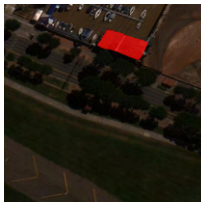
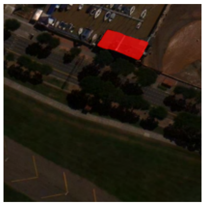
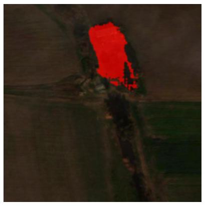
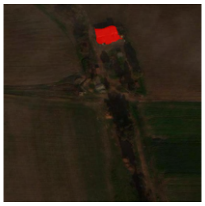
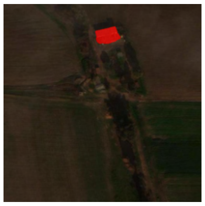
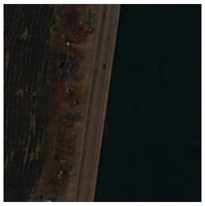
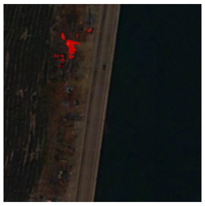
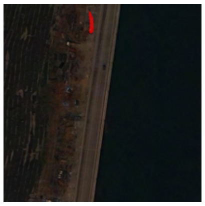
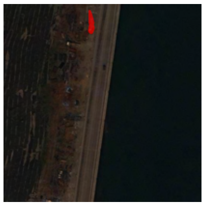
Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Locate the building with a reddish-brown roof next to a dark black structure in the image.				
Identify the damaged building in the center of the image.				
Locate the maritime vessel in the bottom-right of the image.				
Identify the building with a rectangular shape, dark roof, and noticeable white lines across its surface, set against a brownish background with green areas nearby.				
Identify the building with a grayish roof and white linear features.				
Locate the engineering vehicle in the top-left of the image.				

Table E.21: Comparison of Predictions and Ground Truth Across Models (Cont.)

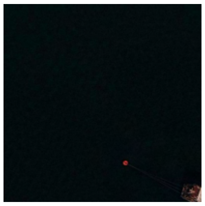
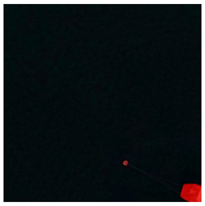
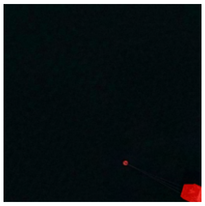
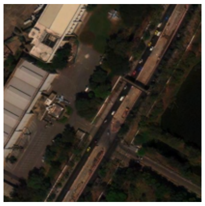
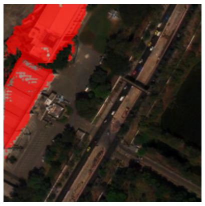
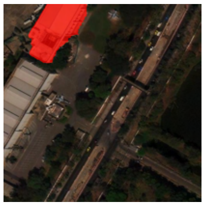
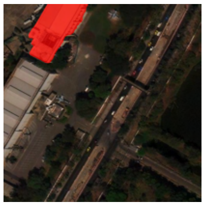
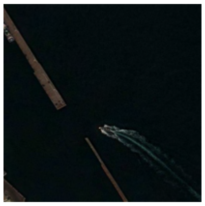
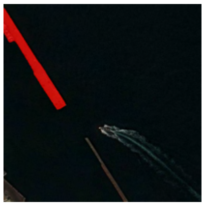
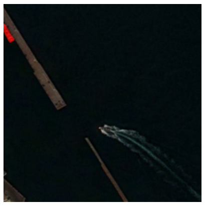
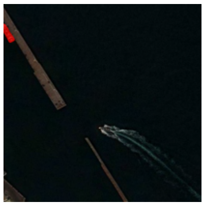
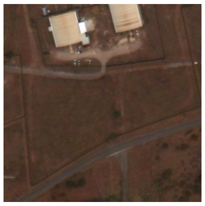
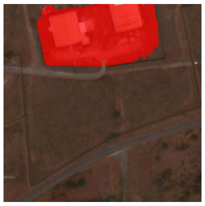
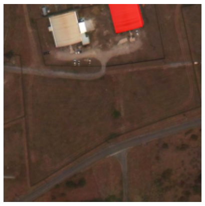
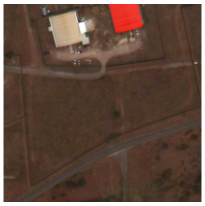
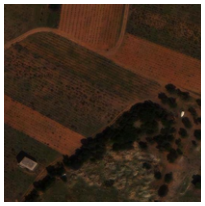
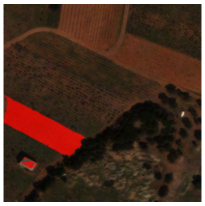
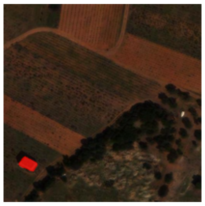
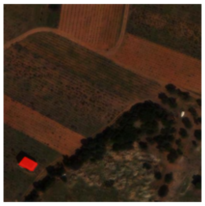
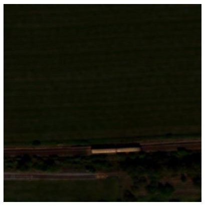
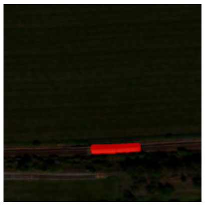
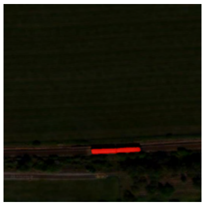
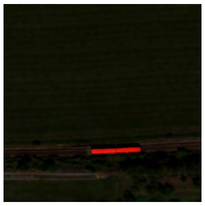
Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Locate the large rectangular structure with stacked, multicolored containers floating on water as the container ship.				
Locate the building in the top-left corner of the image.				
Identify the maritime vessel near the top-left corner of the image.				
Identify the aircraft hangar with the large rectangular structure and curved roof, displaying a uniform beige coloration and surrounded by open areas.				
Identify the large rectangular brown building with a flat roof surrounded by vegetation.				
Identify the railway vehicle with an elongated, rectangular shape and a metallic texture contrasting against the dark background.				

