From Pixels to Semantics: Can a Single Instruction-Tuned VLM Unify Geospatial Building Analysis?

Guneet Mutreja[1], Harisankar Harikumar[2], Chaikal Amrullah[1], Ksenia Bittner[1]

[1] German Aerospace Center (DLR), Münchener Straße 20, Weßling -
(guneet.mutreja, chaikal.amrullah, ksenia.bittner)@dlr.de
[2] Dept. of Civil Engineering, Karlsruhe Institute of Technology, Kaiserstraße 12, - unkyy@student.kit.edu

Keywords: Vision–Language Models, Instruction Tuning, Remote Sensing, Building Analysis, Multi-Task Learning

Abstract

The analysis of buildings from aerial imagery is fundamental for urban planning and disaster response, yet it traditionally requires separate specialized models for tasks such as segmentation, detection, and semantic querying. Generalist Vision-Language Models (VLMs) offer a promising alternative, but adapting them to high-resolution remote sensing remains challenging. This paper proposes and investigates a data-centric methodology for adapting Google’s PALIGEMMA2 into a unified geospatial building analyzer. The main contribution is a pipeline that converts single-modality building polygon annotations into a multi-task instruction-tuning dataset of 16,500 samples spanning segmentation, detection, Visual Question Answering (VQA), and captioning. We conduct a rigorous study addressing three questions: (1) Can a single instruction-tuned VLM outperform specialized models in a multi-task setting? (2) What are the synergistic benefits of multi-task learning? (3) How data-efficient is this adaptation process? Results show that the unified model substantially outperforms the zero-shot PaliGemma2 baseline and strong single-task fine-tuned variants on three of four tasks, while remaining competitive on the fourth. We also observe a strong synergistic effect: multi-task training on visual localization and semantic tasks improves performance on individual localization tasks. Furthermore, high performance is achieved with a surprisingly small instruction dataset. This work provides a complete methodology for efficiently adapting VLMs to multi-task geospatial analysis, suggesting a path toward generalist models in remote sensing. To support further research and fair comparison, the dataset is available at: https://chaikalamrullah.github.io/RoofVIP/

1. Introduction

Building analysis from Very-High Resolution (VHR) aerial imagery is a cornerstone of photogrammetry and remote sensing. Applications ranging from urban planning and cartographic updates to disaster response and energy modeling rely on the accurate extraction of building footprints, locations, and attributes. For decades, this has been a central focus of the ISPRS community. Traditionally, the approach has been to develop a fragmented ecosystem of highly specialized models. For example, a U-Net (Ronneberger et al., 2015) or similar architecture might be trained for semantic segmentation, while a separate Mask R-CNN (He et al., 2018) or YOLO-based model (Redmon et al., 2016) is trained for object detection. If a user wishes to ask semantic questions about the imagery (e.g., “How many buildings have red roofs?”), an entirely different Visual Question Answering (VQA) model would be required. This approach is effective but inefficient, demanding separate training pipelines, data annotation formats, and model maintenance for each discrete task.

The recent emergence of large-scale, generalist Vision-Language Models (VLMs) promises a paradigm shift. Models such as PaLI (Chen et al., 2023) and others, pre-trained on web-scale image-text data, offer the potential to unify these disparate tasks through a single, promptable interface. However, a critical domain gap exists: these models are pre-trained on natural “in-the-wild” images (e.g., the WebLI dataset) and consequently lack the specific visual grammar and grounding to interpret overhead, high-resolution aerial imagery. As will be shown, their zero-shot performance on geospatial tasks is poor, confirming this domain gap. The most promising method for bridging the gap is instruction-tuning. This technique reframes all downstream tasks, including non-textual ones like segmentation, into a unified text-based “instruction→response” format (for example: “segment [image]” → “[mask]”). This allows a single VLM to learn a multitude of tasks simultaneously.

While this approach is gaining traction, the geospatial AI community is grappling with several known unknowns regarding its practical application. This paper positions itself not as a proponent of a new model architecture, but as a rigorous methodological investigation to answer the field’s most pressing questions about this emerging paradigm. The research is structured around three core questions:

- **(RQ1) Unification:** Can a single instruction-tuned VLM match or surpass strong single-task baselines built from the same backbone, while dramatically improving over the zero-shot model?
- **(RQ2) The Synergy Question**: Is there a benefit to multi-task learning? Specifically, does forcing a model to learn high-level semantic tasks (VQA, captioning) make it a better low-level localizer (segmentation, detection)? Or do these tasks interfere, creating a “master-of-none”?
- **(RQ3) The Efficiency Question**: How data-hungry is this adaptation? Given the high cost of manual geospatial annotation, what are the scaling laws for instruction-tuning a large VLM? Does it require thousands of samples, or can high performance be achieved with a modest, curated dataset?

