Adapting Vehicle Detectors for Aerial Imagery to Unseen Domains with Weak Supervision Supplementary Material

Xiao Fang[1], Minhyek Jeon[1], Zheyang Qin[1], Stanislav Panev[1], Celso de Melo[2], Shuowen Hu[2], Shayok Chakraborty[1,3], Fernando De la Torre[1]
[1]Carnegie Mellon University, [2]DEVCOM Army Research Laboratory, [3]Florida State University,
{xfang2, minhyekj, zheyangq, spanev}@andrew.cmu.edu,
{celso.m.demelo.civ, shuowen.hu.civ}@army.mil, shayok@cs.fsu.edu, ftorre@cs.cmu.edu

Algorithm 1 Summary of the pipeline

Input: Source domain data $\mathbf{D}^{\mathcal{S}}$, target domain data $\mathbf{D}^{\mathcal{T}}$

1: Fine-tune Stable Diffusion [45] on both $\mathbf{D}^{\mathcal{S}}$ and $\mathbf{D}^{\mathcal{T}}$ using domain-specific prompts
2: Generate synthetic images $x^{\mathcal{GS}}$ and extract stacked cross-attention maps $\widetilde{A}^{\mathcal{GS}}$ for the source domain
3: Generate synthetic images $x^{\mathcal{GT}}$ and extract stacked cross-attention maps $\widetilde{A}^{\mathcal{GT}}$ for the target domain
4: Train a detector $F^{\mathcal{S}}(\cdot;\theta)$ on $\mathbf{D}^{\mathcal{S}}$
5: Run $F^{\mathcal{S}}(\cdot;\theta)$ on $\mathbf{D}^{\mathcal{S}}$ to obtain pseudo labels $y^{\mathcal{GS}}$
6: Train a detector $F^{\mathcal{A}}(\cdot;\theta)$ on $\left(\widetilde{A}^{\mathcal{GS}}, y^{\mathcal{GS}}\right)$
7: Run $F^{\mathcal{A}}(\cdot;\theta)$ on $\widetilde{A}^{\mathcal{GT}}$ to obtain pseudo labels $y^{\mathcal{GT}}$
8: Train the final detector $F^{\mathcal{T}}(\cdot;\theta)$ on $\left(x^{\mathcal{GT}}, y^{\mathcal{GT}}\right)$
9: Test $F^{\mathcal{T}}(\cdot;\theta)$ on real target domain images from $\mathbf{D}^{\mathcal{T}}$

