

VectorLLM++: A Unified Next-Token-Prediction MLLM for Dense Remote-Sensing Perception and Vector Mapping

1 Details of the GeoPolygon dataset

GeoPolygon contains two complementary splits: an automatically generated train split and a manually refined test split. The train split is produced by the proposed vector-instruction data engine and is used to provide large-scale supervision for dense polygon extraction. It emphasizes scale, target-phrase coverage, and object-level geometric supervision. The test split, referred to as the GeoPolygon benchmark, is manually refined and is used only for evaluation. It emphasizes annotation accuracy, category consistency, and reliable polygon-level assessment. This section first describes the construction and statistics of the train split, and then provides details of the manually refined test split.

1.1 Details of GeoPolygon train split

Dataset composition. The GeoPolygon train split is generated by the vector-instruction data engine and is organized as single-object polygon extraction dialogues. It contains 1,213,855 samples, each comprising an object-centric remote sensing image, a natural-language instruction that specifies the target object, and a polygon response encoded as a coordinate-token sequence. These object-centric samples are derived from 5,739 original remote sensing images sampled from 10 source datasets: DIOR [1], DOTA v2 [2], RSOD [3], FAIR1M v2 [4], NWPU [5], WHU_GID [6], LEVIR [7], SIMD [8], UCAS_AOD [9], and HRSC2016 [10]. It covers 1,834 target phrases, providing broad supervision for polygon extraction across diverse object categories. Based on source tags in the image file names, 672,554 samples are generated from VectorLLM candidates and 541,301 samples are generated from SAM [11] mask candidates. Table 1 summarizes the source-dataset composition of the train split.

Instruction and response format. All samples follow the same one-turn instruction-following format. The user instruction asks the model to extract the vector contour of the central target object in the image and output it in clockwise order starting from the upper-left corner. The assistant response is a JSON-style list containing the target phrase and its polygon. The polygon is represented between `<poly_start>` and `<poly_end>` using alternating discrete coordinate tokens, e.g., `<x*>` and `<y*>`.

Table 1: Source-dataset composition of the GeoPolygon train split.

Source dataset	Dialogue samples	Original images
DIOR [1]	401,860	2,045
DOTAv2 [2]	339,228	1,398
RSOD [3]	158,721	403
FAIR1Mv2 [4]	158,598	781
NWPU [5]	69,752	181
WHU_GID [6]	51,476	740
LEVIR [7]	19,406	91
SIMD [8]	8,751	57
UCAS_AOD [9]	5,847	41
HRSC2016 [10]	216	2
Total	1,213,855	5,739

Table 2: Summary statistics of the GeoPolygon train split.

Property	Value
Dialogue samples	1,213,855
Polygon annotations	1,213,855
Unique target phrases	1,834
VectorLLM candidate samples	672,554
SAM mask candidate samples	541,301
Image width range	16–1000 pixels
Image height range	16–1000 pixels
Median width / height	48 / 50 pixels
Coordinate token range	0–1023

This format converts polygon extraction into an autoregressive token prediction problem while retaining an explicit vector representation of object boundaries.

Target phrase distribution. The train split covers 1,834 raw target phrases. These phrases preserve the wording used in the generated instruction data, so semantically related forms such as singular and plural nouns can appear as separate phrases. The most frequent phrases mainly correspond to common remote sensing objects or land-cover elements, such as trees, buildings, roofs, vehicles, grass, roads, fields, and water. Table 3 lists the 20 most frequent target phrases in the train split.

Polygon complexity. The train split provides simplified yet explicit vector contours. The number of vertices per polygon ranges from 4 to 291, with a median of 6 and a mean of 7.09. The 75th, 90th, and 95th percentiles are 8, 10, and 11 vertices, respectively. This distribution indicates that most training polygons are compact vector descriptions, while the long tail accounts for more complex contours of irregularly shaped objects. These annotations expose the model to both simple geometric primitives and more detailed object boundaries.

Role in model training. The GeoPolygon train split complements box- and mask-based supervision by training the model to generate object contours directly as

Table 3: Top 20 target phrases in the GeoPolygon train split.

Target phrase	Samples	Target phrase	Samples
tree	120,336	road	28,007
building	82,235	buildings	27,688
roof	80,549	vehicles	23,267
trees	69,740	lawn	20,375
vehicle	68,911	water	17,193
grass	62,652	boat	15,572
car	48,745	structure	12,862
vegetation	36,352	shadow	11,656
house	30,357	shrub	10,899
field	28,740	dirt	10,637

ordered vector tokens. Because the samples are formulated as instruction-following dialogues, the same next-token prediction objective applies to language responses and polygon outputs. This design allows VectorLLM++ to learn polygon extraction alongside mask segmentation within the unified MLLM training framework, without requiring a task-specific prediction head for vector contours.

1.2 Details of GeoPolygon test split

The GeoPolygon benchmark is a manually refined dataset for evaluating dense polygon extraction in remote sensing imagery. It assesses models’ ability to generate object-level vector contours aligned with human delineation in multi-category remote sensing scenes. The benchmark contains 1,452 image-level samples, covering 18 object categories and 8,820 polygon instances. The following sections describe the data sources, category distribution, annotation process, and manual refinement procedure of the GeoPolygon benchmark.

Data sources and category distribution. Images in GeoPolygon benchmark are mainly sampled from the evaluation splits of multiple public remote sensing datasets for which no polygon annotations were available, with the soccer-field category additionally collected from Google Earth. The benchmark covers 18 categories, including agriculture, aircraft, baseball-field, basketball-court, buildings, chimney, dam, playground, roundabout, ship, soccer-field, stadium, storage-tank, swimming-pool, tennis-court, vehicle, water, and windmill. Image resolutions range from 256×256 to 1024×1024 , covering diverse object scales, imaging conditions, background complexity, and scene layouts.

GeoPolygon benchmark prioritizes explicit object-level geometric contour representation. For each valid object instance, the benchmark provides a corresponding polygon annotation, allowing evaluation to be conducted directly based on polygon-level IoU between predicted contours and ground-truth contours. This annotation form evaluates not only whether the object is correctly localized, but also whether the generated contour accurately captures object shape, boundary details, and geometric structure. Table 4 summarizes the category distribution of GeoPolygon benchmark, and Fig. 1 and 2 show representative visual examples.

Table 4: Category distribution and data sources of GeoPolygon benchmark.

Category	Data Source	Number of Images	Number of Polygons
Agriculture	FGFD [12]	56	522
Aircraft	DIOR [1]	63	425
Baseball-field	DIOR	100	438
Basketball-court	DIOR	100	433
Buildings	WHU-Mix [13]	100	1122
Chimney	DIOR	100	270
Dam	DIOR	87	94
Playground	DIOR	100	240
Roundabout	MLRSNet [14]	99	99
Ship	DIOR	53	522
Soccer-field	Google Earth	43	152
Stadium	DIOR	97	112
Storage-tank	DIOR	98	2812
Swimming-pool	BHP-Pools [15]	100	462
Tennis-court	DIOR	81	537
Vehicle	DIOR	56	322
Water	OpenStreetMap	49	78
Windmill	DIOR	70	180
Total	–	1452	8820

Annotation procedure. To balance efficiency and annotation quality, the GeoPolygon benchmark adopts a semi-automatic construction pipeline that combines automatic coarse annotation with manual refinement. First, we apply the same data engine used for the training split to generate initial coarse annotations. We then develop a dedicated vector refinement annotation tool for manual correction of the automatically generated polygons. Using this tool, annotators adjust contour boundaries, remove incorrect instances, add missing objects, and correct category labels, refining coarse annotations into high-quality ground-truth polygons. Specifically, the manual refinement process focuses on three aspects. First, annotators verify instance completeness, ensuring that all recognizable objects in the evaluation categories are annotated. Second, they remove redundant polygons caused by shadows, textures, backgrounds, or false candidate regions. Third, they correct deviations to better align with object shapes and human delineation standards. Through this semi-automatic pipeline and manual refinement, GeoPolygon provides high-quality ground truth for dense polygon extraction, enabling evaluation of object localization, contour generation, and geometric alignment.



Fig. 1: Representative visual examples of GeoPolygon benchmark (part 1).



Fig. 2: Representative visual examples of GeoPolygon benchmark (part 2).

2 Dense object detection data details

To support dense object detection training in remote sensing, we curated 17 existing remote sensing object detection datasets, including UCAS-AOD [9], NWPU VHR-10 [5], DIOR [1], LEVIR [7], RSOD [3], DOTA-v2 [2], HRSC2016 [10], FAIR1M-v2 [4], SIMD [8], VEDAI [16], CrowdAI [17], AirbusShip [18], SODA [19], ShipRSImageNet [20], AirplaneDet [21], WHU [22], and WHUMix [13]. These datasets cover a wide range of typical remote sensing targets—such as vehicles, ships, aircraft, storage tanks, harbors, bridges, and sports fields—and include both horizontal and oriented bounding box annotations. However, they exhibit significant heterogeneity in image dimensions, annotation formats, category definitions, and label quality, making direct combination for unified training challenging. To address this, we designed a unified pipeline with three steps: category preprocessing, data unification and quality enhancement, and conversion to the final conversation-style training format. The details of each step are described below.

2.1 Category preprocessing

Different datasets often adopt inconsistent category granularity, particularly for transportation-related objects. For example, some datasets annotate coarse categories such as vehicle, aircraft, and ship, while others provide fine-grained labels such as van, truck, car, plane, helicopter, and boat. To meet the requirement of dense detection training, we first unified transportation-related categories by mapping fine-grained labels for vehicles and aircrafts to their corresponding parent classes. Specifically, car was merged into vehicle, and plane into aircraft. After this coarse-grained alignment, the dataset contained 40 categories. We then further consolidated the remaining categories by merging semantically equivalent labels (e.g., storage-tank, storagetank, and storage tank). Finally, the category vocabulary was reduced from 40 categories to 26 standardized categories, as detailed in Table 5.

2.2 Data unification and quality enhancement

In addition to category inconsistency, the collected detection datasets also differ significantly in annotation formats, image sizes, and annotation quality. We converted all datasets into a unified bounding box based intermediate annotation format. This intermediate format records the image path, image size, category name, box coordinates, and dataset source for each instance. For datasets annotated with oriented bounding boxes (OBBs), we converted each OBB into its minimum enclosing bounding box and filled it into the same intermediate format.