To answer these questions, this paper presents a complete methodology for adapting Google’s PALIGEMMA2-mix model

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-857-2026

(Steiner et al., 2024) for unified building analysis. The primary contributions are:

- Data Conversion Pipeline: A novel and reproducible pipeline for converting unimodal vector data (building polygons) into a rich, four-task instruction-tuning dataset, leveraging Large Language Models (LLMs) for automatic semantic data generation (captions and VQA).
- Unified VLM Baseline: The fine-tuned PALIGEMMA2-mix model itself, which serves as a new, powerful baseline for multi-task building analysis, capable of segmentation, detection, VQA, and captioning within a single model.
- Synergy & Efficiency Study: A rigorous experimental study that provides, to our knowledge, the first clear answers to the three core research questions posed above – demonstrating multi-task synergy and quantifying data efficiency in this context.
- Dataset Release: The release of the multi-task instruction dataset (16,500 samples) derived from a new building annotation benchmark, which is introduced in a companion paper (Amrullah et al., 2026). The instruction-tuning dataset generated by our pipeline will be released publicly with the camera-ready version.

2. Related Work

2.1 Vision-Language Models in Remote Sensing

The application of vision-language models to remote sensing is a rapidly emerging field. Current approaches can be broadly categorized into two groups: (i) from-scratch pre-training and (ii) adaptation of generalist models.

From-scratch models are pre-trained partially or entirely on remote sensing-specific image-text datasets. Examples include GeoPixel (Shabbir et al., 2025) and GeoVLM (Dagda et al., 2025), which are designed for geospatial tasks from the ground up. GeoPixel, in particular, focuses on pixel-level grounding and referring expression segmentation, making it a strong baseline for building segmentation. Other works like SatCLIP (Klemmer et al., 2024) have also explored pre-training on satellite imagery. These models demonstrate the value of domain-specific pre-training but are computationally expensive to create.

Adaptation approaches, by contrast, take a powerful generalist VLM pre-trained on web-scale data and fine-tune it for remote sensing tasks. This leverages the immense world knowledge embedded in the base model and steers it toward the remote sensing domain. Our study falls into this adaptation category, using the state-of-the-art PALIGEMMA2 model (Steiner et al., 2024) as its foundation. Early results in this vein underscore the need to carefully bridge the domain gap when transferring to overhead imagery.

2.2 Instruction-Tuning for Geospatial Analysis

Instruction-tuning is the key methodology enabling the new paradigm of multi-task, promptable VLMs. Instead of training separate model heads or networks for each task, all tasks are converted into a unified text-in, text-out format. This general approach, popularized in works like InstructBLIP (Dai et al., 2023), has only very recently been applied to remote sensing. The MISTA dataset (Wu et al., 2024) demonstrated an automated pipeline using GPT-4 and LLaVA-1.5 to generate diverse instruction-following data from the DOTA aerial object detection benchmark (Xia et al., 2019). Other works have similarly constructed multi-task instruction datasets for tasks such as scene classification, VQA, and captioning (Wang et al., 2022; Sharma et al., 2025; Wang et al., 2023). Our work builds directly on this concept. However, while previous efforts focused on creating such datasets, our primary focus here is on systematically evaluating this methodology. In particular, this paper’s contribution is not just the dataset itself, but the scientific insights derived from its application – specifically regarding multi-task synergy and data-scaling properties in the geospatial context.

2.3 Multi-Task Learning for Building Analysis

Multi-task learning (MTL) in remote sensing (Wang and Others, 2022, 2020; Sharma et al., 2025) is not new. Researchers have long recognized that related tasks can benefit from shared representations. For example, several studies have combined building footprint segmentation with building boundary delineation or with change detection tasks. These conventional MTL approaches have shown moderate gains, mainly when tasks are closely related (e.g., two forms of localization).

The novelty of the VLM-based instruction-tuning approach lies in its ambitious scope. Prior MTL work typically combined related localization tasks. In contrast, our study investigates the synergy between fundamentally different task families: low-level geometric localization (segmentation, detection) and high-level semantic reasoning (VQA, captioning). We hypothesize that forcing a model to learn semantic context about a scene will, in turn, make it a more accurate and robust localizer. This goes beyond traditional MTL, testing whether understanding concepts like “dense residential area” or “L-shaped building” can regularize and improve the model’s pixel-level predictions.

3. Methodology: Unified Instruction-Tuning of PaliGemma2

The proposed methodology consists of three core components: (1) the foundation VLM, (2) the source data, and (3) the multi-task instruction conversion pipeline. We overview each in turn below.

3.1 Foundation Model: PaliGemma2-mix

The model selected for this investigation is PALIGEMMA2 (specifically, the “mix” variant). PALIGEMMA2 is a large Vision-Language Model explicitly designed for strong fine-tuning performance across a wide range of vision-language tasks. Architecturally, PALIGEMMA2 is composed of two powerful components: a SigLIP Vision Transformer image encoder and a Gemma 2 language model as the text decoder. In other words, it uses the SigLIP visual backbone (inspired by CLIP/ViT) paired with an advanced Gemma-2 transformer for language, enabling multimodal encoder–decoder capabilities. The model performs modality fusion via cross-attention in the decoder, allowing it to integrate visual and textual features for generation.