A. Methods

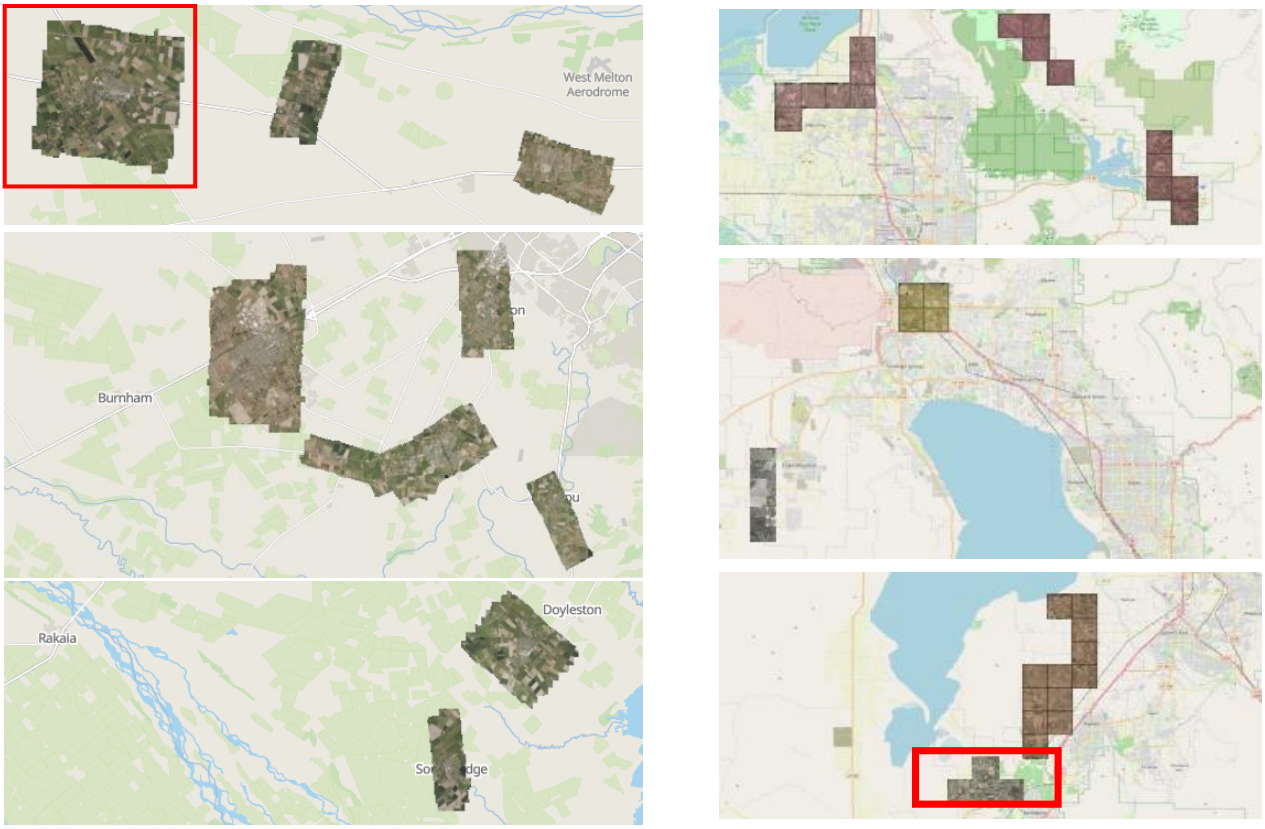
In this section, we present a summary of steps regarding our proposed pipeline, as shown in Algorithm 1.

B. Datasets

In this section, we provide additional details and examples of the LINZ and UGRC datasets. Figure 8 illustrates the geographic regions from which our data samples were obtained. The LINZ online platform captured aerial imagery from nine areas in Selwyn. For dataset construction, we designated one of these nine areas as the testing region, from which test set images were sampled, while the remaining eight areas were used for training and validation samples. Similarly, the UGRC online platform collected aerial imagery from seven regions in Utah. One of these seven areas was designated as the testing region, with the remaining six serving as sources for training and validation data. This spatial partitioning strategy ensures that our datasets do not suffer from data leakage, as the training and testing areas are spatially independent. Within each area, we randomly sample square images of size 112 px $\times$ 112 px. Due to the sampling strategy, a single vehicle can appear in multiple images. Figure 7 presents the subcategories within the *small vehicle* class. Except for the *Pickup truck* category, all other small vehicle subcategories fall under the broader

Figure 7. Vehicles belonging to the object class *small vehicle*.

classification of cars. Accordingly, we utilize the word "car" in prompts to guide the Stable Diffusion [45] during image generation, leveraging its pre-trained perceptual understanding related to cars. Figure 9 provides visual examples of both LINZ and UGRC images, highlighting distinct visual characteristics: UGRC includes a notable proportion of off-road vehicles, reflecting its sandy and rocky terrain, while LINZ images primarily feature urban vehicles within structured road networks. Lastly, Figure 10 compares the performance of four object detectors under two settings: cross-dataset evaluation (trained on LINZ and tested on UGRC) and within-dataset evaluation (trained and tested on UGRC). The results show that within-dataset performance surpasses cross-dataset performance by at least



(a) Selwyn (New Zealand)

(b) Utah (USA)

Figure 8. Geographic regions where we construct LINZ and UGRC datasets. Red bounding boxes denote the testing area.

25.7% higher AP_{50} across all detectors, underscoring a significant domain gap between the two datasets.