Image size inconsistency poses another major challenge for remote sensing detection data. The original images range from small fixed-size patches to large-scale scenes, making direct training computationally inefficient and unstable. A fixed cropping strategy risks cutting through bounding boxes. Instead, our algorithm dynamically selects crop boundaries within a range of 768–1280 pixels per spatial dimension. For each dimension, we evaluate candidate boundaries within this range and select the one that minimizes the number of cut objects while keeping the crop size as close as possible to 1024 pixels. If the selected boundary still intersects a few objects, we discard the

Table 5: Reclassification mapping of object detection categories.

Standard Category	Original Categories
vehicle	vehicle
ship	ship
aircraft	aircraft
storage-tank	storage-tank, storagetank, storage tank, oiltank
tennis-court	tennis-court, tenniscourt, tennis court
harbor	harbor
bridge	bridge
baseball-field	baseballfield, baseball field, baseball-diamond, baseball diamond
basketball-court	basketball court, basketballcourt, basketball-court
swimming-pool	swimming-pool
windmill	windmill
ground-track-field	groundtrackfield, ground-track-field, ground track field
overpass	overpass
soccer-field	football field, soccer-ball-field
expressway-service-area	expressway-service-area
airport	airport
roundabout	roundabout
chimney	chimney
expressway-toll-station	expressway-toll-station
stadium	stadium
dam	dam
trainstation	trainstation
golffield	golffield
playground	playground
container-crane	container-crane
helipad	helipad

affected annotations and keep the intact ones. In addition, we apply random rotation to the cropped sub-images as data augmentation to increase orientation diversity.

After cropping, we further inspected the annotation quality and found that some datasets suffered from various types of noise, including missing objects, inaccurate localization, and oversized bounding boxes. Such low-quality annotations could introduce biased supervision signals and lead the MLLM to learn incorrect object-location associations. We therefore performed SAM3 [23] assisted relabeling, apply both text prompts and visual prompts in parallel to recover missing objects. Newly detected boxes were added to the original annotations, while valid existing annotations were preserved.

After the above processing, we obtained a unified dense object detection dataset containing 444,491 images and 4,497,766 bounding boxes in 443,872 valid images in total. Detailed statistics for each dataset are provided in Table 6.

Table 6: Statistics of processed object detection datasets.

Dataset	Total Images	Valid Images	Instances	Avg. Density
UCAS-AOD [9]	2,950	2,942	26,013	8.84
NWPU VHR-10 [5]	779	779	5,111	6.56
DIOR [1]	11,717	11,717	67,636	5.77
LEVIR [7]	2,870	2,422	9,487	3.92
RSOD [3]	1,867	1,867	14,917	7.99
DOTAv2 [2]	9,866	9,843	251,684	25.57
HRSC2016 [10]	1,105	1,105	4,194	3.80
FAIR1M-v2 [4]	24,176	24,176	545,922	22.58
SIMD [8]	3,991	3,947	59,698	15.12
VEDAI [16]	938	842	3,001	3.56
CrowdAI [17]	280,741	280,741	2,395,553	8.53
AirbusShip [18]	41,136	41,136	78,551	1.91
SODA [19]	11,759	11,759	363,245	30.89
ShipRSImageNet [20]	3,322	3,322	15,718	4.73
AirplaneDet [21]	1,056	1,056	5,474	5.18
WHU [22]	2,604	2,604	143,252	55.01
WHUMix [13]	43,614	43,614	508,310	11.65
Total	444,491	443,872	4,497,766	10.13

2.3 Conversion to conversation format

After processing the bounding box annotations, we converted the intermediate detection data into an instruction-following conversation format for MLLM training. Each bounding box coordinate is normalized and discretized onto a $[0, 1023]$ integer grid. The four coordinates of each box are then represented by detection-specific coordinate tokens, i.e., $\langle x000 \rangle - \langle x1023 \rangle$ and $\langle y000 \rangle - \langle y1023 \rangle$, forming a vocabulary of 2048 spatial tokens. This discretized representation allows the model to generate bounding boxes as textual token sequences while preserving spatial precision.

Each dialogue turn targets a single category within an image. The user instruction asks the model to detect all objects of a specified category, and the assistant response provides the complete list of bounding boxes. For images with multiple categories, we construct a multi-turn dialogue, where each turn addresses one category. The dialogue template is defined as follows:

Human: `<image>\n Please detect all objects belonging to [category] in the image.`
Assistant: `{"category": "[category]", "bboxes": ["bbox_1", "bbox_2", ...]}`.

3 Dense segmentation data details

To support dense segmentation training with discrete mask tokens, we construct a unified segmentation corpus from heterogeneous segmentation sources. The details are described below.

3.1 Data composition

The dense segmentation corpus contains 27 existing remote sensing segmentation datasets and 5 close-range datasets. These sources cover semantic, instance, panoptic, and referring segmentation, providing supervision for both category-conditioned mask prediction and expression-conditioned mask grounding. The remote sensing datasets span diverse scene types and spatial resolutions (0.05 m to 30 m), including both land-cover regions and object-level targets such as buildings, roads, water bodies, vehicles, storage tanks, aircraft, ships, farmland, and vegetation.

To handle different image sizes, label names, and annotation formats, we first organize all sources into a common intermediate representation. Each record stores the image path, dimensions, target phrase (or category name), and associated binary mask(s). To maintain a consistent training vocabulary, we unify the segmentation taxonomy into eight primary classes: building, water, vegetation, man-made surface, vehicle, structure, farmland, and barren. The 71 level-2 labels from the source datasets are mapped to these classes (Table 8). After preprocessing, the final corpus contains 3,598,350 image-level samples and 8,470,441 mask instances (Table 7).

3.2 Unified segmentation via mask tokenization

The collected datasets define segmentation targets in different forms. Semantic segmentation datasets provide category-level region maps; instance segmentation provide per-object masks; panoptic segmentation combines stuff regions and thing instances; and referring segmentation associate a natural-language expression with a target mask. We convert all of these supervision signals into phrase-conditioned binary mask prediction. For semantic segmentation, all pixels of a queried category are merged into a single binary mask. For instance and panoptic segmentation, each thing instance is preserved as an individual mask, while stuff regions are represented as category-conditioned region masks. For referring segmentation, the expression itself conditions the mask output.

Mask tokenization provides a unified output interface: each mask is encoded into a short sequence of discrete tokens delimited by `<mt_start>` and `<mt_end>`. In the final training data, a mask is serialized as a textual token span such as `<mt_start><mt_0159><mt_0261><mt_0633><mt_1005><mt_end>`. The assistant response is then organized as a JSON-style list of phrase-mask records, e.g., `[{"phrase": "building", "mask": "..."}]`. When multiple masks correspond to the same phrase, the mask field contains multiple mask-token spans. This design removes task-specific output heads and unifies semantic, instance, panoptic, and referring segmentation under a single autoregressive objective.

3.3 Conversation format conversion

We crop large images into patches of at most 1024×1024 pixels before constructing conversations. Images exceeding this size are divided into non-overlapping patches, with optional overlap only near the final edge to ensure complete coverage. Annotations are projected to the cropped image coordinates, converted to binary masks, and stored as Run-Length Encoding (RLE) before mask tokenization.

Each training sample is formatted as a one-turn image-grounded dialogue. The human message contains the image placeholder followed by a query for the mask of a target category, object type, or referring expression. The assistant message is a JSON-style list with phrase and mask fields per entry. This format follows the same instruction-following interface as the dense detection and polygon extraction data, but uses discrete mask tokens instead of box or polygon coordinate tokens:

Human: <image>\n Segment with one mask: [phrase]

Assistant: [{"phrase": "[phrase]", "mask": "<mt_start><mt_****>...<mt_end>"}].

Table 7: Segmentation datasets used for VectorLLM++ training. The table reports the number of training images, image-size range after preprocessing, and total mask instances in the final conversation-style data.

Datasets	Images	Image Size	Instances
CHN6-CUG [24]	2,533	512×512	2,533
DeepGlobe Road [25]	6,226	1024×1024	6,226
Ottawa [26]	327	1024×1024	327
FineGrip [27]	2,496	$590\text{--}1071 \times 563\text{--}944$	6,911
LaSeRS [28]	10,000	$600\text{--}1024 \times 600\text{--}1024$	13,156
OpenEarthMap [29]	62,498	512×512	62,498
BSB [30]	9,430	512×512	71,047
NWPU [5]	717	$533\text{--}1728 \times 358\text{--}1028$	3,178
SIOR [31]	17,253	800×800	54,998
LRSNY [32]	602	1000×1000	602
WHU Mix [13]	42,540	$300\text{--}650 \times 300\text{--}650$	475,012
CASID [33]	90,313	1024×1024	377,597
LandCoverAI [34]	15,350	512×512	15,350
ISPRS Potsdam [35]	13,021	512×512	13,021
DLRSD [36]	6,415	512×512	6,415
DeepGlobe Land Cover [25]	35,128	512×512	35,128
FLAIR [37]	292,586	512×512	292,586
LoveDA [38]	28,300	512×512	28,300
SOTA [31]	30,017	1024×1024	327,494
FAST [31]	157,814	600×600	796,978
Globe230k [39]	323,434	512×512	323,434
RISBench [40]	26,283	512×512	26,283
RRSISD [41]	12,179	$800 \times 784\text{--}802$	12,179
RefSegRS [42]	2,172	512×512	2,172
CITY-OSM [43]	47,507	1024×1024	317,040
CrowdAI [17]	227,809	300×300	1,901,296
EVLab-SS [44]	13,117	512×512	13,117
COCONUT-Base [45]	1,597,688	$59\text{--}640 \times 51\text{--}640$	2,648,443
PADT-RefCOCO [46]	321,327	$128\text{--}640 \times 104\text{--}640$	321,327
RefCOCOg [47]	190,204	$164\text{--}640 \times 146\text{--}640$	263,753
UDD5 [48]	6,032	1024×1024	28,121
UDD6 [48]	7,032	1024×1024	23,919
Total	3,598,350	—	8,470,441

Table 8: Reclassification mapping from level-2 labels to unified primary classes. Source datasets use heterogeneous category names and granularities; we group semantically related labels into eight primary classes to construct a consistent dense segmentation training vocabulary.

