SkySense-VITA: Towards Universal In-context Segmentation of Multi-modal Remote Sensing Imagery

Supplementary Material

A. Task Definition and Evaluation Protocol

A.1. In-context Segmentation Task Definition

In-context segmentation aims to segment all object instances of a specific category in a target image, guided by a context prompt. Formally, given a target image I_t and a prompt P (which can be a visual example or a text description) specifying a target category c, the model is required to predict a binary mask $M_t \in \{0,1\}^{H\times W}$, where pixels belonging to category c are labeled as 1 and others as 0.

Unlike traditional semantic segmentation which classifies every pixel into a fixed set of categories, in-context segmentation is query-dependent. The semantic definition of the target relies entirely on the provided prompt P.

A.2. Evaluation Protocol

Standard Protocol. For our SkySense-VITA and other in-context segmentation baselines (e.g., Painter [49], SegGPT [50]), we follow the standard few-shot segmentation evaluation protocol. For each test episode, the model is provided with a prompt (e.g., a support image-mask pair for visual prompting) corresponding to class c, and it must segment class c in the target image. We evaluate performance using the mIoU of the binary foreground prediction against the ground truth.

Adaptation for Open-Vocabulary Models. Existing Open-Vocabulary (OV) segmentation methods (e.g., SegEarth-OV [17], SkySense-O [61]) typically operate in a multi-class setting driven by textual prompts. To ensure a fair comparison with the binary nature of in-context segmentation tasks, we adapt these OV models using two prompting strategies:

- **Strategy A (Target vs. Background):** We construct the text prompt set as $\{$"c", "background"$\}$, forcing the model to perform binary classification.
- **Strategy B (Full-Vocabulary):** We construct the text prompt set using all categories present in the dataset (e.g., $\{c_1, c_2, ..., c_N\}$). We then extract the probability map corresponding to the target class c and treat all other categories as the negative class (background).

To report the most competitive results for these baselines and mitigate the influence of prompt engineering, we report the higher metric achieved between these two strategies in our main experimental tables (Tab. 1 and Tab. 2). This ensures that the limitations of OV models in our comparison stem from their architectural differences (e.g., lack of visual prompt support) rather than suboptimal inference settings.

B. Performance on 5-shot Prompting

Motivation: Mitigating Visual Instability. In our main experiments (Tab. 1), we observed that the fused prompt mode did not always surpass the single-modality baselines. We attribute this phenomenon to the inherent instability and ambiguity often present in single-shot visual prompts [43]. A single visual exemplar may fail to capture the comprehensive intra-class variance or may introduce noise, resulting in non-representative feature extraction. In such instances, these unstable visual features can act as interference against the clear semantic guidance provided by the text prompt during the fusion process. Therefore, increasing the number of visual prompts serves as an effective strategy to mitigate this instability and construct a more robust visual representation.

Impact of Prompt Scaling. Leveraging the flexibility of SkySense-VITA, we can mitigate this issue by scaling the number of visual prompts (i.e., few-shot prompting) to obtain a more robust visual representation. We report the performance on the test datasets under a 5-shot setting in Tab. S1. The results demonstrate two key findings: (1) Scaling the number of visual prompts significantly improves the overall performance of the visual-dependent modes. (2) Crucially, with the robust visual features aggregated from 5 shots, the fused mode (V+T) outperforms both visual-only and textual-only modes in the two datasets. This confirms that our fusion mechanism is highly effective when provided with reliable features from both modalities.

Table S1. Comparison of 1-shot vs. 5-shot performance. Under the 5-shot setting, the fused mode consistently achieves the best performance. All metrics are mIoU (%).

Dataset	Setting	V-only	T-only	V+T
FBP	1-shot	47.04	**58.87**	57.26
	5-shot	54.04	58.87	**59.28**
iSAID	1-shot	56.07	**61.55**	59.89
	5-shot	58.50	61.55	**62.38**

C. Is Decoupled Training Superior to Independent Training?

In Sec. 4.4, we demonstrated that the proposed decoupling strategy significantly enhances robustness when handling varying prompt modalities during inference. To further investigate whether our unified decoupled training paradigm

Table S2. **Comparison with independent training strategies.** We verify if decoupled joint training outperforms specialized single-modality training. Values are mIoU (%) on FloodNet [35]. "-" denotes unsupported modes.

