Low-Rank Prompt Adaptation for Open-Vocabulary Object Detection

Supplementary Material

Zekun Zhang*
zekzhang@cs.stonybrook.edu
Stony Brook University

Vu Quang Truong*
vuquang27102001@gmail.com

Minh Hoai
mh.nguyen@adelaide.edu.au
The University of Adelaide

This is the supplementary material document for the paper *Low-Rank Prompt Adaptation for Open-Vocabulary Object Detection* accepted by **The 4th Workshop on What is Next in Multimodal Foundation Models?** at the **International Conference on Computer Vision 2025**. In this document, we show more experiment results that cannot fit in the main paper. In Sec. 1, we provide more implementation details of the baseline methods used in the main paper. In Sec. 2, we analyze the inference efficiency of the enhancer. In Sec. 3, we present a qualitative analysis with Scenes100, HOIST, EgoPER, OV-COCO, and Rareplanes. In Sec. 4, we compare the proposed method of adapting OVD with adapting closed-vocabulary object detection models on different datasets. In Sec. 5, we present additional results of our enhancer on ODinW-13 dataset. In Sec. 6, we provide the mapping from video IDs used in the qualitative analysis to the corresponding original IDs for result reproduction purposes.

1. More implementation details

This section provides the implementation details of baseline methods in the main paper, which are BitFit [2], Prompt Tuning [9], LoRA [5], LoSA [14], and Res-Tuning [6].

In the BitFit [2] paper, the authors trained the bias terms of the BERT model. We apply the same implementation and train the bias terms of the BERT backbone of GroundingDINO.

With Prompt Tuning [9], we insert five learnable tokens at the start of the prompt sequence. Specifically, the shape of the learnable part is $5 \times d$, with d as the embedding dimension of the BERT backbone.

With LoRA [5], we apply the typical practice of it, where LoRA layers are inserted along all the query, key, and value-embedding layers of both backbones. The rank r of LoRA is set as 16, and the scaling parameter α is set as $1/16$.

With LoSA [14], we follow the best result in the original paper, where the authors use the outputs of encoder layers as inputs for the side network. Here, we attach the side network to the transformer head and use the outputs of the head's encoder layers as inputs for the side network. The rank r of LoSA is set as 16, and the scaling parameter α is initialized as $1/16$.

With Res-Tuning [6], we also follow the best method in the original paper. In this method, three adapter-like tuners are attached to each transformer encoder layer, referred to in the paper as Tri-Res-Tuner. They are attached in parallel with the Multi-Head Attention block, the Feed Forward Network, and the whole encoder layer itself. Following that, we attached Res-Tuners in the transformer head's encoder layers. The rank r of the tuners is set as 16, and the scaling parameter α is initialized as $1/16$.

2. Inference Efficiency of Enhancer Module

The proposed prompt feature enhancement module adds less than 0.1% to the total parameters of GroundingDINO and is efficient to train in the adaptation setting. However, it may still increase the model's inference latency. Here, we measure the inference latency of the vanilla GroundingDINO model and models using different adaptation methods. For transformer models, computational complexity depends on the input sequence length; for GroundingDINO, this depends on the number of tokens in the prompt and the resolution of the input image. We use an image resized to 750×1333 from the EgoPER dataset and test it with three prompts (I, II, and III) from Table 1 of the main paper and a detailed prompt from Sec. 4.4.4 of the main paper, referred to as prompt IV. All experiments are conducted on an NVIDIA V100 GPU, and the inference latency results are shown in Table 1. Results indicate that the proposed lightweight enhancer achieves the fastest inference time among adaptation methods, adding minimal latency to the model.

3. Qualitative analysis

We visually compare how the behavior of the vanilla GroundingDINO model changes after adding the adaptive

*Equal contribution.