Primary Class	Secondary Classes
Building	facade, roof, commercial building, residential building, house, small construction, etc.
Water	swimming pool, lake
Vegetation	low vegetation, tree, forest, rangeland, woodland, grassland, grass
Man-made Surface	road, bridge, railway, parking apron, hardstand, runway, bikeway, side-walk, etc.
Vehicle	airplane, ship, car, van, truck, trailer, bus, small-vehicle, large-vehicle, plane, helicopter
Structure	chimney, storagetank, dam, windmill, storage tank, container-crane
Farmland	agriculture land, agriculture
Barren	bare soil, bareland

4 In-context data construction details

This section provides implementation and statistical details of our In-context Learning (ICL) dataset. The training samples are derived from our established dense detection dialogue data and restructured to enable few-shot visual prompting—activating the model’s detection capability for novel categories from one or a few exemplars. We focus on mitigating repetition bias and ensuring spatial consistency in dense-instance scenarios.

4.1 Strategy for mitigating repetition bias

A critical challenge in training MLLMs for ICL is repetition bias: models tend to replicate provided prompt coordinates rather than performing actual visual retrieval. We addressed this through two primary mechanisms. One is dynamic sampling. For a category with N instances, we sample k prompt boxes as visual cues, where $k \in [1, \max(1, \lfloor N \times 0.3 \rfloor)]$. This upper bound ensures that the model actively searches for at least 70% of targets rather than relying solely on provided examples. The second mechanism introduces controlled null-retrieval samples. We included a 5% subset of single-target dialogues, where the prompt box already covers the only instance of that category. In these samples, the correct response is to terminate generation rather than output additional boxes, teaching the model to stop when no further objects of the prompted category exist.

4.2 Dataset statistics

The final ICL dataset comprises 47,175 unique images and 54,786 dialogue turns. During the instruction-tuning phase, the assistant outputs all bounding boxes for the requested category. As summarized in Table 8, the dataset contains 938,879 instances in the responses, of which 774,852 are newly detected instances, i.e., beyond the provided prompts.

4.3 Sequence ordering and dialogue template

To enable stable long-sequence generation in dense-instance scenarios, all bounding boxes are normalized to a $[0, 1023]$ discrete integer grid, following the same scheme used for object detection data. We further apply a patch-based sorting strategy (with a factor of 28) to the output sequences. We compute a linear index for each box from its center point within a 28×28 grid, enforcing a consistent top-to-bottom, left-to-right spatial ordering. This mechanism significantly reduces cumulative error in autoregressive generation and minimizes hallucinations for dense objects. The data are organized into a multi-turn conversation format. Each turn focuses on a single category to reinforce class-specific retrieval:

```
Human:      <image>\nPlease detect all objects belonging to the same  
category as the boxes [<prompt_boxes>] in the image.  
Assistant:  {"category": "objects", "bboxes": ["<prompt_boxes>",  
"<new_boxes>"]}
```

Table 9: Detailed statistics of the in-context learning dataset.

Source Dataset	Total Images	Dialogue Turns	Prompt Boxes	Newly Detected
DIOR [1]	7,223	9,186	13,709	45,843
DOTAv2 [2]	8,182	11,914	41,805	205,884
FAIR1Mv2 [4]	21,450	23,020	90,831	450,265
HRSC2016 [10]	856	856	978	2,967
LEVIR [7]	1,767	1,793	2,220	6,593
NWPU [5]	640	733	1,030	3,718
RSOD [10]	1,228	1,228	2,468	11,367
SIMD [8]	2,265	2,476	5,384	25,053
UCAS_AOD [9]	2,942	2,942	4,892	21,121
VEDAI [16]	622	638	710	2,041
Total	47,175	54,786	164,027	774,852

5 Metrics and benchmarks for seven tasks

5.1 Dense detection

Metrics. For dense detection, we adopt a confidence-agnostic evaluation protocol based on F1-score. Unlike conventional COCO-style mAP—which relies on confidence scores for ranking—our model output bounding boxes as discrete coordinate tokens without producing calibrated detection confidences. Therefore, predictions are evaluated as hard box outputs. For each IoU threshold $\tau \in \{0.50, 0.55, \dots, 0.95\}$, predicted and ground-truth boxes of the same category are matched one-to-one by descending IoU. Precision, recall, and F1-score are then computed from the resulting true positives, false positives, and false negatives. We report the averaged F1-score over all IoU thresholds and categories as the primary metric.

Datasets. We evaluate dense detection on five representative remote sensing benchmarks: DIOR [1], HRSC2016 [10], WHU [22], and WHU-Mix [13]. DIOR is a comprehensive large-scale benchmark for object detection in optical remote sensing images. It contains 23,463 high-resolution images and 192,472 annotated instances across 20 object categories, including airplanes, ships, stadiums, and windmills. DIOR is characterized by substantial scene diversity, covering different weather conditions, seasons, imaging qualities, and object scales. HRSC2016 is a widely used benchmark tailored for ship detection in optical remote sensing imagery. It consists of 1,061 high-resolution images with spatial resolutions ranging from 0.4 m to 2 m, and provides 2,976 ship instances captured from sea and coastal harbor environments around the world. WHU is a high-quality benchmark focused on building extraction and detection in remote sensing imagery. It includes both aerial and satellite imagery subsets with diverse building annotations. The WHU-Mix is a large-scale building detection benchmark. It aggregates, curates, and corrects data from multiple sources, including WHU, Inria [49], SpaceNet [50], and CrowdAI [17], together with newly collected samples. Its test set contains imagery from five cities across five continents, with no geographic overlap with the training and validation data.

5.2 Dense segmentation

Metrics. For dense segmentation, we adopt mIoU as the primary evaluation metric and additionally compute precision, recall, and Dice. mIoU measures average region-overlap between predicted and ground-truth masks and is the standard metric for semantic segmentation. For clarity and compactness, we report only mIoU in the main results.

Datasets. We evaluate on four representative dense segmentation benchmarks: Globe230k [39], SIOR [31], SOTA [31], and LoveDA [38]. The test sets cover diverse targets—from large-area land-cover regions to discrete manmade objects—with varying resolutions and scene layouts. Globe230k is a large-scale land-cover segmentation benchmark derived from global remote sensing imagery. It focuses on dense pixel-level parsing of diverse land-cover types, including cropland, forest, grassland, wetland, water, impervious surface, bare land, ice/snow, and tundra. In our evaluation, Globe230k contains 92,453 mask samples from 45,953 unique RGB images at 512×512

resolution. SIOR is an object-centric remote sensing segmentation benchmark with instance-level masks for optical aerial images. It covers 20 object categories, such as airplane, airport, bridge, harbor, ship, vehicle, storage tank, stadium, windmill, and sports fields. The SIOR test set used in our experiments contains 856 mask samples from 586 RGB images at 800×800 resolution. SOTA is an object-level segmentation benchmark for remote sensing imagery. It contains 17 aerial object categories, including plane, ship, bridge, harbor, storage tank, small vehicle, large vehicle, swimming pool, tennis court, and roundabout. The evaluated SOTA test set includes 1,607 mask samples from 874 RGB images, with image size 1024×1024 . LoveDA is a high-resolution land-cover dataset designed for both semantic segmentation and domain adaptation. It covers urban and rural scenes and includes common land-cover categories such as building, road, water, barren land, forest, and agriculture. In our evaluation, the LoveDA test set contains 6,959 mask samples from 1,667 unique RGB images at 1024×1024 resolution..

5.3 Dense polygon extraction

Metrics. For dense polygon extraction, we evaluate generated vector contours with a multi-class polygon AP protocol. Predictions are grouped by image and category, ranked by confidence score, and greedily matched to ground-truth polygons of the same category using polygon-level IoU. We report the macro-averaged mAP over all categories as the primary metric, as it summarizes performance across IoU thresholds from 0.50 to 0.95. Detailed per-category results and additional metrics, including AP50, AP75, AR, Precision@0.50, Recall@0.50, F1@0.50, polygon union IoU, C-IoU, and N-ratio, are provided in the supplementary material.

Datasets. We evaluate on GeoPolygon, a self-constructed multi-category remote sensing benchmark for vector contour generation. The test set contains 1,452 image-level samples covering 18 categories and 8,820 polygon instances. The categories include agriculture, aircraft, baseball-field, basketball-court, buildings, chimney, dam, playground, roundabout, ship, soccer-field, stadium, storage-tank, swimming-pool, tennis-court, vehicle, water, and windmill. Images are collected from multiple sources, including DIOR [1], WHU-Mix [13], BHP-Pools [15], and other datasets [12, 14], with resolutions ranging from 256×256 to 1024×1024 . Unlike dense detection or raster segmentation benchmarks, GeoPolygon provides polygonal object contours as ground truth, enabling evaluation of both object localization and regularized geometric boundary fidelity.

5.4 In-context detection

Metrics. For in-context detection, we adopt a confidence-free evaluation protocol with greedy one-to-one IoU matching, as the model does not produce calibrated detection scores. A predicted box is a true positive if it matches to an unmatched ground-truth box with $\text{IoU} \geq 0.50$; otherwise it is a false positive, and unmatched ground-truth boxes are false negatives. We report F1@50 for the three categories—marine aquaculture cages (MAC), Oilwell, and SolarPanel—as the primary metric.

F1@50 reflects precision and recall under the standard IoU=0.50 criterion and enables compact comparison in the main results.

Datasets. We report results on three category-specific test sets: MAC, Oilwell, and SolarPanel. MAC contains 50 query images with 229 annotated instances, all at 1024×1024 JPEG resolution. Oilwell contains 50 query images with 74 annotated instances, covering mixed resolutions including 768×768 , 1024×1024 , 1024×768 , and 768×1024 . SolarPanel contains 50 query images with 242 annotated instances at 416×416 TIFF resolution. Each sample is paired with reference boxes according to the evaluated shot setting, with detection performance measured on the query annotations.

5.5 Referring detection

Metrics. For referring detection, each expression corresponds to a single target object, and the model predicts one bounding box for the referred object. We follow the TransVG [51] protocol and compute IoU between the predicted and the ground-truth boxes in xyxy format. A prediction is correct if its IoU ≥ 0.50 ; the final metric is Acc@0.5, i.e., the percentage of correctly localized expressions. If no valid box is parsed, the sample is counted as incorrect.