Training Strategy	Testing Mode (mIoU)		
	V-only	T-only	V + T
Visual-only Training	52.96	-	-
Textual-only Training	-	56.91	-
Fused-only Training	33.12	34.57	59.31
Decoupled (Ours)	**53.19**	**57.52**	**62.19**

yields superior performance compared to training specialized models independently for each modality, we conducted a comprehensive ablation study. We compared our approach against models trained exclusively with specific prompt types (i.e., Visual-only, Textual-only, and Fused-only).

The results, presented in Tab. S2, indicate that our decoupled strategy consistently outperforms the independent training baselines across all testing modes. Specifically, our unified model achieves 53.19% and 57.52% mIoU on visual and textual tasks, respectively, surpassing the specialized models trained solely on those modalities (52.96% and 56.91%). This confirms that the proposed decoupling strategy not only ensures flexibility but also facilitates mutual reinforcement among branches, enabling the model to learn more robust representations than separate training schemes.

D. Additional Implementation Details

Building upon the summary in the main paper, we provide the detailed configurations for the training process. The model is implemented using PyTorch and Detectron2, accelerated by DeepSpeed to ensure efficient large-scale training.

D.1. Training Hyperparameters

For the Pixel-Level In-context Pretraining stage, we utilize the AdamW optimizer with a base learning rate of 1×10^{-4}. The training is conducted with a total batch size of 64 distributed across 8 NVIDIA A100 GPUs (batch size of 8 per GPU). To improve memory efficiency and numerical stability, we employ BFloat16 mixed-precision training. The input images are processed at a resolution of 512×512. We apply standard multi-scale data augmentation, where input images are randomly resized with a scale factor in the range of $[0.7, 1.3]$ followed by a random crop with a ratio of 0.5. The key hyperparameters are summarized in Tab. S3.

D.2. Class-aware Sampling with SGA

A critical component of our training pipeline is the data sampling strategy. Unlike standard image-based sampling which samples images uniformly, we adopt a category-based sampling mechanism. In each iteration, the sampler first selects a target semantic category from the current pool of active classes and subsequently retrieves an image-mask pair containing that category.

Table S3. **Hyperparameters for Pixel-Level In-context Pretraining.**

Configuration	Value
Optimizer	AdamW
Base Learning Rate	1×10^{-4}
Batch Size	64 (8 per GPU)
Precision	BFloat16 (`bf16`)
Image Size	512×512
Augmentation Scale	$[0.7, 1.3]$
DeepSpeed	Enabled

This strategy has two significant implications:

- **Dataset Balancing:** Datasets characterized by high semantic diversity (i.e., a larger number of unique categories, such as iSAID [52] or FLAIR [8]) are statistically sampled more frequently than datasets with fewer categories. This prevents the model from being biased towards large but semantically simple datasets.
- **Integration with SGA:** Our Semantic Granularity Annealing (SGA) strategy directly intervenes in this sampling step. SGA dynamically alters the labels in the dataset according to the hierarchical semantic tree defined in Sec. 3.3 of the main paper. As training progresses, the "active class list" available to the sampler evolves from coarse super-classes (e.g., *Vehicle*) to fine-grained leaf nodes (e.g., *Small Car*, *Large Vehicle*). Consequently, the sampler initially focuses on broad semantic concepts and gradually shifts the distribution towards fine-grained distinctions, realizing a robust coarse-to-fine curriculum.

D.3. Details on Cross-Modal Alignment Loss

The cross-modal alignment loss $\mathcal{L}_{\text{CMA}}$ is used in the Image-level Alignment Pretraining stage to align the SAR encoder with the frozen optical encoder. We adopt the InfoNCE objective [32], which maximizes the similarity between co-registered optical and SAR global features while minimizing similarity with negative samples.