Prompt	# tokens	Inference latency (ms) ↓				
		Vanilla	LoRA [5]	Res-Tuning [6]	LoSA [14]	Enhancer
I	5	148.9	161.2	155.4	153.8	**149.4**
II	9	152.5	160.6	158.5	157.5	**155.8**
III	22	169.6	175.4	171.3	169.6	**169.6**
IV	95	186.4	201.1	194.5	195.1	**190.7**

Table 1. Inference latency (in milliseconds) of vanilla GroundingDINO and GroundingDINO adapted with different methods using various prompts is shown in Sec. 2. The column # tokens indicates the token count in each prompt, and the column Increase shows the latency added by the enhancer module. All inferences use the same input image for consistency. The best result among adaptation methods is in **bold**. The low-rank enhancer with $r = 8$ adds minimal latency overhead and achieves the fastest speed among the methods.

enhancer on Scenes100, HOIST, EgoPER, OV-COCO, and RarePlanes. Scenes100, HOIST, and OV-COCO are "hard" datasets since the base model has a very low performance. The enhancer with $r = 16$ is used in this analysis. For each dataset, we choose the initial prompt that is simple but still obtains relatively high AP^m before adaptation. This simulates the situation when the detector user has some experience and familiarity with the dataset. Specifically, we use "person · vehicle ·" on Scenes100, "hand-held object ·" on HOIST, "kitchen object ·" for EgoPER, "coco objects ·" for OV-COCO, and "planes ·" for RarePlanes. We show the detected object bounding boxes with a score higher than θ^*, where θ^* is the score threshold that maximizes the F_1 score at IoU=0.75 in the precision-recall evaluation per the COCO protocol [11].

The comparison on several videos from Scenes100 is shown in Fig. 1a. It is clear that GroundingDINO misses many instances of *vehicle* objects from the prompt phrase vehicle. This is possibly caused by the fact that in the training set of GroundingDINO, most of the vehicles are labeled by other words such as car, truck, SUV, etc. This type of bias is very common in datasets [3, 19]. And the bias transfers to models that are trained on those datasets. Our proposed method can adaptively learn to compensate for such bias and identify different types of vehicles.

As shown in Fig. 1b, detecting the objects in hand in HOIST dataset is challenging. To get correct results, a certain level of understanding of the underlying physical world is needed to determine if an object is held by hand. Before adaptation, even with a prompt specifying hand-held objects, the GroundingDINO model still tends to label most objects in the image. After adaptation, the enhancer enables the model to reason about the relation between objects and hands, and find the correct hand-held objects. However, given the difficulty of the task, it can still treat objects close to hands as objects held by hands, as in video HOIST-004. Or still misidentify the object when the hand is blurry as in video HOIST-005.

The detection performance on EgoPER videos is compared in Fig. 1c. The model can already locate most of the target objects correctly before adaptation. However, the proposed method can still help the model to make more fine-grained decisions, such as detecting the hands in the videos or identifying the bowl in video EgoPER-004.

The qualitative results of OV-COCO are presented in Fig. 1d. The proposed enhancer can filter out false positive bounding boxes as in images OV-COCO-002 and OV-COCO-004. It can also detect objects missed by the unadapted base model, such as OV-COCO-001, OV-COCO-003, OV-COCO-004, and OV-COCO-005. However, it still fails to detect when the object is too blurred, like the cars in OV-COCO-005, or focuses too much on the class with dominant quantity, such as the class "person" in OV-COCO-001, and misses the "clock" in the background.

Fig. 1e shows the qualitative results of RarePlanes. RarePlanes is a "hard" dataset for the base model since it is an aerial imagery dataset, which is significantly different from the datasets used for model pretraining. Hence, the model usually misses the objects, as in RarePlanes-004. However, in many cases, the enhancer helps the model to detect the objects missed by the unadapted model, as in RarePlanes-001 and RarePlanes-002, or remove false-positive boxes, as in RarePlanes-003 and RarePlanes-005.