Method	Vision Backbone	LINZ→UGRC Precision(%)	LINZ→UGRC Recall(%)
Vision Large Language Model			
Gemini 1.5 Flash [55]	-	2.9	44.5
Gemini 2.0 Flash-Lite [54]	-	6.6	26.3
InternVL3-8B [74]	InternViT [74]	4.7	22.0
Qwen2.5-VL-7B [1]	ViT [1]	0.4	4.8
DeepSeek-VL2-Tiny [62]	SigLIP-SO400M [67]	9.2	26.8
LLaVA-NeXT [29]	CLIP ViT [41]	5.5	4.7
Ours	Faster R-CNN [44]	**63.8**	**68.2**
Ours	YOLOv5 [14]	**67.2**	**67.3**
Ours	YOLOv8 [43]	**70.0**	**76.3**
Ours	ViTDet [24]	**72.0**	**67.1**

Table 3. Comparison between our methods and VLLMs on UGRC dataset. We report the precision and recall metrics.

C. Limitation of Foundation Models

In this section, we provide more examples of limitations of various types of foundation models, including open-set detectors, diffusion models, and vision large language models, when detecting small vehicles in aerial view images.

C.1. Open-set Detectors

As shown in Figure 13, Grounding-DINO [30] often detects cars in background images. OmDet-Turbo [71] and OWLv2 [33] often produce false positives by misclassifying objects such as rectangular tanks and boxes, which share visual similarities with cars. OWL-ViT [32] fails to detect any cars in aerial images, highlighting its limitations in this specific context. These findings underscore the challenges faced by open-set object detectors in accurately identifying vehicles in aerial imagery.

C.2. Diffusion Models

As shown in Figure 15 (a), when employing the pre-trained Stable Diffusion [45] with the prompt "an aerial image with cars in Utah", the model fails to understand the geographical reference to "Utah" and does not generate images reflective of the state's landscape. Additionally, Stable Diffusion struggles with generating small objects such as cars in aerial images, resulting in low-quality vehicle depictions. ControlNet [68] may also fail to effectively guide the gener-

Figure 9. **Examples of images from our real-world datasets.** (*first row*) LINZ images containing small vehicles; (*second row*) LINZ images without vehicles; (*third row*) UGRC images containing small vehicles; (*fourth row*) UGRC images without vehicles;

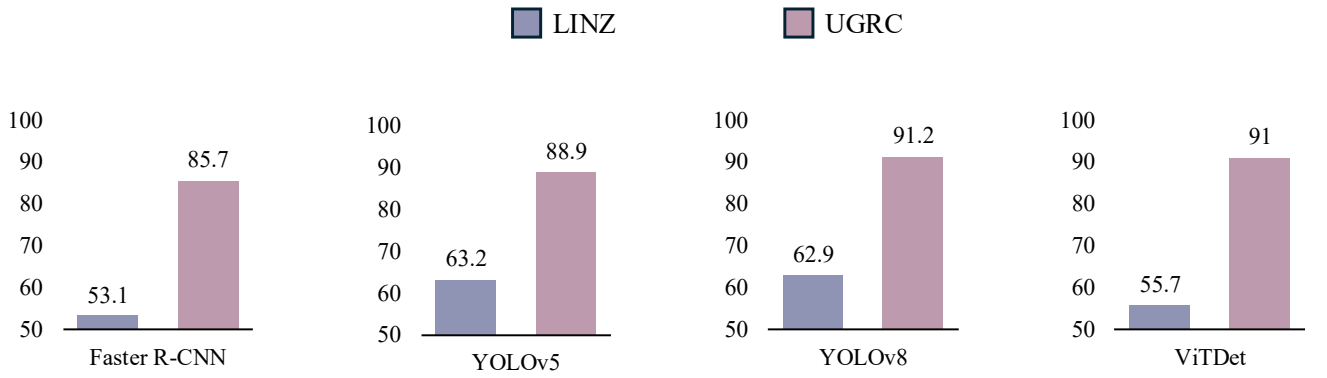
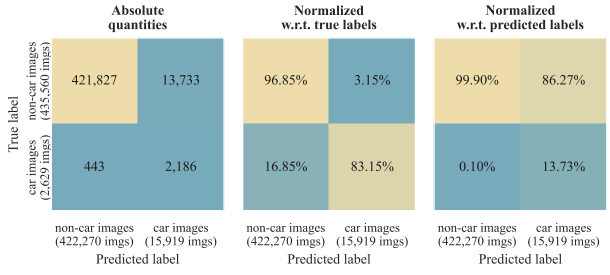


