Supplementary Material

In this supplementary material, we provide additional details which we could not include in the main paper due to space limitations, including more experimental analysis and visualization details that help us develop further insights to the proposed approach. We discuss:

- More results of generalized few-shot object detection.
- Additional analysis on Prototypical Calibration Block.
- Related extensions of Gradient Decoupled Layer.
- Qualitative visualization results of our approach.

A. Generalized Few-Shot Object Detection

A.1. Implementation Details

As mentioned in the main paper, we take two popular evaluation protocols into consideration to assess the effectiveness of our approach, including few-shot object detection (***FSOD***) and generalized few-shot object detection (***G-FSOD***). The difference between these two protocols is whether the performance of base classes is still required after the fine-tuning stage. Following the *G-FSOD* setting in TFA [8], we fine-tune our DeFRCN on a small balanced training set consisting of both base and novel classes, where each class has the same number of annotated objects (*i.e.*, K-shot). In addition to deploying more training iterations ($2\times$), other experimental settings for *G-FSOD* are exactly consistent with the *FSOD* in our paper.

A.2. Experimental Results of *G-FSOD* Setting

In this section, we show the full benchmark results of the *G-FSOD* setting. For each evaluation metric, we report the average results of n random splits ($n = 30$ for VOC and $n = 10$ for COCO) with the same data split in TFA as well as the 95% confidence interval estimate of the mean values.

PASCAL VOC. We present the complete *G-FSOD* results of VOC ($K = 1, 2, 3, 5, 10$) in Table 1 and then analyze our results from the following three aspects: (1) Novel AP. The novel AP of our model is usually over 7% points higher than that of TFA in three data splits, which indicates that the proposed DeFRCN has absolute advantage on novel performance. (2) Base AP. Our approach is able to outperform TFA on split 2 (+1.9% $\sim$ +3.7% AP), however, it is slightly worse on data split 1 and 3 (-0.3% $\sim$ -1.0% AP). We notice that the base performance advantage of TFA comes from the strategy of fine-tuning only the last layer of detectors, which can indeed be eccentric to ensure that the base performance does not decrease too much, but it also results in the novel performance cannot be further improved. (3) Overall AP. As shown in the Table 1, the proposed DeFRCN achieves the best overall performance across all settings (+1.4% $\sim$ +4.0% AP), including data splits and shots.

COCO. The Table 2 shows the *G-FSOD* results on COCO dataset over $K = 1, 2, 3, 5, 10, 30$ shots. Although COCO is much more complicated than VOC, similar observations can be drawn about accuracy on both base classes and novel classes. Concretely, the performance on base classes is comparable to TFA, but we are far superior to TFA in terms of both novel and overall results. In addition, we further notice that as the number of support shots increases, our approach can bring more performance improvements.

B. Additional Analysis on PCB

B.1. Boost Other Approaches with PCB

As a plug-and-play module, the proposed PCB is easily equipped to any other architectures to build stronger few-shot detectors. Here, we verify this argument with introducing PCB into other previous approaches, including FRCN-ft [11], TFA [8], MPSR [9], and all experimental results on COCO dataset are shown in the Table 3. Regardless of methods or the number of shots, we observe that using PCB can consistently achieve much higher performance (+1.0% $\sim$ +3.0% points) on novel classes, which demonstrates the effectiveness and flexibility of our PCB module.

B.2. Employ Other Pre-trained Models

In the main paper, we utilize the standard ImageNet pre-trained model (IN-1K), which is widely adopted in most of few-shot object detection frameworks, to initialize both Faster-RCNN and PCB. Since the core module of PCB is the generalizable feature extractor, which determines the final performance of the score calibration, we further explore other pre-trained models (see Table 4) in this section. SwAV [1] is an efficient method for pre-training without using annotations, *i.e.*, self-supervised learning. IN-SwAV indicates that the model is pre-trained by SwAV on ImageNet. IG-WSL [5] employs the ResNeXt [10] architecture and pre-trains on a much larger social media image dataset (Instagram) with weakly-supervised learning paradigm. Table 5 shows the performance on VOC with utilizing the above three pre-trained models. No matter which one is exploited, the final performance is better with PCB. Moreover, we further notice that using a stronger pre-trained model, the performance of FSOD can be improved more.