Formally, given a batch of N co-registered optical-SAR pairs $\{(f_{\text{opt}}^{(i)}, f_{\text{SAR}}^{a,(i)})\}_{i=1}^{N}$, where $f_{\text{opt}}^{(i)}$ and $f_{\text{SAR}}^{a,(i)}$ are the ℓ_2-normalized global embeddings, the cross-modal alignment loss is defined as:

$$\mathcal{L}_{\text{CMA}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(f_{\text{opt}}^{(i)}, f_{\text{SAR}}^{a,(i)})/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(f_{\text{opt}}^{(i)}, f_{\text{SAR}}^{a,(j)})/\tau)}, \tag{7}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a temperature hyperparameter. This objective encourages the SAR

encoder to produce embeddings that are aligned with the pretrained optical-text embedding space.

E. Efficiency Analysis

We provide a efficiency analysis from two perspectives: Total Response Time for time-critical applications, and inference speed (FPS) comparison with other foundation models.

Total Response Time Analysis. We define "rapid response" via *Total Response Time* (Annotation + Adaptation + Inference), rather than just inference latency. For urgent tasks such as disaster response, traditional lightweight models (e.g., UNet-ResNet18) require time-consuming data annotation and retraining cycles. In contrast, prompt-based methods like SkySense-VITA enable immediate zero/few-shot inference, significantly reducing the deployment cycle by eliminating adaptation time and reducing annotation effort. Even when excluding the substantial annotation time to strictly compare adaptation and inference delays, Fig. S1 presents a scatter plot on FloodNet comparing this restricted Response Time versus Accuracy. SkySense-VITA achieves the best trade-off, offering high accuracy with minimal delay, even surpassing fully supervised lightweight models.

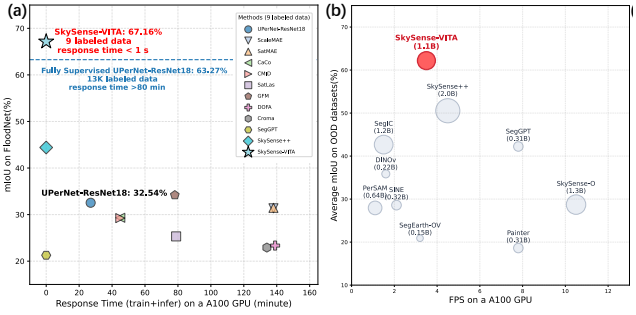

Figure S1. (a) Total Response Time vs. mIoU on FloodNet. (b) FPS vs. Parameters comparison.

Inference Speed Comparison. Fig. S1(b) reports the inference speed (FPS) and model parameters against other zero-/few-shot models. While SkySense-VITA leads in accuracy, we acknowledge that model compression is a promising direction to further approach lightweight specialists in pure inference speed. Currently, SkySense-VITA maintains an acceptable speed of approximately 3.5 FPS on an NVIDIA A100 GPU, which is sufficient for practical remote sensing applications.

F. Extended Comparison with SAM-Series Models

We provide an extended comparison with SAM-series models [14, 36] on intra-image prompting scenarios. While SkySense-VITA is optimized for cross-image in-context segmentation, we evaluate its performance when adapted to the intra-image setting. Fig. S2 shows that: (1) **Robustness:**

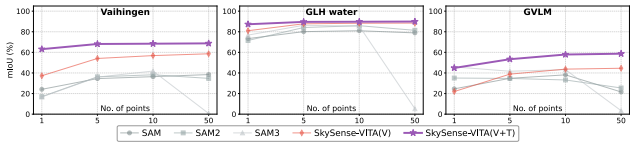

Figure S2. Intra-image comparison with SAM-series models on unseen remote sensing datasets. SkySense-VITA demonstrates superior robustness and prompt fusion capability.

SAM-series models always exhibit performance collapse when provided with an increasing number of point prompts (particularly SAM3), whereas SkySense-VITA remains stable across all settings. (2) **Fusion Advantage:** SkySense-VITA with fused prompts significantly outperforms SAM-series models in all scenarios (e.g., 63.1% vs. SAM3's 16.7% on Vaihingen [11] with 1 point). This demonstrates that fusing textual prompts effectively bridges domain gaps where visual-only generalist models fail.