The PALIGEMMA2-mix variant was chosen for a specific reason: it is pre-trained on a massive mixture of tasks that explicitly includes localization tasks (object detection and

segmentation), as well as captioning and VQA. This means the base model already possesses latent capabilities for all four of our target tasks; it simply lacks the domain-specific knowledge to apply them to overhead imagery. In effect, PALIGEMMA2 comes “out of the box” with a powerful encoder–decoder architecture and multi-task skillset, making it an ideal starting point to test our adaptation methodology. We fine-tune this model with instruction-style supervision: during training, each input is an image with an instruction prefix (e.g., "segment") and the model must generate the correct textual output (mask encoding, answer, caption, etc.).

3.2 Data Foundation: The RoofVIP (Munich-Building) Benchmark

Our training data is derived from the RoofVIP benchmark (Amrullah et al., 2026). The dataset is based on manually double-verified, open-access very high-resolution (VHR) aerial images of Munich, Germany, which were tiled into 512×512 pixel chips for efficient GPU training. It provides RGB orthophotos paired with precise building footprints that are generated by merging roof-segment polygons into complete building outlines in 2D vector format. The data captures a broad range of urban typologies, including dense city centers, residential suburbs, and industrial zones, and features buildings of various sizes that often challenge reconstruction models.

For our experiments, we follow the official split defined by Amrullah et al. (2026), using $1,655$ tiles for training, 207 for validation, and 207 for testing (approximately $80/10/10\%$). This work focuses on the proposed instruction-tuning and model adaptation built on top of this dataset, rather than the dataset itself. Detailed information on the labeling protocol, data splits, and quality assessment is provided in the companion paper (Amrullah et al., 2026).

3.3 The Multi-Task Instruction Conversion Pipeline

This section details the core methodological contribution: a reproducible “recipe” for converting a standard, unimodal vector dataset (building polygons) into a rich four-task instruction-tuning dataset. Figure 1 provides an overview of the conversion pipeline, from input polygons to multi-task instructions.

The conversion process is organized by task type, with two tasks relying on direct geometric transformations and two tasks leveraging LLM-based generation:

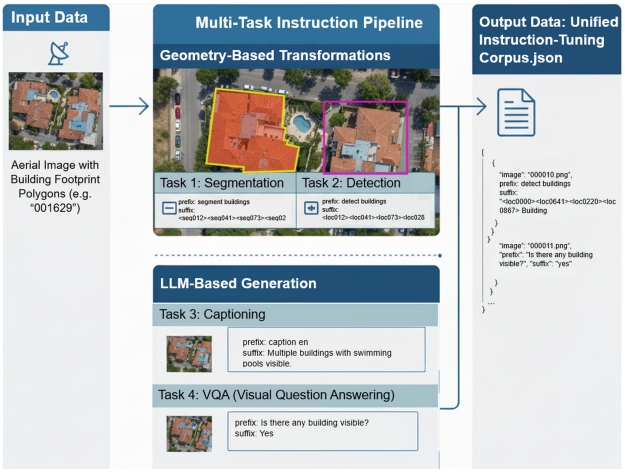

Figure 1. Multi-Task Instruction Conversion Pipeline

3.3.1 Task 1 & 2: Localization Instructions (Geometry–Based) –

The original building footprint polygons are programmatically converted into text representations suitable for instruction tuning. We formulate two localization tasks in text form:

- Task 1: Segmentation. We represent each building instance mask using discrete segmentation tokens derived from a vector-quantized variational autoencoder (VQ-VAE). This encoder is part of the PALIGEMMA segmentation module in Google’s `big_vision` repository. The process begins by extracting the polygon contours of each building and computing its axis-aligned bounding box, which is also used to generate normalized `<loc>` tokens, encoded as `<locYYYY>` using quantized coordinates $[y_{\min}, x_{\min}, y_{\max}, x_{\max}]$.

  The cropped mask region for each building is resized to a fixed 64×64 patch, which serves as input to the pre-trained VQ-VAE. The encoder outputs a 4×4 grid of quantized discrete codebook indices (ranging from 0 to 127), which are then mapped to token strings in the form `<seg000>` through `<seg127>`. Each building instance is thus represented by a fixed-length 16-token grid, preceded by four `<loc>` tokens identifying the bounding box.

  The final training instance is stored in instruction format, with the prefix `"segment buildings"` and a suffix composed of `<loc>` and `<seg>` tokens (Figure 2).

```
{
  "image": "001629.png",
  "prefix": "segment buildings",
  "suffix": "<loc0294><loc0042><loc0354><loc0068>
            <seg012><seg041><seg073><seg028>
            <seg059><seg028><seg091><seg125>
            <seg070><seg097><seg120><seg054>
            <seg012><seg078><seg025><seg003>
            Building"
}
```

Figure 2. Example of a segmentation task input. The suffix begins with location tokens for a bounding box, followed by a sequence of polygon segmentation tokens.