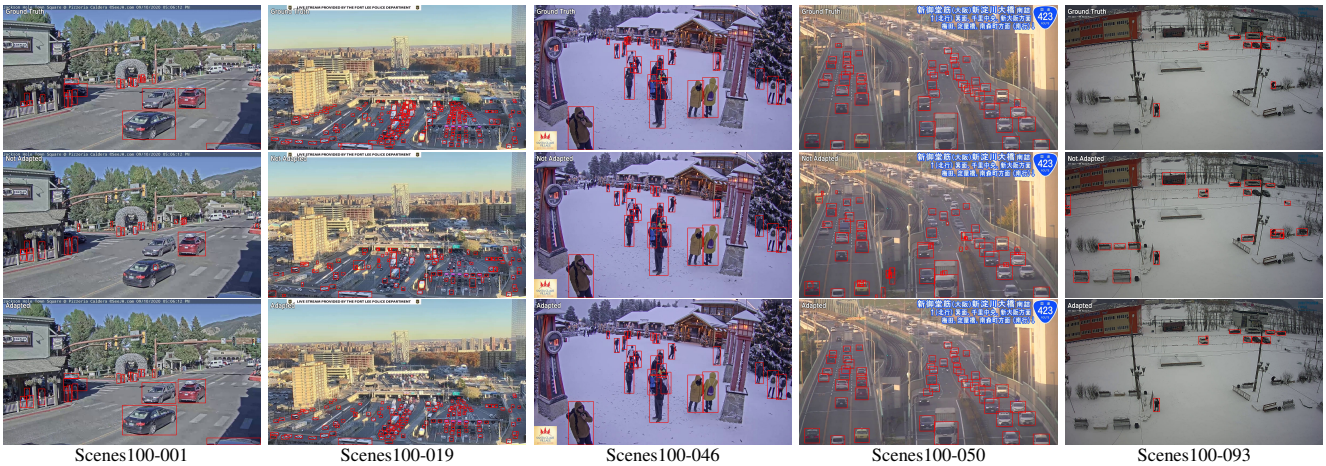
We visually verify that the proposed prompt enhancer module can learn to guide GroundingDINO to the correct direction to improve detection precision from a limited number of sample images.

4. Adaptation of Closed-Vocabulary Detectors

Method	Scenes100 [18]	EgoPER [8]	HOIST [15]	OV-COCO [1]	RarePlanes [17]
Faster-RCNN ResNet-50	40.38	59.38	18.00	32.99	55.94
Faster-RCNN ResNet-101	39.73	59.23	18.43	34.70	56.32
YOLOv8s	16.43	21.23	3.43	13.97	8.33
YOLOv8l	26.82	24.43	9.58	21.04	24.62
GroundingDINO + Enhancer, $r = 32$	**55.97**	**68.46**	**38.51**	**39.78**	**56.54**

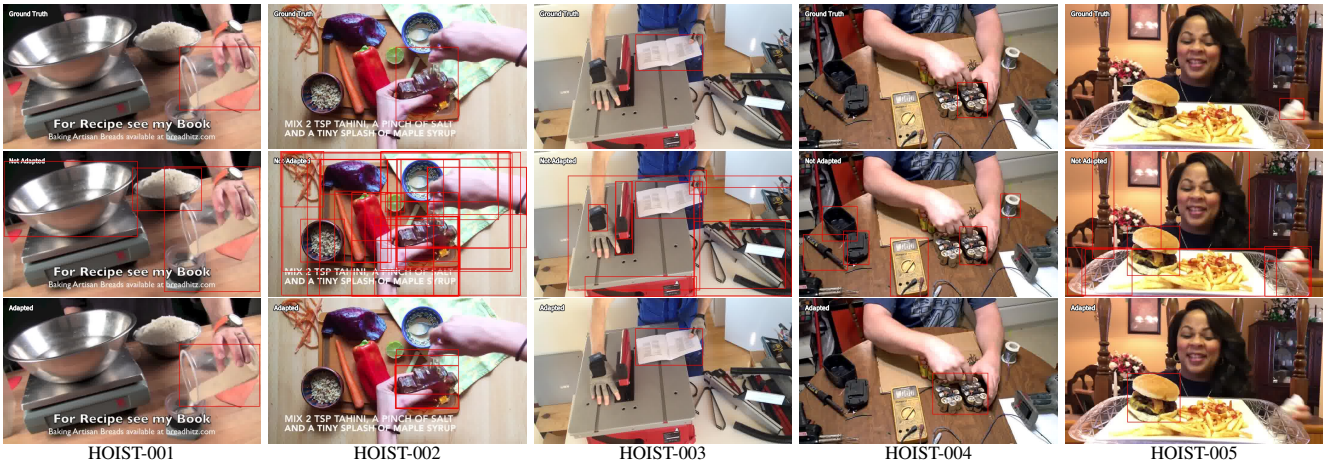
Table 2. Detection AP^m of closed-vocabulary detectors Faster-RCNN [16] and YOLOv8 [7] on different datasets. For each dataset, the average AP^m from the proposed GroundingDINO-based Prompt adaptation method initialized from three prompts in the main paper is shown for comparison. The enhancer with $r = 32$ is used. The best result in each dataset is marked in **bold**. Closed-vocabulary detectors achieve significantly lower AP^m compared to the proposed method.