G. Construction of Hierarchical Semantic Tree

Remote sensing categories inherently exhibit a taxonomic structure (e.g., *Vehicle* $\rightarrow$ *Car* $\rightarrow$ *Small Car*). Standard flat categorization ignores these semantic relations, often leading to confusion among visually similar classes. Inspired by hierarchical comparisons in natural image classification [38], we leverage the commonsense reasoning of Large Language Models (LLMs) to automatically construct a hierarchical semantic tree $\mathcal{T}$ from the flat label set $\mathcal{C}$. This tree serves as the structural basis for our Semantic Granularity Annealing (SGA) strategy described in Sec. 3.3.

Recursive Tree Generation. The construction process is a recursive loop of *Semantic Grouping* and *Discriminative Description*, as outlined in Table S4. First, we encode all category names using the text encoder to obtain initial semantic embeddings. We then employ K-Means clustering to partition the current set of categories into subsets based on their semantic similarity.

For each resulting subset (group), we employ Qwen [56] to generate descriptions. Crucially, we adapt the prompting strategy based on the group size:

- **Abstraction (Coarse Level):** If the group size is large, the LLM is prompted to summarize the *common* visual attributes shared by these categories in remote sensing imagery (e.g., grouping *Forest*, *Grass*, and *Shrub* into *Vegetation*).
- **Discrimination (Fine-grained Level):** If the group size is small, the LLM is prompted to perform explicit *comparisons*. We ask the LLM to identify subtle differences in texture, geometry, and context that distinguish one category from its neighbors (e.g., distinguishing a *Bridge*

Algorithm 1 Hierarchical Semantic Tree Construction

Input: Category Set $\mathcal{C}$, Threshold τ
Output: Semantic Tree $\mathcal{T}$

function BUILDTREE(CurrentNode, Categories)
  $E \leftarrow$ TextEncoder(Categories)
  $\mathcal{S}_1, \ldots, \mathcal{S}_K \leftarrow \text{KMeans}(E, K)$ ▷ *Semantic Grouping*
  for $k \in \{1, \ldots, K\}$ **do**
    if $|\mathcal{S}_k| = 1$ **then**
      return LeafNode($\mathcal{S}_k$)
    else if $|\mathcal{S}_k| > \tau$ **then**
      $P \leftarrow \text{Prompt}_{\text{summary}}(\mathcal{S}_k)$ ▷ *Common features*
    else
      $P \leftarrow \text{Prompt}_{\text{compare}}(\mathcal{S}_k)$ ▷ *Distinctive features*
    end if
    $D \leftarrow \text{LLM}(P)$ ▷ *Generate Description*
    ChildNode $\leftarrow$ Node(Description=D)
    CurrentNode.addChild(BUILDTREE(ChildNode, $\mathcal{S}_k$))
  end for
  return CurrentNode
end function

Table S4. The recursive process for building the semantic tree.

from an *Overpass* based on the presence of water underneath).

These generated descriptions become the nodes of our semantic tree, and the process recurses until individual leaf categories are isolated. Finally, to ensure the reliability of the generated taxonomy, we conducted a manual verification of the entire tree structure. We manually pruned irrelevant or redundant nodes and corrected any hallucinated groupings, ensuring that the final hierarchy is logically sound and aligned with geospatial semantics.

Prompt Engineering for Remote Sensing. Generic prompts often yield descriptions focusing on functional usage or side-view attributes (e.g., cars have wheels), which are irrelevant for satellite imagery. To address this, we design *RS-specific Prompts* that explicitly constrain the LLM's perspective. We instruct the model to focus on "overhead views", "geometric shapes", and "surrounding context". Table S5 showcases the templates used for generating the hierarchy. The constructed tree is as Fig. S3.

H. Dataset Configuration and Prompt Definitions

Dataset Sourcing and Splits. To construct our comprehensive benchmark, we aggregated data from widely recognized public remote sensing datasets, covering both optical and SAR modalities. To ensure the reproducibility of our results and maintain a fair comparison with prior arts, we strictly adhered to the official data splits provided by the original benchmarks whenever available. For datasets where official splits were not explicitly defined, we adopted spatially non-overlapping image splits following standard practices to prevent any data leakage between training and evaluation sets. All data splits will be released to ensure reproducibility. The in-distribution evaluation includes datasets such as FLAIR [8], Potsdam [10], and iSAID [52], while out-of-distribution generalization is evaluated on distinct datasets like Vaihingen [11] and FloodNet [35] to rigorously test domain adaptability.