- Task 2: Detection. For object detection, we convert each building instance’s bounding box into four discrete spatial tokens. The bounding box is first extracted in the format $[y_1, x_1, y_2, x_2]$ from the original mask annotations. To ensure scale invariance across different image sizes, each coordinate is normalized by the image height (H) and width (W), yielding values in the continuous range $[0, 1]$.

  These normalized coordinates are then discretized into one of 1024 spatial bins by multiplying with 1023 and rounding to the nearest integer. The resulting integer values, ranging from 0 to 1023, are converted into location tokens of the form `<loc0000>` through `<loc1023>`.

  Each detection instance is formatted as an instruction, with the prefix `"detect buildings"` and the suffix containing the four `<loc>` tokens representing the bounding box corners (Figure 3).

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-857-2026 | © Author(s) 2026. CC BY 4.0 License.

```
{
  "image": "000242.png",
  "prefix": "detect buildings",
  "suffix": "<loc0434><loc0807><loc0541><loc0913>
             Building"
}
```

Figure 3. Example of a detection task input. The suffix contains location tokens defining a bounding box and the object class.

This representation allows the model to learn detection as a token prediction problem, aligned with the instruction-tuning framework used for other tasks.

These text encodings allow traditionally non-textual outputs (masks, boxes) to be learned in an encoder–decoder fashion.

3.3.2 Task 3 & 4: Semantic Instructions (LLM-Generated) – A key innovation of our pipeline is the use of external large language models (LLMs) to enrich the purely geometric dataset with high-level semantic content. This step addresses the lack of descriptive captions or questions in the original labels by generating them in a controlled way. We employ an instruction-tuned vision–language model (Qwen-VL 7B) to produce captioning and VQA supervision for each image.

While these LLM-generated labels greatly reduce manual annotation effort, they also introduce some degree of semantic noise (e.g., occasional mistakes in roof type or building use). In this work we treat them as weak supervision and rely on the large number of instances to average out individual errors, but a more systematic assessment of label quality and its impact on model performance is an important avenue for future work.

- **Task 3: Captioning.** For the captioning task, each aerial image is paired with a descriptive textual summary focusing on the built environment. Captions are generated using Qwen-VL conditioned with the prefix `caption en`, following the PALIGEMMA instruction format for English captioning. The system and user prompts guide the model to describe (i) the presence and rough count bucket of buildings, (ii) predominant building and roof types and approximate height (e.g., low-rise), (iii) spatial pattern and street orientation, and (iv) one notable feature of the urban context (e.g., parking lots, courtyards, vegetation). We request 3–4 concise sentences (10–40 words each) and discard outputs that violate basic length or formatting constraints. Each caption is stored as an instruction with the image identifier, the prefix `"caption en"`, and the generated text as suffix (Figure 4).

```
{
  "image": "000873.png",
  "prefix": "caption en",
  "suffix": "Many low-rise apartment blocks with
            flat roofs. Compact grid layout with
            N-S streets."
}
```

Figure 4. Example captioning instruction format, including the image reference, prefix, and descriptive suffix.

- **Task 4: Visual Question Answering (VQA).** To incorporate structured reasoning and semantic interpretation, we define a fixed-schema VQA task with five short questions per image, covering key attributes that can be inferred from overhead imagery (e.g., building presence, predominant use, dominant roof appearance, typical footprint shape, and visible construction activity). Each question is posed independently to Qwen-VL, which produces a concise textual response from a compact vocabulary (e.g., `yes/no`, `residential`, `tile`, `irregular`). The resulting instruction uses the question string as prefix and the answer as suffix, aligned with the instruction-tuning format. Each image thus yields five independent training instances (Figure 5). This design injects interpretable scene-level semantics without requiring manual question–answer annotation.

```
{"image": "002051.png",
 "prefix": "Is there any building visible?",
 "suffix": "yes"}
{"image": "002051.png",
 "prefix": "Predominant building use?",
 "suffix": "residential"}
{"image": "002051.png",
 "prefix": "Dominant roof appearance?",
 "suffix": "tile"}
```

Figure 5. Example VQA instruction format, including the image reference, prefix, and descriptive suffix.

Through the above steps, every original image and its polygon labels are transformed into a set of instruction–output pairs covering four tasks. The final step is unification.

3.3.3 Final Unification – After generating instruction-output pairs across the four task types—segmentation, detection, captioning, and VQA—we consolidate them into a unified instruction-tuning corpus. Each data point is stored as a separate entry in a `.jsonl` file, containing the image identifier, a task-specific prefix string, and the corresponding output (suffix). This format is compatible with autoregressive instruction-tuning and enables multi-task supervision in a single-stage model.

In total, our dataset comprises approximately 16,500 samples, averaging about 8 instruction pairs per image. These include one segmentation instance, one detection instance, 1–2 captions, and 2–3 VQA items. Tasks are balanced approximately uniformly but not strictly enforced per image. Each training sample is treated independently during fine-tuning.

The PALIGEMMA2 model is trained using a fully mixed task stream. Each input sequence begins with a task-specific prefix token such as `segment buildings`, `detect buildings`, `caption en`, or the natural-language question for VQA (e.g., `Is active construction visible?`). The model is trained to condition its output on these instructions, allowing it to implicitly learn task routing and output formatting within a single unified architecture.