Datasets. We evaluate on two remote sensing visual grounding benchmarks, DIOR-RSVG [52] and RSVG [52], using both their validation and test splits. DIOR-RSVG is built from the DIOR detection dataset and contains 38,320 referring expressions, split into 26,991 training, 3,829 validation, and 7,500 test samples. All images are JPG, mostly at 800×800 resolution. RSVG contains 4,239 images and 7,933 expressions, split into 5,505 training, 1,201 validation, and 1,227 test samples. All evaluated RSVG images are 1024×1024 JPGs.

5.6 Change detection

Metrics. For change detection, we evaluate predicted binary change masks using mIoU. Given a pair of bi-temporal images, the model predicts a pixel-level mask of changed versus unchanged regions. We compute class-wise IoU and report the mean, which measures both change localization and false-alarm suppression in unchanged areas.

Datasets. We evaluate on three benchmarks: WHU-CD [22], SYSU-CD [53], and CCD [54]. WHU-CD is a high-resolution building change detection dataset derived from bi-temporal aerial images, with binary masks for building changes. In our evaluation split, it contains 193 test pairs. SYSU-CD is a large-scale binary change detection benchmark covering diverse land-cover and object changes with 4,000 test pairs at 256×256 resolution. CCD is a season-varying change detection dataset built from bi-temporal images with pixel-level binary change annotations. It provides 3,000 test pairs at 256×256 resolution.

5.7 VQA

Metrics. We follow the standard VQA protocol and report Overall Accuracy (OA), the percentage of exact matches between predictions and ground-truth answers.

Predictions are normalized before evaluation: lowercasing, removing articles and punctuation, and mapping equivalent forms (e.g., yes/no vs. true/false, or digit vs. word numbers) to a canonical representation. We report OA per split as the primary metric. The supplementary material provides per-question-type accuracy (e.g., presence, comparison) for finer-grained analysis of object recognition and quantity reasoning.

Datasets. We evaluate VQA on two representative remote sensing VQA benchmarks, RSVQA-LR [55] and RSVQA-HR [55], using three splits in total. RSVQA-LR is built from low-resolution Sentinel-2 imagery at 10 m spatial resolution (tiles of size 256×256) and contains 772 remote sensing images, with question-answer pairs automatically derived from OpenStreetMap (OSM); its test split (denoted LR) contains 33,212 questions covering presence, comparison, counting, and rural/urban classification. RSVQA-HR is built from USGS high-resolution (15.24 cm) orthorectified aerial RGB imagery; we evaluate on two splits. The standard test split (HR test), drawn from the same distribution as the training data, contains 960,669 questions, and the test_phili split (HR test_phili), collected over a geographically distinct region (Philadelphia) to assess cross-region generalization, contains 1,066,316 questions. Both HR splits cover presence, comparison, counting, and area-comparison questions, which together require the model to recognize remote sensing objects, reason about their quantity and area, and capture scene-level semantics.

We evaluate on two VQA benchmarks, RSVQA-LR [55] and RSVQA-HR [55], using three splits total. RSVQA-LR is built from Sentinel-2 imagery (10 m resolution, 256×256 tiles) and contains 772 images, with question-answer pairs derived from OpenStreetMap; its test split (denoted as LR) contains 33,212 questions covering presence, comparison, counting, and rural/urban classification. RSVQA-HR is built from USGS high-resolution (15.24 cm) orthorectified aerial RGB imagery; with two splits. The standard test split (HR test) contains 960,669 questions from the same distribution as the training data. The test_phili split (HR test_phili) contains 1,066,316 questions from a geographically distinct region (Philadelphia) to assess cross-region generalization. Both HR splits cover presence, comparison, counting, and area-comparison questions, requiring the model to recognize objects, reason about quantity and area, and capture scene-level semantics.

6 Ablation study on mask tokenizer design

6.1 Ablation experiment setup

We conduct ablation studies on seven variables within the design space of the mask tokenizer. Let **K** denote the number of continuous mask embeddings, **Compact** indicate whether a compact design is adopted, **Quant.** denote the quantization algorithm, **D** refer to the dimension of latent vectors in the codebook, **Width** represent the number of latent vectors in each codebook layer, **Depth** denote the number of quantization iterations, and **Shared** indicate whether latent vectors are shared across codebook layers. The ablation experiments evaluate mask tokenizer quality via mask reconstruction accuracy. Specifically, we feed a ground-truth mask through the tokenizer to obtain discrete code IDs, decode them back into predicted masks resize the predictions to the original image resolution, binarize them at a threshold of 0.5, and finally report IoU. Reconstruction accuracy is evaluated on the test sets of nine remote sensing segmentation datasets: DGLandCover [25], DGRoad [25], DLRSD [36], EVLab-SS [44], LandCoverAI [34], LoveDA [38], NWPU [5], OEM [29] [35] and Postdam [35].

6.2 Ablation analysis of mask encoder

Under non-compact vector quantization (VQ) settings, increasing the number of continuous mask embeddings steadily improves reconstruction quality. With a fixed codebook width of 4096, mIoU rises from 73.97 (4 embeddings) to 74.88 (8 embeddings), 77.95 (16 embeddings), and 83.58 (32 embeddings). This indicates that, as the number of discrete tokens grows with K, the continuous representation capacity of the mask encoder becomes the primary bottleneck; more mask queries preserve finer-grained shape and contour details.

However, this gain incurs significant autoregressive cost. The 32-token VQ achieves the best reconstruction among non-compact variants, but requires generating 32 discrete tokens per mask—substantially increasing generation length and training difficulty for MLLMs.

The compact design reshapes this trade-off. It uses multiple continuous embeddings to capture geometry, then compresses them into a single latent for quantization. Under the compact residual quantization (RQ) setting with 8 continuous embeddings, a mean IoU of 80.78–81.03 is achieved with only 4 discrete residual tokens—approximately 5.9 points higher than the non-compact 8-token VQ, while halving the token count.

6.3 Ablation analysis of codebook

Under the non-compact VQ setting with 16 consecutive embeddings, codebook width shows a monotonic positive relationship with reconstruction quality. Reducing width from 4096 to 256 decreases mIoU from 77.95 to 69.88, while increasing it to 8192 raises mIoU to 79.38. This indicates that single-level VQ is highly sensitive to codebook capacity: each consecutive embedding is approximated by a single code.

For compact RQ, the marginal gain of width are smaller. With latent dimension 512 and depth 4, increasing per-stage width from 256 to 1024 raises mIoU from 80.78

to 81.03. Residual combinations already provide strong representational capacity, so a moderately sized per-stage codebook suffices for satisfactory quality.

Codebook sharing leads to a slight performance drop. Under compact RQ setup with latent dimension 512, width 256, and depth 4, independent codebooks achieve 80.78 mIoU, which drops to 80.16 with sharing. This suggests that different residual stages capture distinct distributions, and stage-specific codebooks better fit stage-wise residuals.

The compact latent dimension has a notable impact on performance. Reducing it from 512 to 256 causes a sharp mIoU decline from 80.78 to 70.85, while increasing it to 1024 only yields only marginal gain (80.78 to 80.82). Therefore, 512 is a suitable choice under the current configuration.

6.4 Ablation analysis of quantization algorithm

Under a tight token budget, RQ significantly outperforms single-level VQ. With four discrete tokens, compact RQ achieves 80.78 mIoU using four 256-way residual codebooks, whereas non-compact VQ reaches only 73.97 mIoU with four 4096-way tokens. This indicates that residual composition is more effective than independent quantization of a few mask embeddings.

RQ depth is critical to performance. Reducing depth from 4 to 2 and tokens from 4 to 2 drops mIoU from 80.78 to 75.09. The drop is especially pronounced on datasets with slender structures or complex boundaries (e.g., DGRoad and OEM), confirming that subsequent residual stages encode important shape refinements than redundant tokens.

A single compact VQ token cannot adequately represent a complete mask: compact VQ with four 256-way codebooks (depth 4, flattened) attains only 36.74 mIoU. This negative result shows that expanding the flat vocabulary cannot replace residual quantization; the hierarchical decomposition of RQ provides a discrete representation that is easier to optimize and more combinatorial.

6.5 Recommended configuration

For reconstruction quality upper bound, the best-performing configuration in the current ablation study is non-compact 32-token VQ: $K = 32$, *codebooksize* = 4096, *depth* = 1, achieving 83.58 mIoU. This configuration is suitable for scenarios where tokenizer reconstruction quality is the primary concern and token length is not a bottleneck.

For autoregressive generation in MLLMs, compact RQ offers a more favorable quality–token trade-off. The recommended configuration is: $K = 8$ consecutive mask embeddings, a compact latent dimension of 512, and 4 independent 256-way residual codebooks. This generates 4 mask tokens with 80.78 mIoU. If a slightly larger vocabulary is acceptable, increasing codebook width to 1024 marginally improves mIoU to 81.03. The ablation study supports three main conclusions: first, the mask encoder should retain multiple consecutive mask embeddings before discretization; second, single-level VQ requires a large codebook and long token sequences to achieve high

Table 10: Mask tokenizer ablation study results. Ablation results across seven design variables: **K** (number of continuous mask embeddings), **Compact** (whether a compact bottleneck is used), **Quant.** (quantization algorithm: VQ for single-level vector quantization, or RQ for residual quantization), **D** (latent dimension), **Width** (codebook size per layer), **Depth** (quantization iterations), and **Shared** (whether codebooks are shared across layers). The number of discrete tokens is determined by these variables. Reconstruction accuracy is measured by mean IoU (%) averaged over nine remote sensing segmentation datasets.