Table S5. **Prompt Templates for Tree Construction.** We inject domain-specific constraints to ensure descriptions are relevant to remote sensing imagery.

Type	Prompt Template
Summary (Coarse)	"Summarize the common visual features of the following remote sensing categories: { *class_list* }. Focus on their **spectral signature**, **texture**, and **appearance from an overhead satellite view**."
Comparison (Fine)	"Compare the following remote sensing categories: { *class_list* }. Identify the key differences to distinguish them in a top-down aerial image. Focus on: 1. **Geometric Shape**, 2. **Surrounding Context** (e.g., near water), 3. **Scale**."

Textual Prompt Construction. Regarding the textual prompts, we adopted a minimalist strategy to evaluate the model's inherent semantic understanding rather than its sensitivity to prompt engineering. Specifically, we utilized the original raw category names defined in the source annotations (e.g., *Building*, *Forest*, *Small Vehicle*) directly as the input class descriptors. This approach ensures that the text embeddings represent the core semantic concepts of the classes, facilitating a direct and robust alignment with the visual features.

I. Additional Qualitative Visualizations

To provide a more comprehensive assessment of SkySense-VITA's capabilities beyond the quantitative results in the main paper, we present additional qualitative visualization results in Fig. S4 and Fig. S5. These figures demonstrate the model's segmentation performance across distinct geographic scenes and diverse semantic categories.