We try to treat the target objects in each dataset as an object category, and adapt state-of-the-art closed-vocabulary detection models to detect them. We use the same adaptation setting and the same number of training iterations, learning rate, and batch size for fair comparison. Specifically, we use Faster-RCNN [16] model with ResNet-50 or ResNet-101 [4] backbone and feature pyramid network [12], and YOLOv8 [7] of small (YOLOv8s) or large (YOLOv8l) variants. Both models are trained on MSCOCO



Scenes100-001 Scenes100-019 Scenes100-046 Scenes100-050 Scenes100-093

(a) Results on Scenes100 dataset [18] using initial prompt "person · vehicle ·". Bounding boxes that overlap with the pre-defined non-annotation regions are removed.



HOIST-001 HOIST-002 HOIST-003 HOIST-004 HOIST-005

(b) Results on HOIST dataset [15] using initial prompt "hand-held object ·".

Figure 1. Qualitative results on several videos from different datasets. Each column of three images is from one video. The video IDs of HOIST are renamed for better readability. Please refer to the supplement for the original IDs. The images show the ground-truth bounding boxes, the detected bounding boxes from the pre-trained GroundingDINO model, and the detected bounding boxes from the model with an adaptively trained prompt enhancer. Please refer to Sec. 3 for more details. This figure is best viewed on color screens.

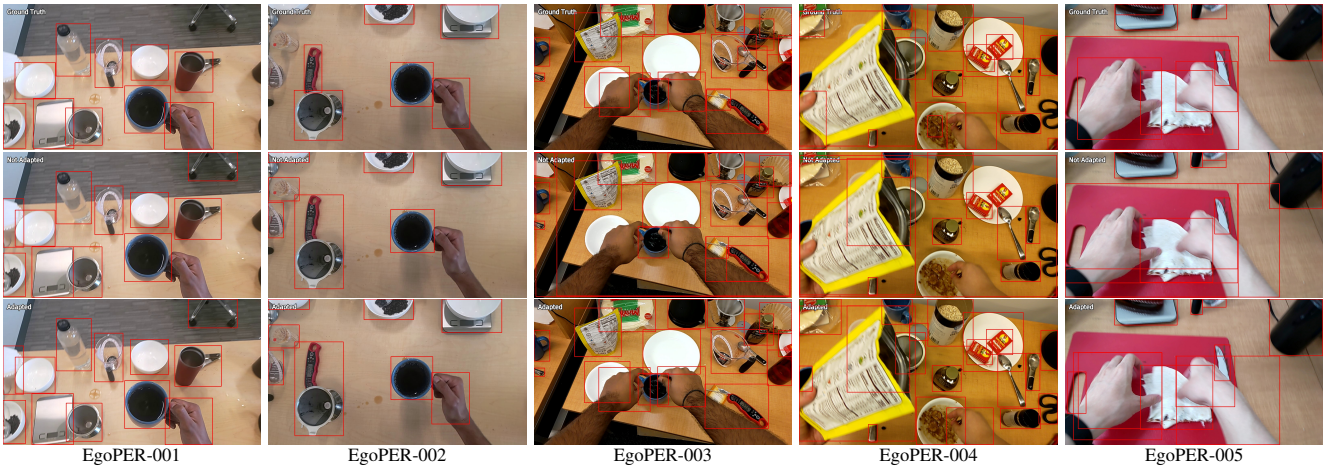
dataset [11]. For each model, we use a new classification head, which only has one object category, and finetune the whole network on the training images. Faster-RCNN and YOLOv8 use their corresponding sets of loss functions that are different from GroundingDINO [13].

The detection performance is compared in Table 2. It is clear that closed-vocabulary detectors fall behind adapted GroundingDINO by a significant margin. Faster-RCNN achieves acceptable performance on all datasets but HOIST, where the target objects still belong to certain categories. However, on HOIST dataset, where the target object is a more abstract concept of hand-held objects, all closed-vocabulary models get low AP^m. Please note that those two detection models are based on convolution operation, and both have considerably fewer parameters than GroundingDINO. Closed-vocabulary detectors have different overall architectures and training procedures than OVDs. Yet, it still shows the benefits of adapting an OVD model with a lightweight enhancer for such difficult detection tasks.