Table E.22: Failure Cases

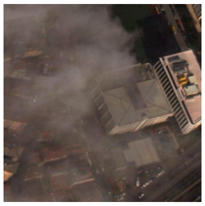
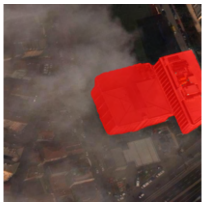
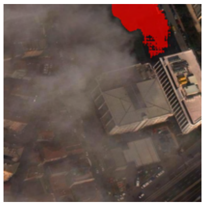
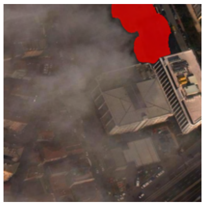
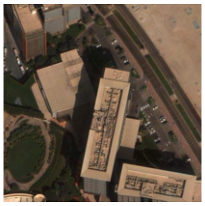
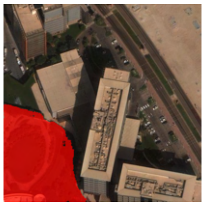
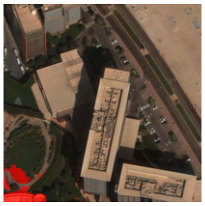
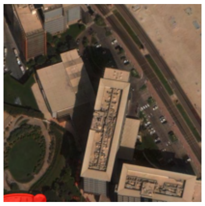
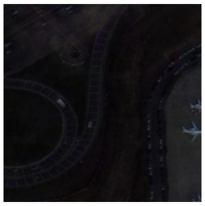
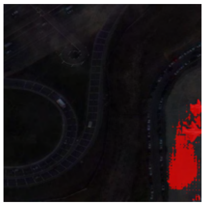
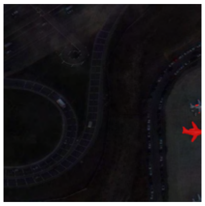
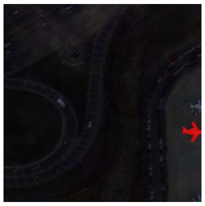
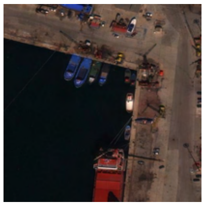
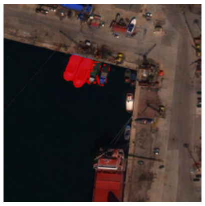
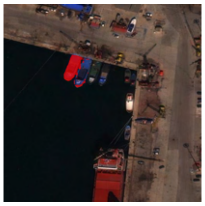
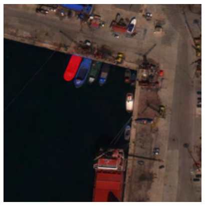
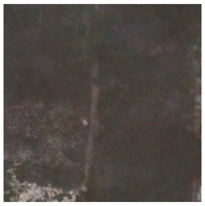
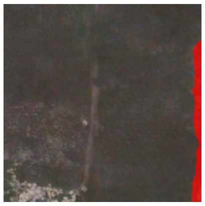
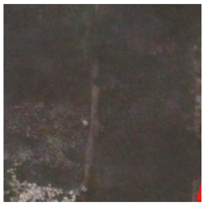
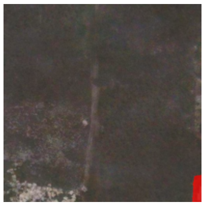
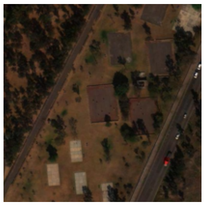
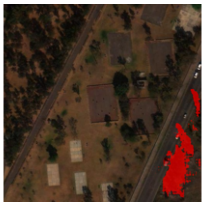
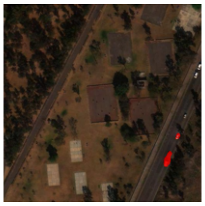
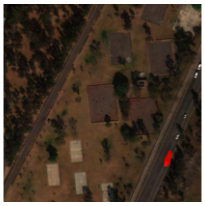
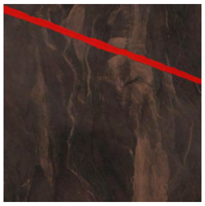
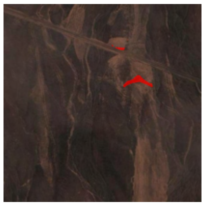
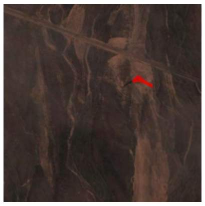
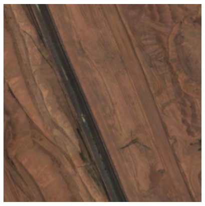
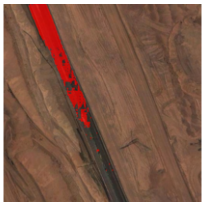
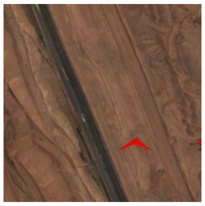
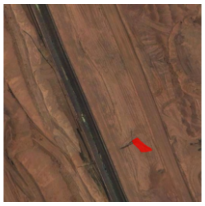
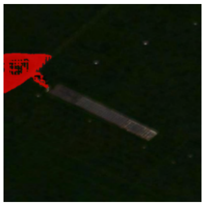
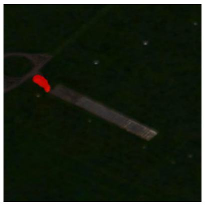
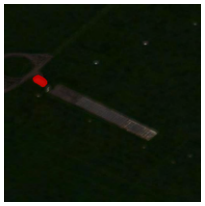
Queries	**RGB**	**LISA**	**LISAT (Ours)**	**Ground Truth**
Locate the facility in the top-center of the image for identification.				
Find the facility in the bottom-left corner of the image.				
Identify the plane in the bottom-right of the image.				
Locate the barge in the top-left of the image.				
Locate the building with a distinctive light gray color and rectangular shape against the darker background.				
Identify the trailer in the bottom-right of the image with a distinct shape, typically metallic or painted, connected to a truck cab at the front.				