Figure 10. **Comparison between cross-dataset generalization and within-dataset performance.** The purple bars represent the model trained on the LINZ dataset and evaluated on the UGRC dataset, while the pink bars correspond to both training and testing conducted on the UGRC dataset. We report the AP_{50} result.

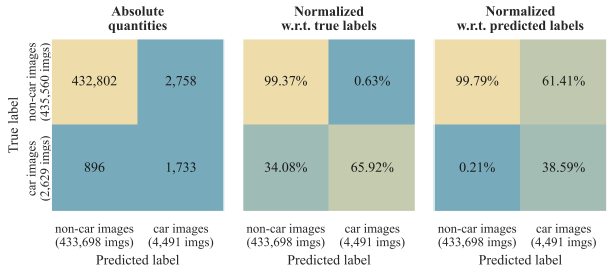
ation direction, leading to outputs that do not align with the provided edge and segmentation map conditions. As shown in Figure 15 (b), DoGE [58] frequently fails to generate cars and surroundings in the precise locations corresponding to the input LINZ image under the guidance of canny edges and semantic segmentation maps. Moreover, without finetuning on UGRC images, DoGE fails to encode the domain difference by modeling the average CLIP [41] image embedding difference and is unable to produce images that accurately reflect Utah's landscape. Figure 15(c) illustrates that even after fine-tuning GLIGEN [25] on UGRC data, the model frequently fails to comply with the specified bounding box layout conditions. It often generates extraneous cars outside the designated bounding boxes or omits cars within the expected bounding areas. These limitations highlight the challenges associated with ensuring diffusion models faithfully adhere to spatial and semantic constraints in conditioned image generation.

C.3. Vision Large Language Models

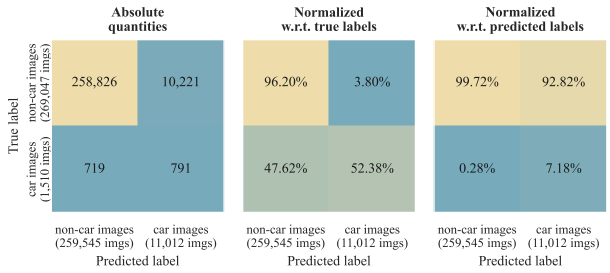
We evaluate the capabilities of Vision Large Language Models (VLLMs) for car presence classification and center localization (VLLMs), as shown in Figure 11 and Table 3. For classification, BLIP2 [21] generates captions based on



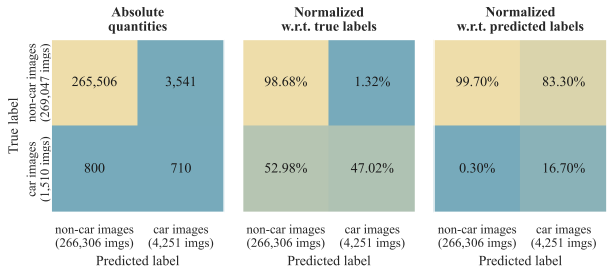
(a) LINZ - BLIP2 Image Captioning



(b) LINZ - Kosmos2 Image Captioning



(c) UGRC - BLIP2 Image Captioning



(d) UGRC - Kosmos2 Image Captioning

Figure 11. VLMs captioning capabilities analysis tested on LINZ and UGRC datasets.