	K	Compact	Quant.	D	Width	Depth	Shared	Discrete tokens	Mean IoU
①	4	No	VQ	256	4096	1	No	4	73.97
②	8	No	VQ	256	4096	1	No	8	74.88
③	16	No	VQ	256	4096	1	No	16	77.95
④	16	No	VQ	256	8192	1	No	16	79.38
⑤	16	No	VQ	256	2048	1	No	16	76.89
⑥	16	No	VQ	256	1024	1	No	16	74.88
⑦	16	No	VQ	256	256	1	No	16	69.88
⑧	32	No	VQ	256	4096	1	No	32	83.58
⑨	8	Yes	RQ	512	256	4	No	4	80.78
⑩	8	Yes	RQ	512	256	4	Yes	4	80.16
⑪	8	Yes	RQ	1024	256	4	No	4	80.82
⑫	8	Yes	RQ	256	256	4	No	4	70.85
⑬	8	Yes	RQ	512	1024	4	No	4	81.03
⑭	8	Yes	RQ	512	256	2	No	2	75.09
⑮	8	Yes	VQ	512	1024	1	No	1	36.74

Table 11: Per-dataset mask tokenizer ablation results. Reconstruction accuracy (IoU, %) on nine remote sensing segmentation datasets: DGLandCover, DGRoad, DLRSD, EVLab-SS, LandCoverAI, LoveDA, NWPU, OEM and Potsdam.

	Mean IoU	DGLandCover	DGRoad	DLRSD	EVLab-SS	LandCoverAI	LoveDA	NWPU	OEM	Potsdam
①	73.97	86.57	55.73	74.72	75.97	79.50	75.02	81.98	58.88	77.40
②	74.88	87.70	56.18	75.74	77.96	80.00	76.49	81.65	59.14	79.10
③	77.95	89.89	59.16	79.87	82.29	82.20	79.90	83.03	62.34	82.88
④	79.38	90.68	60.04	81.97	84.73	83.20	81.48	83.62	63.84	84.89
⑤	76.89	89.11	58.02	78.03	81.08	80.78	79.52	81.89	61.16	82.40
⑥	74.88	87.59	55.65	76.14	78.92	79.03	76.99	79.95	58.88	80.73
⑦	69.88	82.72	50.53	71.79	73.24	74.10	71.72	75.36	53.47	75.97
⑧	83.58	93.32	66.42	86.07	89.00	87.11	86.16	85.82	69.62	88.73
⑨	80.78	91.63	62.51	83.18	85.93	84.64	83.10	84.05	65.89	86.13
⑩	80.16	91.02	62.34	82.40	85.07	83.90	82.18	83.17	65.41	85.92
⑪	80.82	91.76	62.80	82.54	86.01	84.96	83.18	83.87	65.89	86.37
⑫	70.85	83.38	52.94	71.17	72.02	75.86	71.73	79.84	55.65	75.10
⑬	81.03	91.95	62.81	83.37	86.07	84.89	83.36	84.42	66.08	86.28
⑭	75.09	89.87	56.83	72.91	79.38	79.04	74.64	84.19	58.77	80.18
⑮	36.74	59.60	10.61	33.66	41.32	30.04	37.91	61.10	23.46	32.96

reconstruction quality; third, compact residual quantization provides the optimal balance between quality and token budget, making it the more suitable mask tokenizer for MLLMs.

7 Ablation: discrete vs. continuous mask representation

Table 12: Discrete vs. continuous mask representation on Globe230k. Both models are trained for one epoch on the same segmentation data and evaluated with mIoU. RL gains are reported relative to each model’s SFT checkpoint.

Method	Mask Representation	SFT mIoU	SFT + RL mIoU	RL Gain
Sa2VA-style baseline	Continuous mask embedding	49.3	49.5	+0.2
VectorLLM++	Discrete mask tokens	53.6	55.9	+2.3

VectorLLM++ represents each binary mask as a short sequence of discrete mask tokens, bringing dense segmentation into the same next-token prediction paradigm as language, bounding boxes, and polygons. To test whether this discrete formulation outperforms the continuous mask-embedding approach used in prior work, we compare against a Sa2VA-style [56] continuous baseline.

Both models share the same Qwen3-VL [57] backbone and SAM2-based segmentation module [11]; the only difference is the output-side mask representation. VectorLLM++ encodes each mask into four residual-quantized discrete codes via its mask tokenizer. In contrast, the continuous baseline follows the [SEG]-token design, where a generated marker indicates the position of a continuous embedding, which is then passed to the mask decoder. Since the original Sa2VA is mainly designed for referring or semantic segmentation and supports only a single [SEG] token per output, we extend it to [SEG001]–[SEG300], allowing up to 300 masks per response. For a fair comparison, both models are trained for one epoch on the datasets in Table 7 and evaluated on Globe230k [39] with mIoU. We further apply RL post-training to both to assess how each scheme benefits from the same alignment stage.

As shown in Table 12, the discrete mask-token formulation outperforms the continuous baseline under the same training data. After SFT, VectorLLM++ improves Globe230k mIoU from 49.3 to 53.6. After RL post training, the gap widens further: VectorLLM++ reaches 55.9 mIoU (+2.3 over SFT), while the continuous baseline improves by only 0.2 points. These results indicate that discrete mask tokens offer a more effective interface for both SFT and RL post-training. Unlike continuous embeddings, which are attached to special markers and optimized via hidden feature regression, discrete tokens are predicted directly from the language model vocabulary and can be optimized jointly with other text and geometric tokens. This aligns binary-mask generation more closely with the autoregressive training objective and facilitates joint optimization with other dense remote sensing tasks.

8 Detailed polygon extraction results

Dense polygon extraction is evaluated on the GeoPolygon benchmark. Since this task requires explicit vector contours rather than bounding boxes or raster masks, it is central to assessing the geometric generation capability of VectorLLM++. The main text reports overall mAP for compactness. Here, we provide per-category results and additional metrics: AP50, AP75, AR, mean polygon IoU, and contour-IoU (C-IoU).

Few remote sensing MLLMs support polygon outputs, and those that do generally lack open-vocabulary category specification. We therefore use VectorLLM—equipped with Rex-Omni, an open-vocabulary detector—as the polygon-output baseline for the multi-category GeoPolygon benchmark. Tables 13 and 14 report the detailed comparison. The VectorLLM++ results are from the SFT checkpoint; the RL checkpoint results are in the main table.

Overall, VectorLLM++ outperforms VectorLLM by 12.0 points in mAP, 18.5 in AP50, 13.4 in AP75, and 14.4 in AR. It also raises mean polygon IoU from 49.9 to 70.0 and C-IoU from 41.3 to 56.8, reflecting both better object matching and contour alignment. At the category level, VectorLLM++ achieves higher mAP on 14 of 18 categories, with the largest gains on vehicle, chimney, stadium, water, and buildings (all >20 points). Performance on windmill and agriculture remains limited, suggesting that sparse objects, small structures, and highly variable land-cover regions remain challenging for polygon extraction.

Table 13: Detailed dense polygon extraction results on GeoPolygon (part 1 of 2). The table reports the overall macro-average and the first nine categories.

Category	Method	mAP	AP ₅₀	AP ₇₅	AR	IoU	C-IoU
Overall	VectorLLM	23.5	38.6	23.0	33.2	49.9	41.3
	VectorLLM++	35.5	57.1	36.4	47.6	70.0	56.8
agriculture	VectorLLM	4.4	11.5	3.0	9.1	35.0	17.9
	VectorLLM++	0.2	0.8	0.1	0.6	63.4	14.8
aircraft	VectorLLM	15.1	45.3	5.9	30.9	60.3	54.7
	VectorLLM++	16.5	59.9	4.0	29.7	62.3	55.8
baseball-field	VectorLLM	38.0	54.1	42.3	52.4	65.4	60.0
	VectorLLM++	36.9	54.8	41.6	50.0	63.6	57.0
basketball-court	VectorLLM	38.1	48.3	39.9	51.3	77.6	64.9
	VectorLLM++	47.8	60.4	53.5	59.8	81.9	73.5
buildings	VectorLLM	16.0	33.0	14.2	26.2	51.0	34.0
	VectorLLM++	39.2	65.1	43.7	56.9	75.5	66.7
chimney	VectorLLM	2.1	4.1	1.6	5.8	8.0	7.5
	VectorLLM++	43.2	75.5	42.5	53.9	75.3	67.6
dam	VectorLLM	0.9	2.0	0.7	5.0	14.0	11.2
	VectorLLM++	8.5	28.8	3.8	20.9	42.8	32.3
playground	VectorLLM	14.6	30.9	12.7	24.0	30.0	21.4
	VectorLLM++	32.6	60.4	32.0	47.0	69.4	52.6
roundabout	VectorLLM	19.9	40.4	13.3	36.1	54.9	44.0
	VectorLLM++	33.3	49.4	35.7	54.8	65.5	53.5

Table 14: Detailed dense polygon extraction results on GeoPolygon (part 2 of 2) . Remaining nine categories (continued from Table 13).

Category	Method	mAP	AP ₅₀	AP ₇₅	AR	IoU	C-IoU
ship	VectorLLM	62.6	85.9	72.4	73.9	81.1	70.1
	VectorLLM++	68.0	92.0	77.7	76.2	85.3	72.7
soccer-field	VectorLLM	23.4	40.4	24.2	35.0	69.7	63.5
	VectorLLM++	30.5	46.1	34.1	43.0	75.0	65.4
stadium	VectorLLM	28.7	56.8	24.1	44.9	64.7	48.2
	VectorLLM++	54.6	83.5	54.0	64.1	78.7	59.5
storage-tank	VectorLLM	26.8	51.3	24.8	43.5	67.9	63.3
	VectorLLM++	18.6	38.3	15.5	44.1	74.6	68.4
swimming-pool	VectorLLM	38.4	64.3	40.5	49.0	64.3	53.6
	VectorLLM++	48.4	72.2	54.5	58.4	73.5	61.6
tennis-court	VectorLLM	82.2	92.4	87.4	87.4	90.4	89.1
	VectorLLM++	81.3	90.2	88.8	86.5	88.8	86.9
vehicle	VectorLLM	4.4	7.6	3.8	5.3	17.7	14.4
	VectorLLM++	45.6	86.2	45.0	59.0	74.5	68.0
water	VectorLLM	8.0	25.7	3.1	17.2	37.3	18.4
	VectorLLM++	31.9	58.9	29.3	47.1	71.3	34.7
windmill	VectorLLM	0.0	0.1	0.0	0.3	7.9	6.4
	VectorLLM++	0.9	5.6	0.0	3.9	39.1	31.8

9 Mask ordering strategies for autoregressive prediction

Table 15: Ablation of mask ordering strategies for autoregressive mask prediction. All variants are trained on the same 1M dense segmentation samples for one non-repeated epoch and evaluated on Globe230k with mIoU.