This design enables PALIGEMMA2 to serve as a single multi-task geospatial model capable of zero-shot inference across diverse visual-linguistic tasks, without explicit task switching or architectural changes.

Table 1 summarizes the dataset composition and provides illustrative examples of instruction-output pairs across tasks.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-857-2026 | © Author(s) 2026. CC BY 4.0 License.

Task	# Samples	Example Instruction Pair (Prefix ⇒ Suffix)
Detection	4125	`detect buildings` ⇒ `<loc0434><loc0807><loc0541><loc0913> Building`
Segmentation	4125	`segment buildings` ⇒ `<loc0294>...<seg003> Building`
VQA	4125	`Is active construction visible?` ⇒ `no`
Captioning	4125	`caption en` ⇒ "Many low-rise buildings with flat roofs arranged in a compact grid. . . ."
Total	16 500	*(All tasks combined into a shared multi-task instruction set)*

Table 1. Instruction-Tuning Dataset Overview. Each row shows one task type, sample count, and an illustrative prefix–suffix pair used during training.

4. Experimental Setup

We now describe the experimental protocol used to answer the three research questions from Section 1. We first introduce the baselines, then the ablation settings, and finally the evaluation metrics.

4.1 Baseline Models

To quantify the impact of instruction-tuning, we compare our unified model to the underlying PALIGEMMA2 checkpoint in zero-shot mode. The zero-shot baseline is the public PALIGEMMA2-mix model used out-of-the-box, without any fine-tuning on our data; it is prompted with the same task prefixes and question strings as the fine-tuned models, but its parameters remain unchanged. This baseline measures how well a generic web-pretrained VLM transfers to nadir-view aerial imagery without adaptation.

Our main model is the multi-task PALIGEMMA2 variant trained jointly on all four tasks (segmentation, detection, captioning, VQA) using 100% of the instruction corpus, referred to as *Unified-100%*. In addition, we train several single-task specialist models based on the same backbone, described below. All models are evaluated on the same held-out test split of 207 images with identical preprocessing and prompts.

4.2 Ablation Settings

We consider two ablations: multi-task versus single-task training (Synergy, RQ2) and varying training set size (Efficiency, RQ3).

Multi-task vs. single-task learning. To assess synergy, we train four single-task specialists, each fine-tuned only on instructions for one task: **SegOnly** (segmentation, prefix `segment buildings`), **DetOnly** (detection, prefix `detect buildings`), **CapOnly** (captioning, prefix `caption en`), and **VQAOnly** (VQA, using the five fixed question types described in Section 3.3.2). Each specialist sees all instructions of its own task, but no supervision from the other three. We compare them to the Unified-100% model, which is trained on the full mixture of instructions across all four tasks, to measure how multi-task training affects per-task performance (Section 5.1).

Training set size (data efficiency). To study data efficiency, we train three unified models on progressively larger fractions of the instruction corpus: **Unified-25%**, **Unified-50%**, and **Unified-100%** (full set, 16,500 instances). Subsampling is performed at the instruction level while approximately preserving the relative proportions of segmentation, detection, captioning, and VQA. All three models are evaluated on the full test set to obtain learning curves as a function of training set size (Section 5.2).

4.3 Evaluation Metrics

We adopt standard metrics tailored to each task:

- **Segmentation.** We report Average Precision at IoU thresholds 0.50 and 0.75 (AP@0.50, AP@0.75), and the COCO-style averaged AP over IoU thresholds from 0.50 to 0.95 in steps of 0.05 (AP@[0.50:0.95]), computed at the building-instance level from the decoded segmentation masks.
- **Detection.** For bounding box prediction, we use the same AP@0.50, AP@0.75, and AP@[0.50:0.95] metrics, computed over boxes decoded from the `<loc>` tokens, making segmentation and detection directly comparable in terms of instance-level localization.
- **Captioning.** We evaluate caption quality with BLEU–4, METEOR, and ROUGE–L, computed between the generated captions and the reference captions described in Section 3.3.2. These metrics capture complementary aspects of n-gram overlap, semantic similarity, and sequence-level alignment.
- **VQA.** For the fixed-schema VQA task, we report accuracy: the fraction of questions for which the predicted answer string exactly matches the reference answer (after basic normalization). Given the constrained answer vocabulary (e.g., `yes/no`, roof type, footprint shape), exact-match accuracy is appropriate.

All metrics are computed on the same 207-image test set. For multi-task models, we evaluate each task by using the corresponding prefix or question type; for the zero-shot and single-task models, we apply the same evaluation scripts, restricting to the tasks they produce. This protocol allows direct comparison across zero-shot, single-task, and unified multi-task training regimes in Section 5.

5. Ablation Studies and Results

We now present the results of our experiments, organized to address the three guiding questions. Qualitative results and error analysis are also provided to illustrate the model's behavior (Figure 6).