Table E.23: GT Mistake Cases

Queries	RGB	LISA	LISAT (Ours)	Ground Truth
Identify the pylon in the top-right area of the image.				
Identify the vertical, metallic structure with a lattice framework contrasting against the brown, earthy background.				
Identify the building with a large, rectangular structure and a distinct reddish-brown roof, surrounded by greenery.				

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly present our main contributions: the introduction of a new multi-modal dataset and a baseline model for Geo-spatial Artificial Intelligence. These contributions are substantiated by experimental results, Code Release, and additional details provided in the rest of the appendix, accurately reflecting the scope and limitations of our work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our proposed pipeline are discussed in the Limitations and Failure Cases section in the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper provides a comprehensive disclosure of all information necessary to reproduce the main experimental results that support the core claims and conclusions. This includes detailed descriptions of dataset construction in the Geospatial Reasoning Segmentation Dataset section, model architecture details in the Training VLMs for Geospatial Reasoning Segmentation section, and training and evaluation procedures in the Experimental Results section, ensuring reproducibility independent of the released code and data.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have released all code and dataset for both GitHub and HuggingFace , with sufficient instructions to faithfully reproduce the main experimental results, as described in the paper on project page.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discussed all experimental and evaluation setting/details experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: While evaluating LISAT, we provided standard deviations in the model performance Table 5.4 and Figure 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute ressources used for experiments in this paper are provided in the Implementation Details section of experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This DB track submission complies with the NeurIPS Code of Ethics. It adheres to the requirements outlined in the single-blind submission guidelines, provides the necessary dataset and benchmark code, and aligns with the specific scope and criteria for dataset and benchmark paper submissions.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We highlighed potential positive societal impacts and negative societal impact in the Impact Statement section of the paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: . We have discused this in our Impact Statement.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All licenses for our code, models, and datasets are provided in their respective repositories. The code and models are released under the MIT License. The GRES dataset annotations are released under the CC BY-NC-SA 4.0 license, and its images inherit the license of the xView dataset (CC BY-NC-SA 4.0). The PreGRES annotations are released under the MIT License, while the images are subject to the respective licenses of their original source datasets. Citations for the relevant papers are provided in Geospatial Reasoning Segmentation Dataset section.

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All new assets are documented in the respestive sections 3, 4, and 5.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: As this paper focuses on advancing research in Vision-Language Models (VLMs), it references the various large language models (LLMs) used in our experiments throughout the content.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.