Ordering strategy	30% progress	60% progress	100% progress
Patch row-major	40.1	54.4	58.2
Random	20.8	43.4	49.5
Mask area	37.7	48.1	53.3
Bbox top-left	39.8	52.0	57.4

When an image contains multiple instances, their masks must be serialized into a 1D token sequence for autoregressive prediction. This ordering is non-trivial because masks are defined on the 2D image plane, yet both language and visual tokens are processed as 1D sequences in an MLLM. Therefore, the mask ordering strategy determines how the spatial layout of multiple instances is exposed to next-token prediction.

We compare four ordering strategies. (1) patch row-major: each mask is patchified and flattened in row-major order; masks are ordered by the earliest occupied patch in this sequence. (2) Random: masks are shuffled per sample, removing any spatial prior. (3) Area-based: masks are sorted by pixel area descending, generating larger (typically more salient) instances first. (4) Bbox top-left: masks are sorted by the top-left corner of their axis-aligned bounding box in top-to-bottom, left-to-right order. While bbox top-left ordering is common for detection-style outputs, it can be less suitable for remote sensing masks with elongated or irregular shapes, where the top-left corner may be far from the main object body.

To select the default strategy, we ablate on the same dense segmentation training data. Each variant is trained on 1M samples for one non-repeated epoch, with checkpoints at 30%, 60%, and 100% training progress evaluated on Globe230k using mIoU. As shown in Table 15, patch row-major ordering achieves the best mIoU at all three checkpoints.

The comparison shows that preserving row-major order of the patchified image provides the most effective serialization prior. Random ordering performs worst, especially in the early training stage, confirming that a consistent spatial order helps. Area-based ordering is weaker as it decouples output from image layout. Bbox top-left ordering is competitive but underperforms patch row-major ordering at all checkpoints. We therefore adopt patch row-major as the default mask serialization strategy in VectorLLM++.

10 Qualitative results

To visually assess VectorLLM++’s dense perception capability, we present qualitative comparisons on six tasks: dense detection, dense segmentation, dense polygon extraction, in-context detection, referring detection, and change detection. These examples complement the quantitative results, illustrating differences in recall, spatial coverage, boundary quality, visual and text grounding, and change completeness.

Dense detection. As shown in Fig. 3, VectorLLM++ detects more instances than the previous best, especially in scenes with dense, small, or repeated objects. The RL checkpoint further improves recall, recovering instances missed by the SFT model. The examples also show that VectorLLM++ better respects category-specific annotation granularity, producing object-level boxes rather than merging adjacent instances or producing inconsistent region-level responses.

Dense segmentation. Fig. 4 shows a similar trend. The previous best often misses many objects, resulting in incomplete coverage. VectorLLM++ substantially improves mask recall and identifies more foreground regions. However, the examples also reveal a remaining limitation: objects with long, thin, fragmented, or topologically complex shapes, such as roads, remain difficult to segment precisely. This suggests that the main performance comes from better object coverage, while fine-grained boundary tracing for complex structures remains challenging.

Dense polygon extraction. As illustrated in Fig. 5, VectorLLM++ produces polygons that align more closely with object contours than the previous baseline. The contours are also more regular, with fewer spurious vertices and less shape distortions. This is important for vector mapping applications, where outputs must be both correctly localized and usable as regular vector boundaries.

In-context detection. As shown in Fig. 6, the previous best exhibits two typical failure modes: missing objects of the exemplar category and including visually similar but semantically different distractors. VectorLLM++ better interprets the exemplar as a category-level visual prompt, retrieving more same-category objects while reducing false positives from other types. This indicates stronger exemplar-conditioned matching, which is particularly useful for categories that are difficult to specify by text alone.

Referring detection. Fig. 7 compares text-conditioned grounding. VectorLLM++ shows stronger localization for small targets and crowded scenes, where the referred object may occupy only a small image fraction. This suggests better alignment between language expressions and fine-grained visual evidence.

Change detection. Since Falcon [58]’s weights are not publicly available, we do not include its qualitative results. Instead, Fig. 8 compares predictions before and after RL post-training. The RL checkpoint produces more complete change masks and suppresses fragmented predictions. It also identifies changed objects more accurately, indicating that RL improves the spatial quality of change-mask generation.



Fig. 3: Qualitative comparison on dense detection. VectorLLM++ improves object recall and better matches category-specific annotation granularity; RL post-training further recovers missed instances.

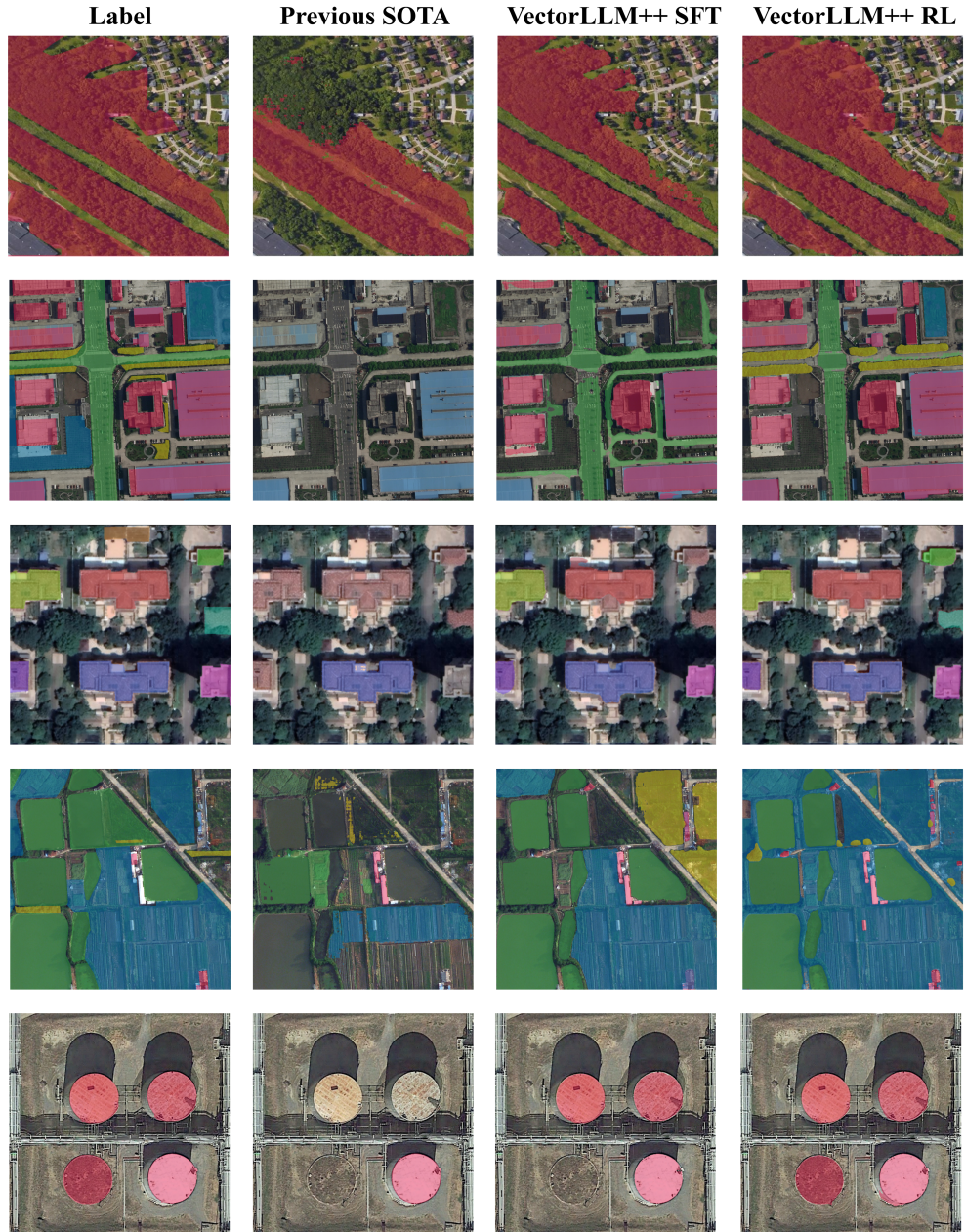


Fig. 4: Qualitative comparison on dense segmentation. VectorLLM++ improves mask recall and scene coverage, though elongated and structurally complex objects remain challenging.