5.1 Multi-Task vs. Single-Task Performance

To investigate the effects of multi-task learning (the Synergy Question), we compare the unified model trained jointly on all four tasks against four single-task specialist variants: *SegOnly*, *DetOnly*, *CapOnly*, and *VQAOnly*. All models use the same PALIGEMMA2 backbone and are trained on 100% of the available

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-857-2026 | © Author(s) 2026. CC BY 4.0 License.

	Segmentation			Detection			Captioning			VQA
Model	AP@0.50	AP@0.75	AP@[.5:.95]	AP@0.50	AP@0.75	AP@[.5:.95]	BLEU–4	METEOR	ROUGE–L	Accuracy
Multi-task (100%)	**0.485**	**0.342**	**0.318**	**0.566**	**0.382**	**0.369**	**0.566**	**0.705**	**0.709**	0.837
SegOnly	0.464	0.306	0.299	0.373	0.222	0.219	0.006	0.141	0.164	0.435
DetOnly	0.092	0.053	0.055	0.524	0.319	0.319	0.063	0.151	0.171	0.426
CapOnly	0.039	0.023	0.022	0.031	0.010	0.014	0.530	0.679	0.701	0.460
VQAOnly	0.024	0.005	0.010	0.024	0.005	0.010	0.007	0.147	0.168	**0.869**

Table 2. Multi-task vs. single-task training. We report AP metrics for segmentation and detection, BLEU–4 / METEOR / ROUGE–L for captioning, and accuracy for VQA. All models use the same backbone; the multi-task model uses all instructions jointly, while the single-task models only see task-specific instructions.

Figure 6. Qualitative multi-task results from the Unified-100% model. Each column shows one test scene, with the RGB tile, predicted segmentation mask, predicted detection boxes, generated caption, and answers to the five fixed VQA questions.

instructions for their respective task(s). Table 2 summarizes the results.

The multi-task model clearly outperforms the single-task variants on the **localization** tasks. For segmentation, AP@[0.50:0.95] increases from 0.299 for *SegOnly* to 0.318 for the multi-task model. For detection, the gain is even more pronounced: AP@[0.50:0.95] improves from 0.319 (*DetOnly*) to 0.369 with multi-task training. In contrast, models trained only on non-localization tasks (*CapOnly*, *VQAOnly*) exhibit almost no segmentation or detection capability, confirming that cross-task generalization does not emerge automatically without explicit supervision.

For **captioning**, the multi-task model also achieves the best scores across all three metrics. Compared to *CapOnly*, BLEU–4 improves from 0.530 to 0.566, METEOR from 0.679 to 0.705, and ROUGE–L from 0.701 to 0.709. This indicates that sharing representations with localization and VQA tasks helps the model generate captions that are both more faithful and more aligned with the reference descriptions.

The picture is slightly different for **VQA**. The *VQAOnly* specialist achieves the highest raw accuracy (0.869), while the multi-task model attains a slightly lower but still strong accuracy of 0.837. All other single-task models perform substantially worse on VQA (0.435–0.460). Thus, multi-task training incurs a modest trade-off of roughly 3 percentage points compared to a pure VQA specialist, but provides vastly better performance on all other tasks within a single model.

Overall, these results support a nuanced view of multi-task synergy. Joint training on segmentation, detection, captioning, and VQA substantially *improves* localization and captioning performance compared to single-task training, while maintaining competitive VQA accuracy. Rather than being a “jack-of-all-trades, master-of-none”, the multi-task model is strictly superior to specialists on three out of four tasks and only slightly behind the VQA-only model on the fourth. This provides strong evidence that instruction-tuned VLMs can benefit from cross-task supervision in geospatial settings, and that a single unified model can replace a collection of separate, single-purpose models in practice.

5.2 Impact of Training Set Size

To study the data efficiency of the unified model, we train the same architecture on three subsets of the instruction-tuning corpus: 25%, 50%, and 100% of the available samples. The subsets are created by random sampling at the instruction level while preserving the overall task distribution (segmentation, detection, captioning, VQA). Table 3 reports the full set of metrics for each task.

For **segmentation**, the unified model exhibits a clear but gradual improvement as more data is used. AP@0.50 increases from 0.457 (25%) to 0.485 (100%), while AP@0.75 increases from 0.282 to 0.342 and AP@[0.50:0.95] from 0.279 to 0.318. The 50% model already recovers most of the full-data performance (0.292 vs. 0.318 AP@[0.50:0.95]), with the remaining half of the instructions mainly providing refinements.

A similar trend is observed for **detection**. AP@0.50 improves from 0.533 (25%) to 0.551 (50%) and 0.566 (100%); AP@0.75 from 0.318 to 0.353 and 0.382; and AP@[0.50:0.95] from 0.320 to 0.347 and 0.369. Again, the model trained on 50% of the data is very close to the 100% model, indicating diminishing returns beyond roughly half of the instruction set.