5. Results on ODinW-13 dataset

ODinW is a suite of datasets covering a wide range of domains. We report the average AP^m of our enhancer with $r = 16$ on the subset of 13 ODinW datasets [10] and compare with ResTuning [6], BitFit [2], and LoRA [5]. The initial prompt for each dataset in ODinW-13 is presented in Table 4. We construct the prompts based on the categories in each dataset or a general term for all classes in the case of Pascal VOC. The results are present in Table 3.

The results show that our enhancer outperforms all other



EgoPER-001 EgoPER-002 EgoPER-003 EgoPER-004 EgoPER-005

(c) Results on EgoPER dataset [8] using initial prompt "kitchen object .".

Figure 1. Qualitative results on several videos from different datasets. Each column of three images is from one video. The video IDs of EgoPER are renamed for better readability. Please refer to Sec. 6 for the original IDs. The images show the ground-truth bounding boxes, the detected bounding boxes from the pre-trained GroundingDINO model, and the detected bounding boxes from the model with an adaptively trained prompt enhancer. Please refer to Sec. 3 for more details. This figure is best viewed on color screens.

methods in average AP^m and achieves the best AP^m on 7 out of 13 datasets and the second-best on 4 datasets. We can also observe that our method improves the base model on all datasets, whereas other methods make the base model worse in some datasets like Rabbits, Vehicles, and Pistols. These are easy datasets on which the base model already has high AP^m; hence, bringing improvement is more challenging.

6. Original IDs of videos used in the qualitative analysis

Here, we provide the mapping from the video IDs used in the qualitative analysis to the corresponding original IDs for result reproduction purposes. The mapping is given in Table 5.

References

[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018. 2, 5

[2] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, 2022. Association for Computational Linguistics. 1, 3

[3] Simone Fabbrizzi, Symeon Papadopoulos, Eirini Ntoutsi, and Yiannis Kompatsiaris. A survey on bias in visual datasets. *ArXiv*, 2021. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[5] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 2, 3

[6] Zeyinzi Jiang, Chaojie Mao, Ziyuan Huang, Ao Ma, Yiliang Lv, Yujun Shen, Deli Zhao, and Jingren Zhou. Res-tuning: A flexible and efficient tuning paradigm via unbinding tuner from backbone. In *NeurIPS*, 2024. 1, 2, 3

[7] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. 2

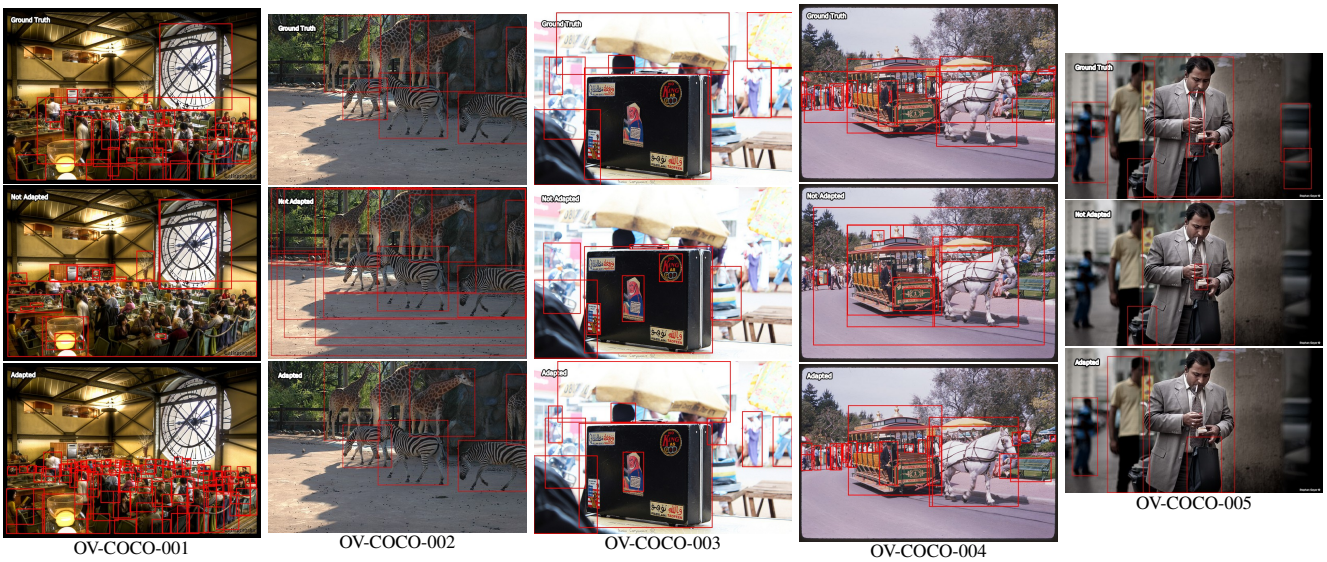
[8] Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. Error detection in egocentric procedural task videos. In *CVPR*, 2024. 2, 4