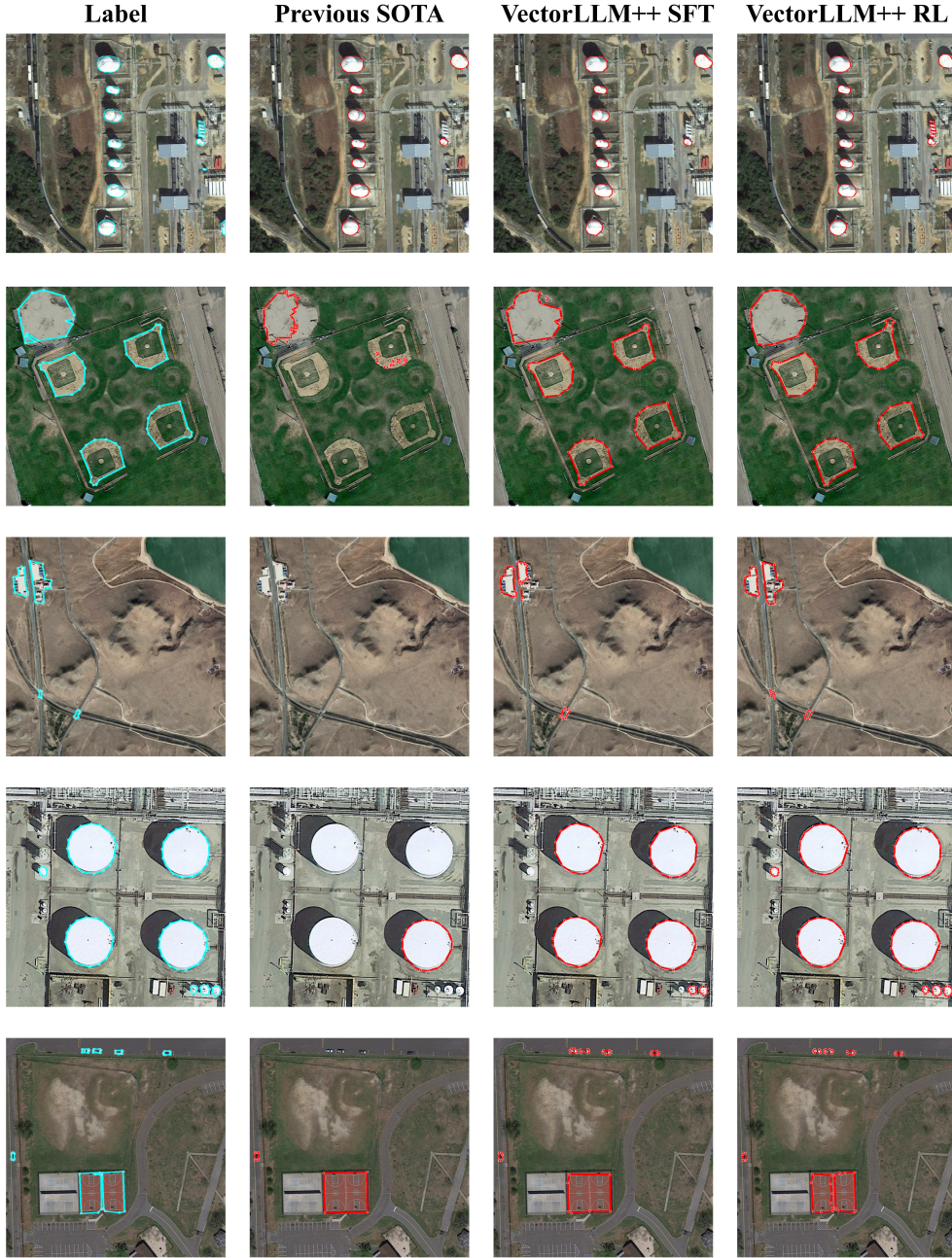


Fig. 5: Qualitative comparison on dense polygon extraction. VectorLLM++ generates more regular vector contours and better aligns polygons with object boundaries.

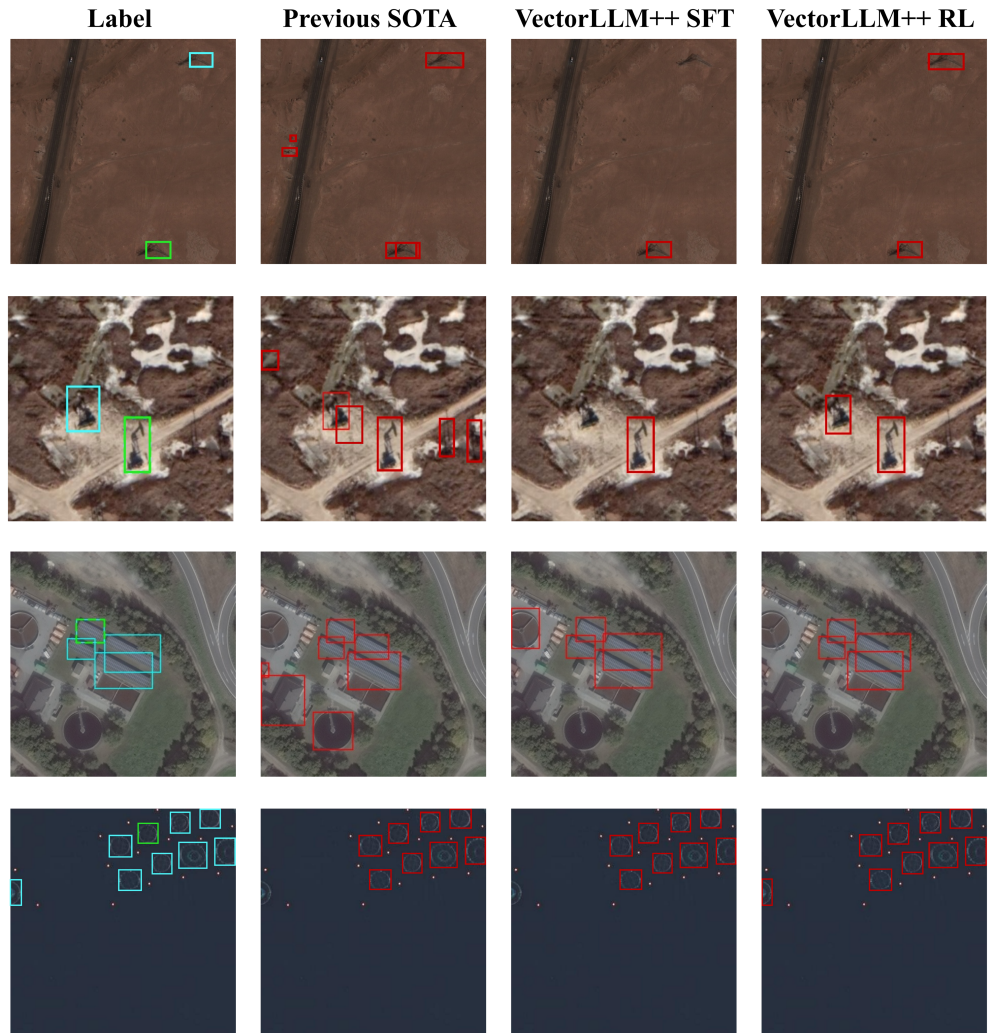


Fig. 6: Qualitative comparison on in-context detection. Given a visual exemplar, VectorLLM++ retrieves more same-category objects and suppresses distractors from other categories.

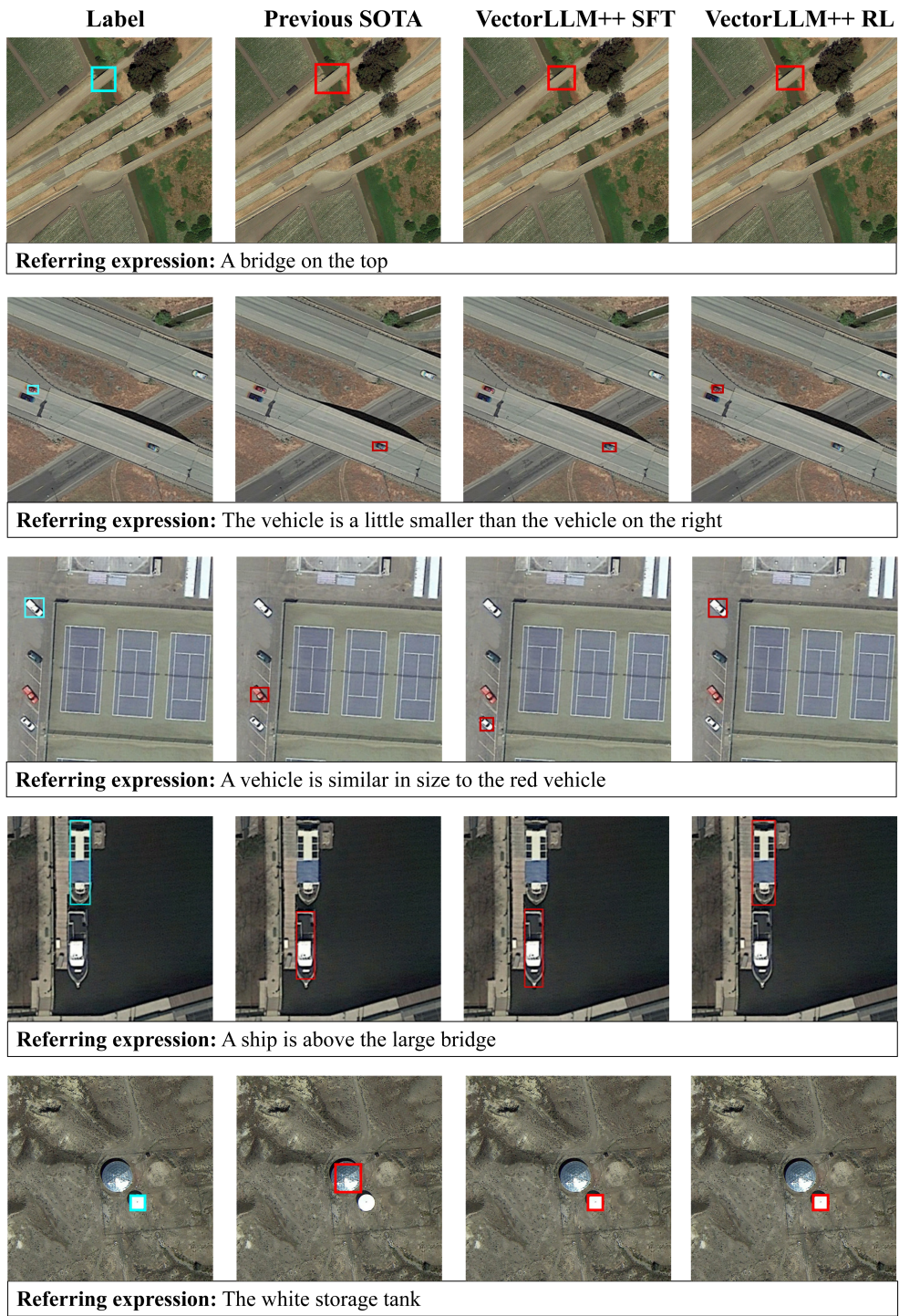


Fig. 7: Qualitative comparison on referring detection. VectorLLM++ better grounds text expressions to small or visually crowded remote sensing objects.



Fig. 8: Qualitative comparison on change detection. Since Falcon weights are not publicly available, we compare VectorLLM++ before and after RL post-training; RL improves change-mask completeness and changed-object localization.