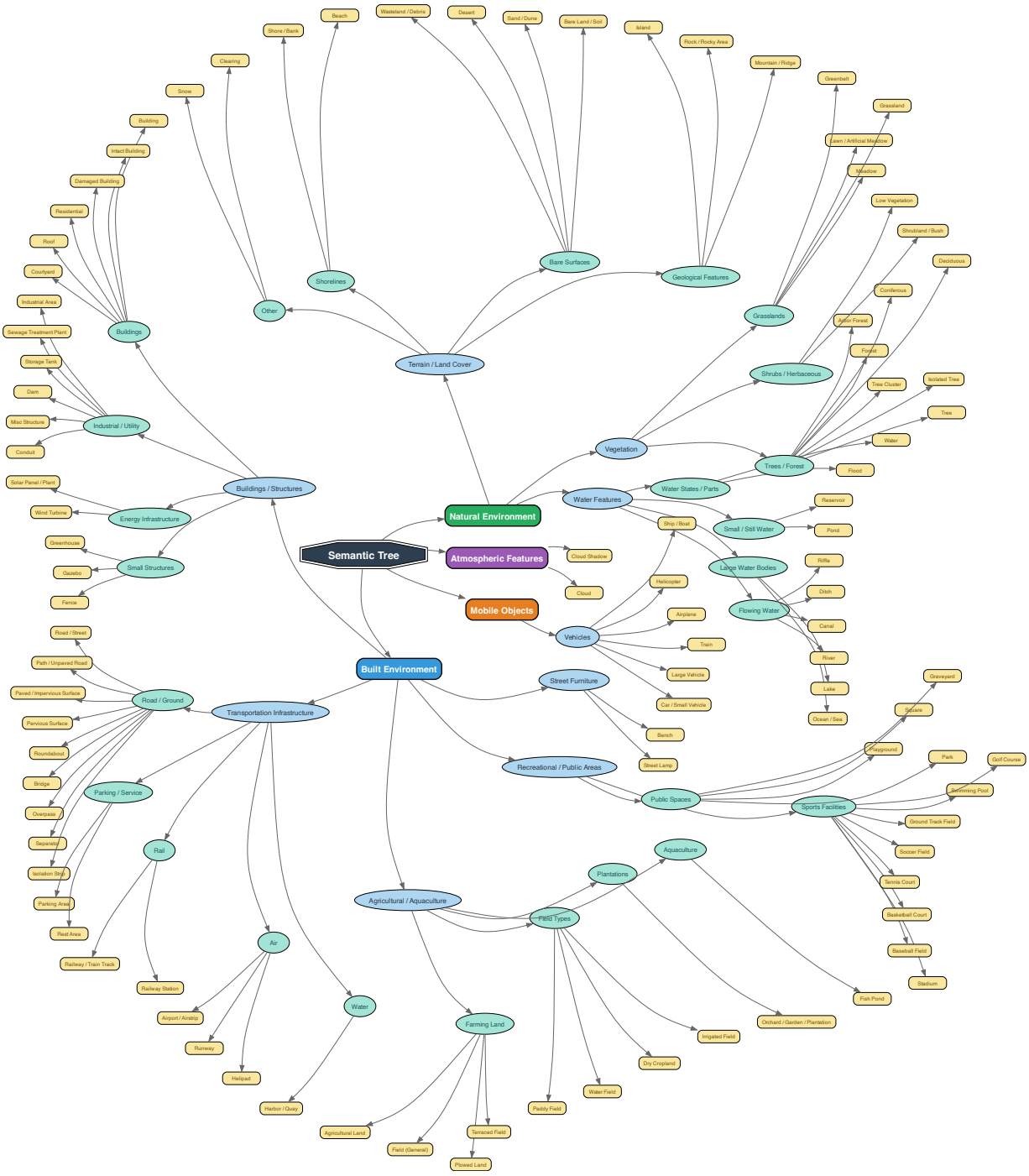

Figure S3. **Visualization of Hierarchical Semantic Tree.**

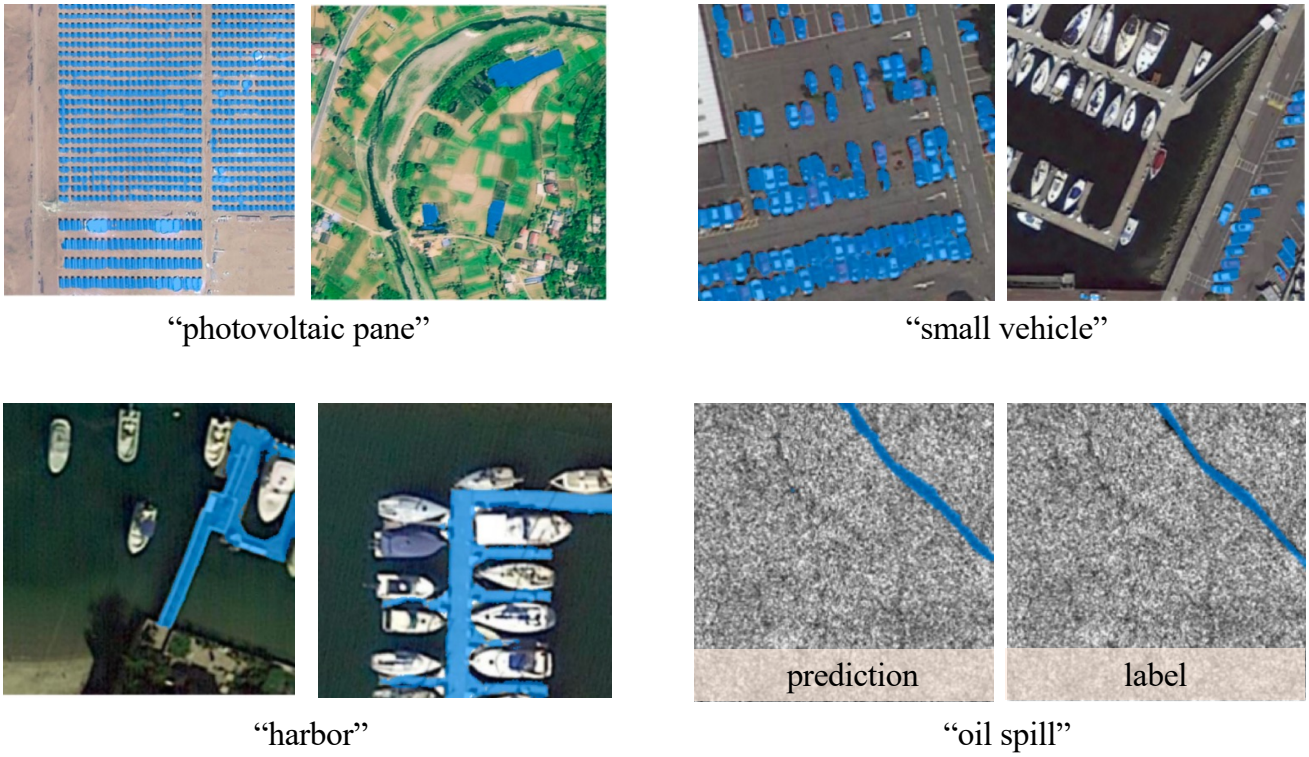