the images while Kosmos2 [39] is prompted to complete the sentence "an aerial image of {}". We consider a model to have predicted the presence of a car in an image if the generated caption includes the word "car". As shown in Figure 11 (a), only 13.73% of the images predicted to contain cars by BLIP2 are true positives in the LINZ dataset. A similar trend is observed in Figure 11 (b), (c) and (d), where the true positive rates for predicted car images are 38.59%, 7.18%, and 16.70%, respectively. Furthermore, among all images that contain cars, only 52.38% are correctly identified by BLIP2 in the UGRC dataset, as shown in Figure 11 (c). A similar trend is observed in Figure 11 (d), where 47.02% of all images that actually contain cars are correctly predicted by Kosmos2. Figure 12 supports that VLLMs struggle to accurately detect cars in aerial imagery by presenting the F1 score, Precision, and Recall metrics of their classification performance.

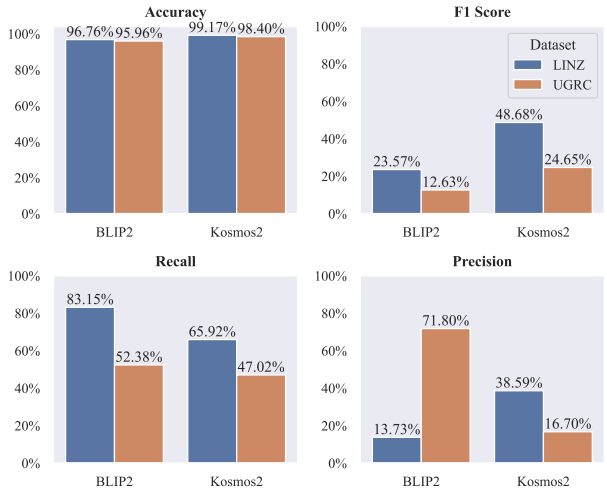


Figure 12. Two popular VLLMs (BLIP2 and Kosmos2) tested as zero-shot image car presence classifier. The severe imbalance of the positive and negative classes causes the high levels of accuracy, which is deceptive. F1 score, Precision and Recall metrics clearly show that the classification quality is less than ideal.

For localization, we assess the detection performance of VLLMs as shown in Table 3. Since VLLMs do not provide confidence scores for the predicted bounding boxes, we define *detection accuracy* as the proportion of predicted bounding boxes that achieve an Intersection over Union (IoU) greater than 0.5 with at least one ground truth bounding box. Based on this definition, we compute precision and recall, which are reported in Table 3. To establish pseudo labels on the UGRC test set for our method, we set the detection threshold according to the highest F1 score achieved by each detector on UGRC test set. Under this setting, VLLMs exhibit significantly lower performance compared to our method, often producing a large number of false positives in aerial imagery. These findings highlight the current limitations of pre-trained VLLMs in accurately detecting and localizing vehicles in aerial imagery.

D. More Implementation Details

D.1. Decision circle and Pseudo-bounding box label

In this section, we provide a detailed explanation for defining the pseudo-bounding box size to be 42.36 px. As illustrated in Figure 14 (b), any point p within the region en-

Figure 13. Failure cases of Open-set detectors. (a) Detection results of Grounding-DINO. (b) Detection results of Omdet-Turbo. (c) Detection results of OWLV2. (d) Detection results of Owlvit. The blue bounding boxes with dotted lines denote the predicted pseudo bounding box labels while the dots denote the predicted car centers.

