
DON'T NEED RETRAINING: A Mixture of DETR and Vision Foundation Models for Cross-Domain Few-Shot Object Detection

Anonymous Author(s)

Affiliation

Address

email

Supplemental Materials

1 Introduction

In the following sections, we will present additional experimental results of our method. Firstly, we show the impact of incorporating different foundation models in Sec. 2 and analyze the impact of foundation models with varying parameter scales in Sec. 3. Then, we investigate the impact of hyperparameter choices in Sec. 4. Finally, we examine performance under different fine-tuning strategies in Sec. 5. All experiments are conducted on six cross-domain datasets with distinct domain shifts under the 10-shot task setting. Details of the datasets are summarized in Table 1.

Table 1: Dataset details under the 10-shot setting.

Method	ArTaxOr[1]	Clipart1k[2]	DIOR[3]	DeepFish[4]	NEU-DET[5]	UODD[6]
Class number	7	20	20	1	6	2
Train data number	68	20	155	10	60	20
Test data number	1383	500	5000	909	360	506

2 Ablation Study on Different Foundation Models

To assess how various foundation models enhance backbone features, we use DINO [7] with a ResNet [8] backbone as our baseline, comparing its performance when augmented with CLIP[9], DINOv2[10], and a combination of both. As shown in Table 2, DINOv2 consistently achieves superior results on cross-domain tasks, while CLIP underperforms in comparison. We further explore integrating multiple foundation models simultaneously to boost feature representation. The integration of DINOv2 and CLIP exceeds the GPU memory limitation, forcing us to reduce the batch size to 1. Experimental results demonstrate that integrating multiple foundation models leads to a slight performance improvement, confirming the complementary nature of different foundation models. However, this integration significantly increases training time and memory consumption. Considering the trade-off between performance gains and computational costs, we ultimately select only DINOv2 as the expert model.

Table 2: Comparison results of combining different foundation models. The best results are highlighted with **bold**. *bs* denotes the batch size. *Params* denotes the parameter count of the entire model.

Method	ArTaxOr	Clipart1k	DIOR	DeepFish	NEU-DET	UODD	Training time	Params
Baseline (bs = 2)	11.4	23.2	14.4	20.5	11.8	9.9	0.8h	135M
+ CLIP (bs = 2)	53.7	47.7	35.0	28.9	24.0	19.2	1.7h	660M
+ DINOv2 (bs = 2, ours)	71.3	49.9	37.8	34.1	23.7	22.1	2.0h	660M
+ DINOv2&CLIP (bs = 1)	71.8	50.2	38.1	33.9	23.8	21.7	3.6h	980M

3 Ablation Study on Foundation Models of Different Parameter Sizes

To explore the relationship between the parameter size of foundation models and their performance enhancement, we evaluate DINOv2-small, DINOv2-base, DINOv2-large, DINOv2-giant, CLIP-base-16, CLIP-base-32, CLIP-large-14 and CLIP-large-336 as expert models for augmenting backbone features. As shown in Table 3, the experiments reveal that using CLIP as expert model to enhance the original object detection model achieves limited improvement. In contrast, introducing DINOv2 with the same parameter size leads to significantly greater performance gains compared to those achieved by CLIP. Notably, DINOv2-giant exceeds the memory capacity of our GPU, forcing us to reduce the batch size to 1 during training. The performance of DINOv2-giant is suboptimal due to its large intermediate feature maps, which require processing a significant number of additional parameters. These extra parameters not only hinder effective fine-tuning on downstream tasks but also considerably increase training time. In contrast, **DINOv2-large offers a balanced trade-off between parameter size, training time, and performance improvement, delivering the best results within a reasonable computational budget.** Consequently, we select DINOv2-large as the expert model in this study.

Table 3: Comparison results of foundation models of different parameter sizes. *Params* denotes the parameter count of different foundation models.

Method	ArTaxOr	Clipart1k	DIOR	DeepFish	NEU-DET	UODD	Training time	Params
DINOv2-S (bs = 2)	54.9	38.5	30.4	28.5	22.4	20.2	1.1h	21M
DINOv2-B (bs = 2)	67.5	45.9	35.6	30.4	21.6	21.3	1.5h	86M
DINOv2-L (bs = 2,ours)	71.3	49.9	37.8	34.1	23.7	22.1	2.0h	300M
DINOv2-g (bs = 1)	69.1	48.7	36.7	33.1	22.0	21.4	4.5h	1100M
CLIP-B-16 (bs = 2)	40.7	40.8	31.1	29.3	20.1	17.1	1.2h	86M
CLIP-B-32 (bs = 2)	30.5	39.0	27.8	27.4	18.3	16.4	1.2h	87M
CLIP-L-14 (bs = 2)	51.8	44.2	33.6	29.1	22.0	18.8	1.6h	300M
CLIP-L-336 (bs = 2)	53.7	47.7	35.0	28.9	24.0	19.2	1.7h	300M

4 Ablation Study on Hyperparameters Setting

In the shared and private expert projection modules, we introduce two hyperparameters, m and n , to control the proportion of shared and private information within the expert features. e.g. $\mathbf{F}_S^l = \mathbf{F}_D^l \cdot \boldsymbol{\theta}_s \in \mathbb{R}^{B \times \frac{n}{m} C \times H \times W}$, $\mathbf{F}_P^l = \mathbf{F}_D^l \cdot \boldsymbol{\theta}_p \in \mathbb{R}^{B \times \frac{m-n}{m} C \times H \times W}$. To determine the optimal configuration, we conduct an ablation study on the values of m and n . As shown in Table 4, the best performance is achieved when $m = 16$ and $n = 15$.