For **captioning**, all three metrics show small but consistent gains as more data is used. BLEU–4 increases from 0.543

	Segmentation			Detection			Captioning			VQA
Training data	AP@0.50	AP@0.75	AP@[.5:.95]	AP@0.50	AP@0.75	AP@[.5:.95]	BLEU–4	METEOR	ROUGE–L	Accuracy
25%	0.457	0.282	0.279	0.533	0.318	0.320	0.543	0.694	**0.730**	**0.882**
50%	0.458	0.302	0.292	0.551	0.353	0.347	0.551	0.703	0.715	0.857
100%	**0.485**	**0.342**	**0.318**	**0.566**	**0.382**	**0.369**	**0.566**	0.705	0.709	0.837

Table 3. Impact of training set size on the unified model. We report AP metrics for segmentation and detection, BLEU–4 / METEOR / ROUGE–L for captioning, and accuracy for VQA.

to 0.566, METEOR from 0.694 to 0.705, and ROUGE–L is broadly stable (0.730 to 0.709). This suggests that the model quickly captures the main descriptive content and style of the captions, with additional data mainly improving phrasing and alignment with reference texts.

Interestingly, the semantic tasks exhibit a mild negative trend as the training fraction increases. **VQA** accuracy decreases slightly from 0.882 (25%) to 0.837 (100%), and ROUGE–L for captioning follows a similar pattern (from 0.730 to 0.705). We view these effects as indications of non-trivial task interactions rather than a failure of the unified model. On the one hand, adding more localization data changes the relative weighting of tasks in the multi-task objective, which may bias optimization towards improving segmentation and detection at the expense of small fluctuations on semantic metrics. On the other hand, the caption and VQA labels are generated automatically by Qwen-VL and therefore contain some amount of semantic noise; as more synthetic data is added, the model may also fit this noise. A more fine-grained analysis of task balancing and label quality is left for future work, but we emphasize that all three settings (25%, 50%, 100%) yield strong VQA and captioning performance.

Overall, these results indicate that the unified model is *data-efficient*: training with only 50% of the instruction data is sufficient to recover the large majority of the performance of the full-data model across segmentation, detection, and captioning, while VQA accuracy remains strong even with 25% of the data. This is encouraging given the cost of manual geospatial annotation.

5.3 Comparison with Public Baselines

As an external reference, we compare our unified model to the underlying PaliGemma2-mix checkpoint used in zero-shot mode, i.e. without any fine-tuning on our geospatial data. This setup directly quantifies the overhead-imagery domain gap and the effect of instruction-tuning. Table 4 reports the full set of metrics for the fine-tuned unified model (100% training data) and the zero-shot baseline.

Model	**Seg. AP**	**Det. AP**	**Cap. R-L**	**VQA Acc.**
`Ours-Unified-100%`	**0.318**	**0.369**	**0.709**	**0.837**
`PaliGemma2 zero-shot`	0.017	0.017	0.161	0.490

Table 4. Main quantitative results vs. a zero-shot PaliGemma2 baseline. We report the main summary metric for each task: AP@[0.50:0.95] for segmentation and detection, ROUGE–L for captioning, and accuracy for VQA.

The gap between the zero-shot and fine-tuned models is substantial across all tasks. For **segmentation**, AP@[0.50:0.95] increases from 0.017 to 0.318, i.e. almost a twenty-fold improvement. A similar pattern holds for **detection**, where AP@[0.50:0.95] rises from 0.017 to 0.369. These near-zero scores in the zero-shot setting confirm that a web-pretrained VLM, applied as-is to nadir-view aerial imagery, does not reliably localize buildings and thus cannot be used directly for building analysis.

For **captioning**, the BLEU–4 score improves from 0.007 to 0.566 after fine-tuning, while METEOR increases from 0.154 to 0.705 and ROUGE–L from 0.161 to 0.709. This indicates that the zero-shot model produces largely uninformative or misaligned captions in the geospatial domain, whereas the instruction-tuned model generates detailed, structurally consistent descriptions of the built environment.

The **VQA** task also benefits markedly from adaptation. Accuracy improves from 0.490 for the zero-shot baseline to 0.837 for the unified model, a gain of more than 34 percentage points. In other words, the pre-trained VLM is only marginally better than a weak heuristic on our five fixed questions, while the fine-tuned model answers them reliably.

Overall, Table 4 provides a clear, quantitative answer to the unification and adaptation question: a single instruction-tuned PaliGemma2 model, trained on our multi-task instruction corpus, not only unifies segmentation, detection, captioning, and VQA, but also transforms an essentially unusable zero-shot model into a strong geospatial building analyzer. This underscores both the severity of the overhead-imagery domain gap and the effectiveness of the proposed instruction-tuning pipeline.

6. Conclusions

This paper presented a systematic study of adapting a generalist Vision-Language Model (VLM) (PALIGEMMA2) for unified, multi-task geospatial building analysis. We framed the work around three questions and can now answer them based on our empirical results.