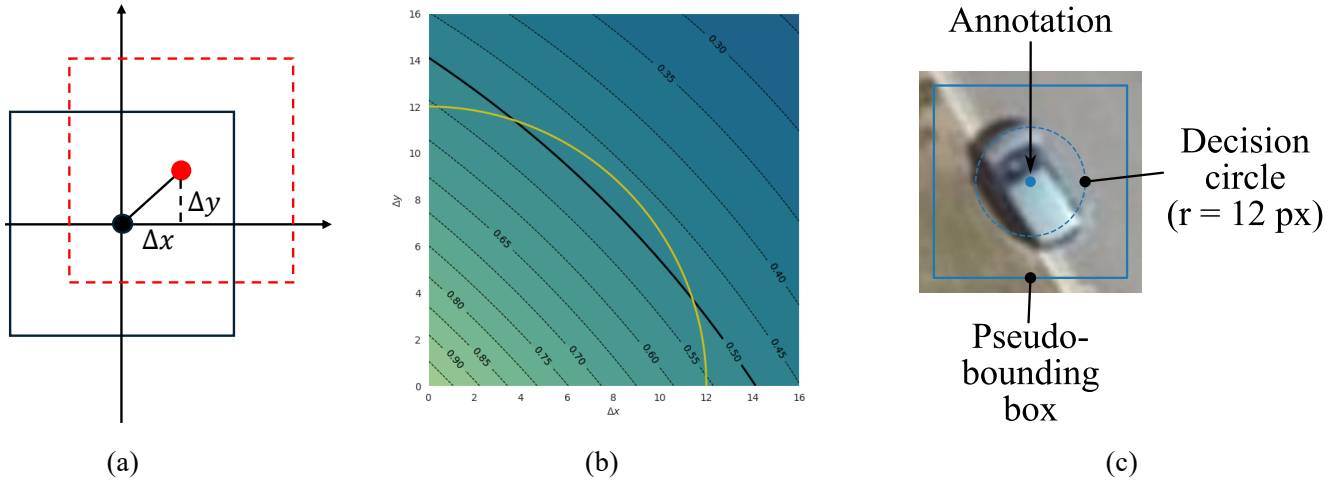


Figure 14. Illustration of how we obtain the 42.36 px bounding box size. (a) The black bounding boxes denote the ground truth pseudo bounding box labels while the red bounding boxes denote the predicted pseudo bounding box labels. The dots denote the corresponding centers. Δx and Δy denote the x-and y-coordinate difference between the ground truth center and the predicted center. (b) Isocontour of Intersection of Union (IoU). The yellow arc is $\frac{1}{4}$ of the decision circle with a 12 px radius while the black curve represents the isocontour where IoU $= 0.5$. (c) An example of a decision circle with a radius of 12 px centered at the car's center with the corresponding 42.36 px pseudo-bounding box.

closed by the isocontour of IoU $= \alpha$ represents a predicted pseudo-bounding box centered at p with an IoU $\geq \alpha$ relative to the ground truth pseudo-bounding box. Ideally, all true positive predicted centers should be contained within the decision circle. However, no isocontour perfectly fits the arc of the decision circle. To minimize this discrepancy, we determine the square pseudo-bounding box size a such that the area enclosed by the isocontour at IoU $= 0.5$ matches $\frac{1}{4}$ of the decision circle. Let Δx and Δy denote the x- and y-coordinate differences, respectively, between the centers of the ground truth and predicted pseudo-bounding boxes, as shown in Figure 14 (a). Without loss of gener-

(a) Stable Diffusion (b) ControlNet + Domain Gap Embedding (c) GLIGEN

Figure 15. Failure cases of diffusion models. (a) Images generated by pre-trained Stable Diffusion V1.4. (b) Images generated by pre-trained Stable UnCLIP [42] with canny edge maps, semantic segmentation maps, and average CLIP image embedding difference between the LINZ and the UGRC dataset as conditions. From left to right: Real images from LINZ dataset, edge maps, semantic segmentation maps, and synthetic images. (c) Images generated by GLIGEN. Left: Real images from the LINZ dataset. Right: Synthetic UGRC images. The bounding boxes that have the same color in left and right images correspond to the same location.