References

- [1] Li, K., Wan, G., Cheng, G., Meng, L., Han, J.: Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing* **159**, 296–307 (2020)
- [2] Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: DOTA: A large-scale dataset for object detection in aerial images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983 (2018)
- [3] Long, Y., Gong, Y., Xiao, Z., Liu, Q.: Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* **55**(5), 2486–2498 (2017)
- [4] Sun, X., Wang, P., Yan, Z., Xu, F., Wang, R., Diao, W., Chen, J., Li, J., Feng, Y., Xu, T., *et al.*: Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* **184**, 116–130 (2022)
- [5] Cheng, G., Han, J., Zhou, P., Guo, L.: Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing* **98**, 119–132 (2014)
- [6] Tong, X.-Y., Xia, G.-S., Lu, Q., Shen, H., Li, S., You, S., Zhang, L.: Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment* **237**, 111322 (2020)
- [7] Chen, H., Shi, Z.: A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote sensing* **12**(10), 1662 (2020)
- [8] Haroon, M., Shahzad, M., Fraz, M.M.: Multisized object detection using space-borne optical imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **13**, 3032–3046 (2020)
- [9] Zhu, H., Chen, X., Dai, W., Fu, K., Ye, Q., Jiao, J.: Orientation robust object detection in aerial images using deep convolutional neural network. In: *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3735–3739 (2015). IEEE
- [10] Liu, Z., Yuan, L., Weng, L., Yang, Y.: A high resolution optical satellite image dataset for ship recognition and some new baselines. In: *International Conference on Pattern Recognition Applications and Methods*, vol. 2, pp. 324–331 (2017). SciTePress
- [11] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., *et al.*: Sam 2: Segment anything in images and

- videos. In: International Conference on Learning Representations, vol. 2025, pp. 28085–28128 (2025)
- [12] Li, J., Wei, Y., Wei, T., He, W.: A comprehensive deep-learning framework for fine-grained farmland mapping from high-resolution images. *IEEE Transactions on Geoscience and Remote Sensing* **63**, 1–15 (2024)
- [13] Luo, M., Ji, S., Wei, S.: A diverse large-scale building dataset and a novel plug-and-play domain generalization method for building extraction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **16**, 4122–4138 (2023)
- [14] Qi, X., Zhu, P., Wang, Y., Zhang, L., Peng, J., Wu, M., Chen, J., Zhao, X., Zang, N., Mathiopoulos, P.T.: Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing* **169**, 337–350 (2020)
- [15] Cunha, H.S., Sclauser, B.S., Wildenberg, P.F., Fernandes, E.A.M., Dos Santos, J.A., Lage, M.d.O., Lorenz, C., Barbosa, G.L., Quintanilha, J.A., Chiaravalloti-Neto, F.: Water tank and swimming pool detection based on remote sensing and deep learning: Relationship with socioeconomic level and applications in dengue control. *Plos one* **16**(12), 0258681 (2021)
- [16] Razakarivony, S., Jurie, F.: Vehicle detection in aerial imagery (vedai): a benchmark. Technical report, Tech. Rep., 2015. 2 (2014)
- [17] Mohanty, S.P.: CrowdAI Mapping Challenge 2018 Dataset. [Online]. Accessed Nov. 10, 2019 (2019). <https://www.crowdai.org/challenges/mapping-challenge>
- [18] Faudi, J., Martin: Airbus Ship Detection Challenge. Kaggle. Accessed Nov. 10, 2019 (2018). <https://kaggle.com/competitions/airbus-ship-detection>
- [19] Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., Han, J.: Towards large-scale small object detection: Survey and benchmarks. *IEEE transactions on pattern analysis and machine intelligence* **45**(11), 13467–13488 (2023)
- [20] Zhang, Z., Zhang, L., Wang, Y., Feng, P., He, R.: Shippersimagenet: A large-scale fine-grained dataset for ship detection in high-resolution optical remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **14**, 8458–8472 (2021)
- [21] İlmak, D., Bakirman, T., Sertel, E.: Exploring you only look once v8 and v9 for efficient airplane detection in very high resolution remote sensing imagery. *Engineering Applications of Artificial Intelligence* **160**, 111854 (2025)
- [22] Ji, S., Wei, S., Lu, M.: Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions*

- on geoscience and remote sensing **57**(1), 574–586 (2018)
- [23] Carion, N., Gustafson, L., Hu, Y.-T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., *et al.*: Sam 3: Segment anything with concepts. In: International Conference on Learning Representations, vol. 2026, pp. 138846–138923 (2026)
 - [24] Zhu, Q., Zhang, Y., Wang, L., Zhong, Y., Guan, Q., Lu, X., Zhang, L., Li, D.: A global context-aware and batch-independent network for road extraction from vhr satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* **175**, 353–365 (2021)
 - [25] Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R.: Deepglobe 2018: A challenge to parse the earth through satellite images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 172–181 (2018)
 - [26] Liu, Y., Yao, J., Lu, X., Xia, M., Wang, X., Liu, Y.: Roadnet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images. *IEEE Transactions on Geoscience and Remote Sensing* **57**(4), 2043–2056 (2018)
 - [27] Zhao, D., Yuan, B., Chen, Z., Li, T., Liu, Z., Li, W., Gao, Y.: Panoptic perception: A novel task and fine-grained dataset for universal remote sensing image interpretation. *IEEE Transactions on Geoscience and Remote Sensing* **62**, 1–14 (2024)
 - [28] Xin, Z., Li, K., Chen, L., Li, W., Yuchen, X., Qiao, H., Zhang, W., Meng, D., Cao, X.: Segearth-r2: Towards comprehensive language-guided segmentation for remote sensing images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13199–13210 (2026)
 - [29] Xia, J., Yokoya, N., Adriano, B., Broni-Bediako, C.: Openearthmap: A benchmark dataset for global high-resolution land cover mapping. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6254–6264 (2023)
 - [30] Carvalho, O.L.F., Carvalho Júnior, O.A., Silva, C.R.e., Albuquerque, A.O., Santana, N.C., Borges, D.L., Gomes, R.A.T., Guimarães, R.F.: Panoptic segmentation meets remote sensing. *Remote Sensing* **14**(4), 965 (2022)
 - [31] Wang, D., Zhang, J., Du, B., Xu, M., Liu, L., Tao, D., Zhang, L.: Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. *Advances in Neural Information Processing Systems* **36**, 8815–8827 (2023)
 - [32] Chen, Z., Wang, C., Li, J., Xie, N., Han, Y., Du, J.: Reconstruction bias u-net for road extraction from optical remote sensing images. *IEEE Journal of Selected*

- [33] Liu, S., Chen, L., Zhang, L., Hu, J., Fu, Y.: A large-scale climate-aware satellite image dataset for domain adaptive land-cover semantic segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* **205**, 98–114 (2023)
- [34] Boguszewski, A., Batorski, D., Ziemba-Jankowska, N., Dziedzic, T., Zambrzycka, A.: Landcover. ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1102–1110 (2021)
- [35] Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U.: The isprs benchmark on urban object classification and 3d building reconstruction (2012)
- [36] Shao, Z., Yang, K., Zhou, W.: Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset. *Remote Sensing* **10**(6), 964 (2018)
- [37] Garioud, A., Giordano, S., David, N., Gonthier, N.: Flair-hub: Large-scale multimodal dataset for land cover and crop mapping. *ISPRS Journal of Photogrammetry and Remote Sensing* **237**, 271–300 (2026)
- [38] Wang, J., Zheng, Z., Ma, A., Lu, X., Zhong, Y.: Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1 (2021)
- [39] Shi, Q., He, D., Liu, Z., Liu, X., Xue, J.: Globe230k: A benchmark dense-pixel annotation dataset for global land cover mapping. *Journal of Remote Sensing* **3**, 0078 (2023)
- [40] Dong, Z., Sun, Y., Liu, T., Zuo, W., Gu, Y.: Cross-modal bidirectional interaction model for referring remote sensing image segmentation. *arXiv preprint arXiv:2410.08613* (2024)
- [41] Liu, S., Ma, Y., Zhang, X., Wang, H., Ji, J., Sun, X., Ji, R.: Rotated multi-scale interaction network for referring remote sensing image segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26658–26668 (2024)
- [42] Yuan, Z., Mou, L., Hua, Y., Zhu, X.X.: Rrsis: Referring remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing* **62**, 1–12 (2024)
- [43] Etten, A.V.: City-scale road extraction from satellite imagery v2: Road speeds and travel times. In: *Proceedings of the IEEE/CVF Winter Conference on*

- Applications of Computer Vision, pp. 1786–1795 (2020)
- [44] Zhang, M., Hu, X., Zhao, L., Lv, Y., Luo, M., Pang, S.: Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images. *Remote Sensing* **9**(5), 500 (2017)
 - [45] Deng, X., Yu, Q., Wang, P., Shen, X., Chen, L.-C.: Coconut: Modernizing coco segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21863–21873 (2024)
 - [46] Su, Y., Zhang, H., Li, S., Liu, N., Liao, J., Pan, J., Liu, Y., Xing, X., Sun, C., Li, C., *et al.*: Patch-as-decodable-token: Towards unified multi-modal vision tasks in mllms. In: *International Conference on Learning Representations*, vol. 2026, pp. 147893–147919 (2026)
 - [47] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20 (2016)
 - [48] Chen, Y., Wang, Y., Lu, P., Chen, Y., Wang, G.: Large-scale structure from motion with semantic constraints of aerial images. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 347–359 (2018). Springer
 - [49] Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P.: Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 3226–3229 (2017). IEEE
 - [50] Van Etten, A., Lindenbaum, D., Bacastow, T.M.: Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232* (2018)
 - [51] Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1769–1779 (2021)
 - [52] Zhan, Y., Xiong, Z., Yuan, Y.: Rsvg: Exploring data and models for visual grounding on remote sensing data. *IEEE transactions on geoscience and remote sensing* **61**, 1–13 (2023)
 - [53] Shi, Q., Liu, M., Li, S., Liu, X., Wang, F., Zhang, L.: A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. *IEEE transactions on geoscience and remote sensing* **60**, 1–16 (2021)
 - [54] Lebedev, M., Vizilter, Y.V., Vygolov, O., Knyaz, V.A., Rubis, A.Y.: Change

detection in remote sensing images using conditional adversarial networks. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **42**, 565–571 (2018)

- [55] Lobry, S., Marcos, D., Murray, J., Tuia, D.: Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* **58**(12), 8555–8566 (2020)
- [56] Yuan, H., Li, X., Zhang, T., Sun, Y., Huang, Z., Xu, S., Ji, S., Tong, Y., Qi, L., Feng, J., et al.: Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001* (2025)
- [57] Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631* (2025)
- [58] Yao, K., Xu, N., Yang, R., Xu, Y., Gao, Z., Kitrungrotsakul, T., Ren, Y., Zhang, P., Wang, J., Wei, N., et al.: Falcon: A remote sensing vision-language foundation model (technical report). *arXiv preprint arXiv:2503.11070* (2025)