Table 4: The 10-shot ablation results on hyperparameters m and n .

m & n	ArTaxOr	Clipart1k	DIOR	DeepFish	NEU-DET	UODD
2, 1	71.9	49.6	36.6	31.1	22.8	18.2
4, 3	70.0	49.0	37.3	32.5	21.1	21.4
8, 7	70.3	48.9	35.7	32.3	22.4	21.4
16, 15 (ours)	71.3	49.9	37.8	34.1	23.7	22.1
32, 31	69.5	48.5	36.4	33.2	23.1	22.9

In the mixture of experts module, we introduce two hyperparameters, α and β , to control the degree of enhancement contributed by region-level and expert-level routing to the original backbone features. e.g. $\mathbf{F}_{\text{fused}}^l = \mathbf{F}_b^l + \sum_{n=1}^N \left(\alpha \cdot \mathbf{G}_{\text{image}}^{l,n} \circledast \mathbf{F}_D^{l',n} + \beta \cdot \mathbf{G}_{\text{token}}^{l,n} \circledast \mathbf{F}_D^{l',n} \right)$. To evaluate the impact of these parameters, we perform an ablation study on α and β . As shown in Table 5, the results indicate that the optimal configuration is $\alpha = 0.5$ and $\beta = 0.5$.

Table 5: The 10-shot ablation results on hyperparameters α and β .

α & β	ArTaxOr	Clipart1k	DIOR	DeepFish	NEU-DET	UODD
0.1, 0.9	71.1	49.1	35.3	34.9	22.9	22.8
0.3, 0.7	70.5	49.7	30.5	34.1	23.0	20.6
0.5, 0.5 (ours)	71.3	49.9	37.8	34.1	23.7	22.1
0.7, 0.3	70.1	49.2	37.6	31.9	21.7	18.1
0.9, 0.1	68.2	49.8	36.4	32.2	21.1	21.0

5 Ablation Study on Fine-Tuning Strategies

As shown in Table 6, All fine-tuning strategies keep the foundation model frozen without updating its parameters. LoRA-based fine-tuning[11] achieves the lowest number of trainable parameters but suffers from limited performance. Partially fine-tuning the model while excluding the backbone introduces a moderate increase in trainable parameters and yields improved results. Full fine-tuning achieves the highest accuracy, yet comes at the cost of significant training overhead and a heightened risk of overfitting in few-shot scenarios. In contrast, our proposed strategy—which fine-tunes only the classification head, regression head, and the proposed module—strikes an effective balance between computational efficiency and detection performance, demonstrating strong generalization in cross-domain few-shot object detection tasks.

Table 6: The 10-shot ablation results on different finetuning strategies. *Full Finetune* denotes fine-tuning all model parameters. *LoRA* denotes fine-tuning all model based on Low-Rank Adaptation. *Partial Finetune* denotes fine-tuning all components except the backbone. *Ours* denotes fine-tuning only the classification head, regression head, and the proposed module. *Trainable Params* indicates the number of trainable parameters in the model.

Fine-Tuning Strategy	ArTaxOr	Clipart1k	DIOR	DeepFish	NEU-DET	UODD	Trainable Params
Full Finetune	55.2	39.0	32.6	22.3	20.8	19.3	+ 57.4M
LoRA	44.5	28.0	26.7	20.9	18.4	15.9	+ 3.4M
Partial Finetune	50.7	34.7	30.9	22.6	20.9	16.2	+ 28.1M
Ours	71.3	49.9	37.8	34.1	23.7	22.1	+ 10.0M

References

- [1] Asger Svenning, Guillaume Mougeot, Jamie Alison, Daphne Chevalier, Nisa Luise Chavez Molina, Song-Quan Ong, Kim Bjerre, Juli Carrillo, Toke Thomas Hoeye, and Quentin Geissmann. A general method for detection and segmentation of terrestrial arthropods in images. *bioRxiv*, 2025.
- [2] Javier S Turek and Alexander G Huth. Efficient, sparse representation of manifold distance matrices for classical scaling. In *CVPR*, 2018.
- [3] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS*, 2020.
- [4] Nahuel Garcia-d’Urso, Alejandro Galan-Cuenca, Paula Pérez-Sánchez, Pau Climent-Pérez, Andres Fuster-Guillo, Jorge Azorin-Lopez, Marcelo Saval-Calvo, Juan Eduardo Guillén-Nieto, and Gabriel Soler-Capdepón. The deepfish computer vision dataset for fish instance segmentation, classification, and size estimation. *Scientific Data*, 2022.
- [5] Aditya M Deshpande, Ali A Minai, and Manish Kumar. One-shot recognition of manufacturing defects in steel surfaces. *Procedia Manufacturing*, 2020.
- [6] Lihao Jiang, Yi Wang, Qi Jia, Shengwei Xu, Yu Liu, Xin Fan, Haojie Li, Risheng Liu, Xinwei Xue, and Ruili Wang. Underwater species detection using channel sharpening attention. In *ACM MM*, 2021.
- [7] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [10] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.
- [11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.