Backbone	Task	Stage 1		Stage 2		Stage 3	
		bs	lr	bs	lr	bs	lr
Faster-RCNN	LINZ → UGRC	1024	0.2	192	0.02	512	0.2
Faster-RCNN	DOTA → UGRC	1024	0.2	192	0.001	1024	0.2
YOLOv5	LINZ → UGRC	1600	0.001	192	0.0001	2400	0.001
YOLOv5	DOTA → UGRC	2400	0.001	192	0.0001	2400	0.001
YOLOv8	LINZ → UGRC	4096	0.001	192	0.0001	1024	0.001
YOLOv8	DOTA → UGRC	4096	0.001	192	0.0001	2048	0.001
ViTDet	LINZ → UGRC	192	0.001	96	0.0001	192	0.001
ViTDet	DOTA → UGRC	192	0.0001	96	0.001	192	0.0001

Table 4. Training parameters of each stage, where "bs" denotes the batch size and "lr" denotes the base learning rate, which will be scaled during training based on batch size following the MMDetection framework.

ality, we assume the predicted pseudo-bounding box center lies in the first quadrant. The IoU can then represented as $\text{IoU} = \frac{(a-\Delta x)(a-\Delta y)}{2a^2-(a-\Delta x)(a-\Delta y)}$. By setting $\text{IoU} = 0.5$, we solve for Δy in terms of Δx, treating a as a constant, which can be represented as $\Delta y = \frac{a(a-3\Delta x)}{3(a-\Delta x)}$. We then integrate Δy with respect to Δx to compute the area under the isocontour of $\text{IoU} = 0.5$, which is a function of a. Finally we equate this integral to $\frac{1}{4}$ of the area of the decision circle and solve for a.

D.2. Multi-Stage Training

In this section, we provide more details regarding the training process of the detectors. As outlined in Sec. 3.3, the labeling of synthetic target domain (UGRC) images is conducted in three stages. In the first stage, we train a detector F^S on fully annotated real source domain data (LINZ or DOTA) and subsequently generate pseudo labels for the synthetic source domain images. In the second stage, we train another detector F^A on the multi-channel cross-attention maps of synthetic source domain images and use it to predict pseudo labels for the multi-channel cross-attention maps of synthetic target domain images. Finally, in the third stage, we train a detector F^T on the synthetic target domain images. For Faster-RCNN [44], we use ResNet50 [10] as backbone. For YOLOv5 [14], we utilize the YOLOv5-M variant, while for YOLOv8 [43], we employ the YOLOv8-M variant. For ViTDet [24], we disable the mask head. In all training stages, we scale the image resolution to 128 px $\times$ 128 px, as YOLOv5 requires input dimensions to be multiples of 32. The specific training parameters for each stage are provided in Table 4. Except for these adjustments, we adhere to the MMDetection [2] framework for implementation.

Backbone	$A_c + A_{fg}$	$A_c + A_{bg}$	$A_c + A_{fg} + A_{bg}$
YOLOv5 [14]	63.7	65.5	**68.8**
YOLOv8 [43]	69.1	73.1	**75.4**

Table 5. Comparison of different cross-attention map configurations for adaptation from LINZ to UGRC.

E. More Ablation Studies

In this section, we present two additional ablation studies to further validate the effectiveness of our proposed pipeline. First, we investigate the impact of stacking different combinations of cross-attention maps. Specifically, we compare our approach with alternative configurations that stack two channels of the object category cross-attention map A_c and one channel from either the learned foreground cross-attention map A_{fg} or background cross-attention map A_{bg}, ensuring compatibility with object detectors that accept only three-channel inputs. As shown in Table 5, integrating both background and foreground information, as in our method, yields the best performance in AP_{50}.

Second, we analyze the effectiveness of our two-stage design, which first fine-tunes Stable Diffusion [45] on both source and target domain datasets, and then introduces learnable tokens to extract cross-attention maps that capture both foreground and background information. We compare this design with a one-stage baseline that jointly fine-tunes Stable Diffusion and learns tokens simultaneously. The two-stage setup is motivated by the limited localization ability of unseen prompts in pre-trained Stable Diffusion models when applied to aerial view images. For instance, prompts such as "an aerial view image with cars in Utah" often fail to localize vehicles, leading to inaccurate attention and suboptimal token learning. By first fine-tuning the model to better align the concept of "cars" with actual vehicle locations, we enable subsequent token learning to more precisely focus on relevant regions. Experimental results show that the one-stage pipeline yields an 8.5% lower AP_{50} on YOLOv5 [14] when adapting from LINZ to UGRC.