- **Can a single VLM unify these tasks? Yes.** A single instruction-tuned PALIGEMMA2 model, trained on our multi-task instruction corpus, performs building segmentation, detection, captioning, and VQA within one architecture. Compared to the zero-shot checkpoint, the unified model yields large gains across all tasks (e.g., AP@[0.50:0.95] for both segmentation and detection improves from 0.017 to 0.318 and 0.369, respectively, and VQA accuracy from 0.49 to 0.84), turning an essentially unusable baseline into a strong geospatial building analyzer.
- **Is there a synergistic benefit to multi-task learning? Mostly yes.** Joint training on all four tasks leads to clear improvements over single-task specialists on segmentation, detection, and captioning. For example,

segmentation AP@[0.50:0.95] increases from 0.299 (SegOnly) to 0.318 in the multi-task model, detection AP@[0.50:0.95] from 0.319 (DetOnly) to 0.369, and caption METEOR from 0.679 (CapOnly) to 0.705. VQA-only training reaches the highest raw VQA accuracy (0.869), but the multi-task model remains competitive (0.837) while being significantly stronger on all other tasks. Rather than a “jack-of-all-trades, master-of-none”, the unified model is superior on three out of four tasks and only slightly behind the VQA specialist on the fourth.

- **Is the adaptation data-efficient? Yes.** Our scaling experiment with 25%, 50%, and 100% of the instruction data shows that performance improves with more data, but exhibits diminishing returns. Using only 50% of the instructions recovers most of the full-data performance for segmentation and detection, while captioning and VQA remain strong even with 25%. This indicates that a few thousand high-quality, multi-task instructions are sufficient to reach a strong regime, which is encouraging given the cost of geospatial annotation.

Taken together, these findings support a data-centric view of geospatial Artificial Intelligence (AI). Instead of designing a separate model for each downstream task, it becomes feasible to steer a single foundation model toward geospatial use cases by curating an appropriate multi-task instruction dataset. In our setting, a single adapted PALIGEMMA2 model replaces a collection of task-specific models, simplifies the deployment stack, and enables richer interactions (e.g., querying the same model for both pixel-level outputs and semantic descriptions).

Future work will extend this methodology beyond building analysis. We plan to investigate additional tasks such as change detection and damage assessment, other domains (e.g., agriculture, maritime monitoring), and additional modalities (e.g., Synthetic Aperture Radar (SAR) or multi-temporal imagery). Another avenue is to move beyond single-turn instructions toward multi-turn dialogue and interactive querying of remote sensing scenes. We hope that the released masks and instruction-tuning data will serve as a starting point for further exploration of unified, instruction-tuned VLMs in the remote sensing community.

References

Amrullah, C., Panangian, D., Mutreja, G., Abdelhedi, Y., Bittner, K., 2026. Roofvip benchmark dataset: 2d roof planar polygons and very high-resolution digital orthophoto pairs. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A., Bradbury, J., Kuo, W., Seyedhosseini, M., Jia, C., Ayan, B. K., Riquelme, C., Steiner, A., Angelova, A., Zhai, X., Houlsby, N., Soricut, R., 2023. Pali: A jointly-scaled multilingual language-image model.

Dagda, B., Awais, M., Fallah, S., 2025. Geovlm: Improving automated vehicle geolocalisation using vision-language matching.

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S., 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2018. Mask r-cnn.

Klemmer, K., Rolf, E., Robinson, C., Mackey, L., Rußwurm, M., 2024. Satclip: Global, general-purpose location embeddings with satellite imagery.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation.

Shabbir, A., Zumri, M., Bennamoun, M., Khan, F. S., Khan, S., 2025. Geopixel: Pixel grounding large multimodal model in remote sensing.

Sharma, R., Chen, C., Chang, F.-J., 2025. Multi-modal multi-task unified embedding model (m3t-uem): A task-adaptive representation learning framework. *Proceedings of ICCV*.

Steiner, A., Pinto, A. S., Tschannen, M., Keysers, D., Wang, X., Bitton, Y., Gritsenko, A., Minderer, M., Sherbondy, A., Long, S., Qin, S., Ingle, R., Bugliarello, E., Kazemzadeh, S., Mesnard, T., Alabdulmohsin, I., Beyer, L., Zhai, X., 2024. Paligemma 2: A family of versatile vlms for transfer.

Wang, J., Zhang, X., Liu, Z., Wu, J., Xu, Y., 2022. Rs-llava: A large vision-language model for joint captioning and question answering in remote sensing imagery. *arXiv preprint arXiv:2207.03462*.

Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., Kang, J., Yang, J., Li, S., Du, C., 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Wang, Y. D., Others, 2022. ChangeMinds: Multi-task Framework for Detecting and Describing Changes in Remote Sensing. *arXiv preprint arXiv:2212.XXXX*. https://arxiv.org/abs/2212.XXXX.

Wang, Y., Others, 2020. Boundary-Aware Multitask Learning for Remote Sensing: Semantic Segmentation, Height Estimation, and Boundary Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 58, 1-14.

Wu, H., Lu, K., Li, Y., Huang, J., Xue, J., 2024. Mista: A large-scale dataset for multi-modal instruction tuning on aerial images. *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2019. Dota: A large-scale dataset for object detection in aerial images.