Figure S4. **Qualitative results1.**

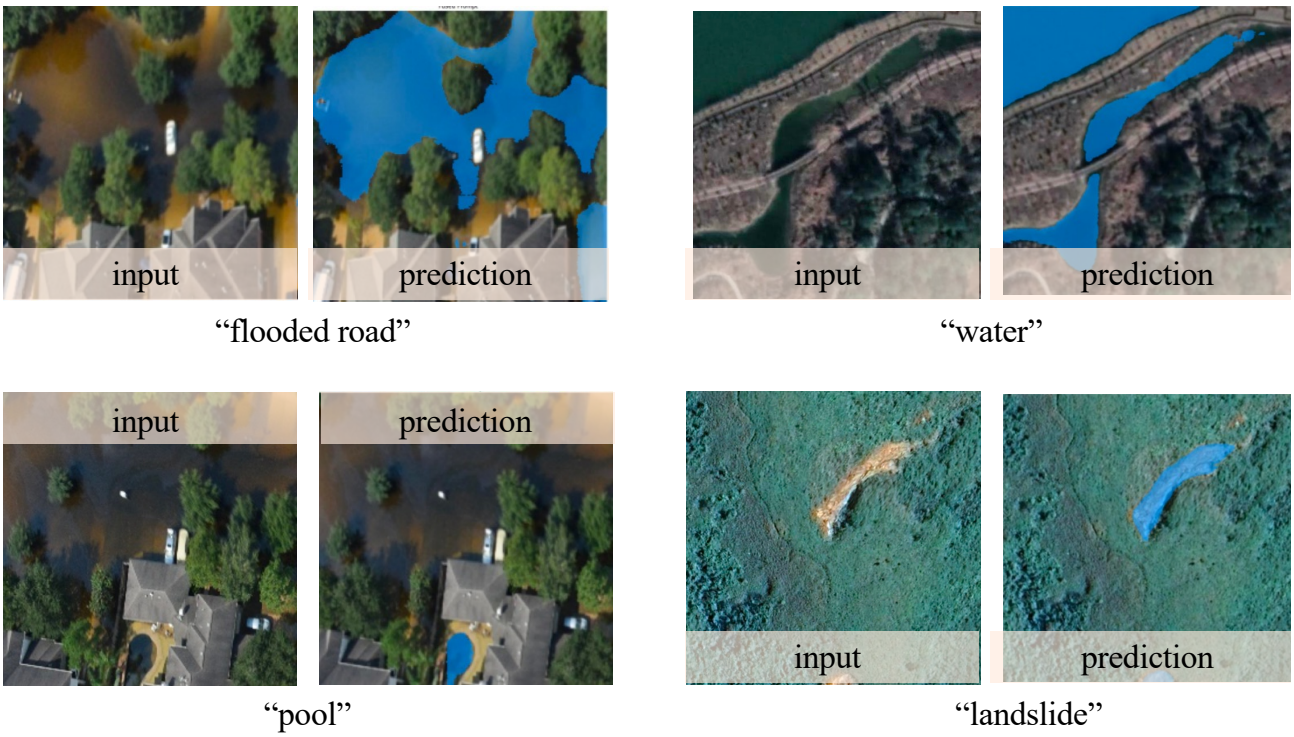

Figure S5. **Qualitative results2.**