[9] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 1

[10] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 3

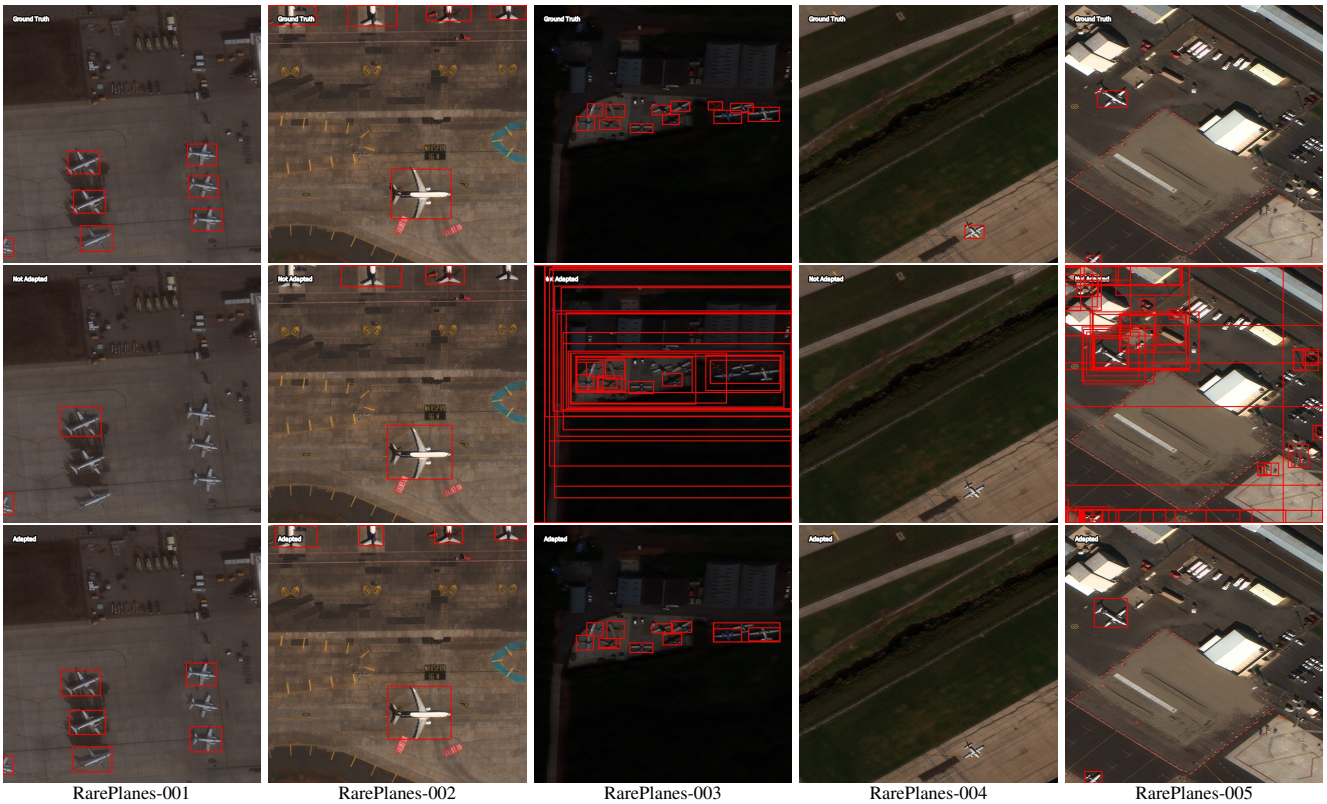
[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *ArXiv*, 2014. 2, 3

[12] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2



OV-COCO-001 OV-COCO-002 OV-COCO-003 OV-COCO-004 OV-COCO-005

(d) Results on OV-COCO dataset [1] using initial prompt "coco objects ·".



RarePlanes-001 RarePlanes-002 RarePlanes-003 RarePlanes-004 RarePlanes-005

(e) Results on RarePlanes dataset [17] using initial prompt "planes ·".

Figure 1. Qualitative results on several videos from different datasets. Each column of three images is from one video. The images show the ground-truth bounding boxes, the detected bounding boxes from the pre-trained GroundingDINO model, and the detected bounding boxes from the model with an adaptively trained prompt enhancer. Please refer to Sec. 3 for more details. This figure is best viewed on color screens.

[13] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *ArXiv*, 2023. 3

Method	Pascal VOC	Aerial Drone	Aquarium	Rabbits	EgoHands	Mushrooms	Packages	Raccoon	Shellfish	Vehicles	Pistols	Pothole	Thermal	Avg. ↑
Base model	40.01	20.21	29.74	66.55	62.24	50.18	60.53	58.95	48.45	61.84	66.58	26.52	66.85	50.67
LoRA	**62.11**	26.03	37.97	65.46	62.43	57.90	**65.45**	62.31	52.32	61.13	66.25	37.54	75.97	56.37
Bitfit	61.49	25.41	36.27	65.68	62.30	61.90	63.86	61.50	53.50	61.20	**67.62**	35.81	77.47	56.46
ResTuning	49.76	27.46	33.64	67.69	**67.53**	**69.07**	63.60	**68.02**	**55.41**	60.96	62.91	33.44	76.03	56.58
Enhancer, $r = 16$	57.63	**29.59**	**38.52**	**71.21**	65.52	60.85	**65.45**	63.24	53.64	**62.90**	66.89	**38.05**	**78.17**	**57.82**

Table 3. Detection AP^m of Grounding DINO before and after adaptation on ODinW-13. We compare the AP^m of base models with no adaptation and models adapted with different adaptation methods. The method with the highest AP^m for each dataset is marked in **bold**. The second-best is underlined. Our enhancer with $r = 16$ achieves the best average AP^m and the best AP^m on 7 out of 13 datasets.

Dataset	Initial prompt
Pascal VOC	common objects ·
Aerial Drone	docks · boats · lifts · jetskis · cars ·
Aquarium	aquatic animals ·
Rabbits	rabbits ·
EgoHands	hands ·
Mushrooms	mushrooms ·
Packages	packages ·
Raccoon	raccoon ·
Shellfish	shrimp · lobster · crab ·
Vehicles	vehicles ·
Pistols	pistols ·
Pothole	pothole ·
Thermal	dogs in thermal images · people in thermal images ·

Table 4. Initial prompt for each dataset in ODinW-13.

Video ID	Original ID
EgoPER-001	COFF_PINW006-01-0
EgoPER-002	COFF_PINW006-04-0
EgoPER-003	V_Tea_3
EgoPER-004	Y_Oat_4
EgoPER-005	Z_Quesadila_4
HOIST-001	LFja1ShZFsA
HOIST-002	6XK5af6bhRM
HOIST-003	2P58pcmym50
HOIST-004	PtOS5evrqrQ
HOIST-005	ZmzY3RILf6c

Table 5. The mapping from video IDs used in the qualitative analysis to the corresponding original IDs.

[14] Otniel-Bogdan Mercea, Alexey A. Gritsenko, Cordelia Schmid, and Anurag Arnab. Time-, memory- and parameter-efficient visual adaptation. In *CVPR*, 2024. 1, 2

[15] Supreeth Narasimhaswamy, Huy Anh Nguyen, Lihan Huang, and Minh Hoai. Hoist-former: Hand-held objects identification, segmentation, and tracking in the wild. In *CVPR*, 2024. 2, 3

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[17] Jacob Shermeyer, Thomas Hossler, Adam Van Etten, Daniel Hogan, Ryan Lewis, and Daeil Kim. Rareplanes: Synthetic data takes flight. In *IEEE Winter Conference on Applications of Computer Vision*, 2021. 2, 5

[18] Zekun Zhang and Minh Hoai. Object detection with self-supervised scene adaptation. In *CVPR*, 2023. 2, 3

[19] Kankan Zhou, Eason Lai, and Jing Jiang. VLStereoSet: A study of stereotypical bias in pre-trained vision-language models. In *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, 2022. 2