Method	Backbone	Paradigm	# Images	# Classes
IN-SwAV [1]	ResNet-50	S-S-L	1.28M	0
IN-1K [3]	ResNet-101	S-L	1.28M	1000
IG-WSL [5]	ResNeXt-101	W-S-L	940M	1000

Table 4: The comparison between different pre-trained classification models. S-S-L, S-L and W-S-L stand for self-supervised learning, supervised learning and weakly-supervised learning respectively.

B.3. Why PCB Works ?

The PCB can be reinterpreted as a non-parameter few-shot classification model, which draws on the idea of Prototypical Network [7]. Based on the COCO 10-shot task, we calculate the channel-wise cosine similarity between different few-shot RoI prototypes ($C \times 1 \times 1$) and the feature map ($C \times H \times W$) of the test image, and then visualize the similarity map in Fig.1. We find that the prototypes from different categories can indeed activate distinct areas of the feature map, which indicates that the metric-based pairwise score in data-scarce scenario is very effective. In addition, we notice that even if the category label of novel prototype is not seen before by the pre-trained classification model, an ideal similarity map can still be obtained, *e.g.*, the novel label 'Person' does not exist in ImageNet 1K sysnets, see the first three lines in Fig.1. Moreover, the results of IN-SwAV (*i.e.* self-supervised paradigm) in Table 5 further prove this argument. According to the visualization and above analysis, we believe that it is reasonable for PCB to utilize the pairwise score based on classification model to calibrate the softmax score from the original classification branch of few-shot detector.

C. Related extensions of GDL

C.1. Conventional Cross-Domain Object Detection

In the experimental section of the main paper, we have verified that the proposed GDL is not only remarkably effective for few-shot object detection (*i.e.*, *FSOD*, *G-FSOD* and cross-domain *FSOD*), but also plays a positive role in conventional object detection. In this section, we further explore the conventional cross-domain object detection and all experimental results are shown in Table 6. We use the Cityscapes [2] and FoggyCityscapes [6] (Normal-to-Foggy) as our benchmarks and follow the same evaluation protocol in [12]. By comparing the experimental results of the second row and the third row in Table 6, we find that adding GDL achieves 32.8% mAP on the weather transfer task, which is +2.8% higher than the plain Faster-RCNN.

C.2. The value range of λ

We discuss the value range of λ into three situations.

- $\lambda_{rpn} \in [0, 1]$ *and* $\lambda_{rcnn} \in [0, 1]$. This setting has been explored in our paper and achieved the best results.
- $\lambda_{rpn} \in (-\infty, 0)$ *or* $\lambda_{rcnn} \in (-\infty, 0)$. $\lambda < 0$ means that the downstream module has a negative effect on the optimization direction of backbone. Without any adversarial strategy, this setup is meaningless for object detection.
- $\lambda_{rpn} \in (1, +\infty)$ *or* $\lambda_{rcnn} \in (1, +\infty)$. $\lambda > 1$ means that the gradient from the downstream module magnifies its effect on the backbone. We notice that slightly increasing λ (*e.g.* $1 \sim 5$) will not affect the stability of detector but incite performance degradation, which is caused by the backbone's update speed faster than before and overfitting. When λ is relatively large (*e.g.* > 5), due to overemphasizing the degree of coupling between the module and the backbone, the model will usually converge to an unreasonable saddle point and cause a collapse solution. The value of 5 is obtained by experiments approximately.

D. More Visualization of Our Approach

We provide qualitative visualizations of the detected novel objects on COCO dataset in Fig.2. We show both success (green box) and failure cases (red box) when detecting novel objects for each image to help analyze the possible error types, including misclassifying novel objects, mislocalizing objects and missing detections.

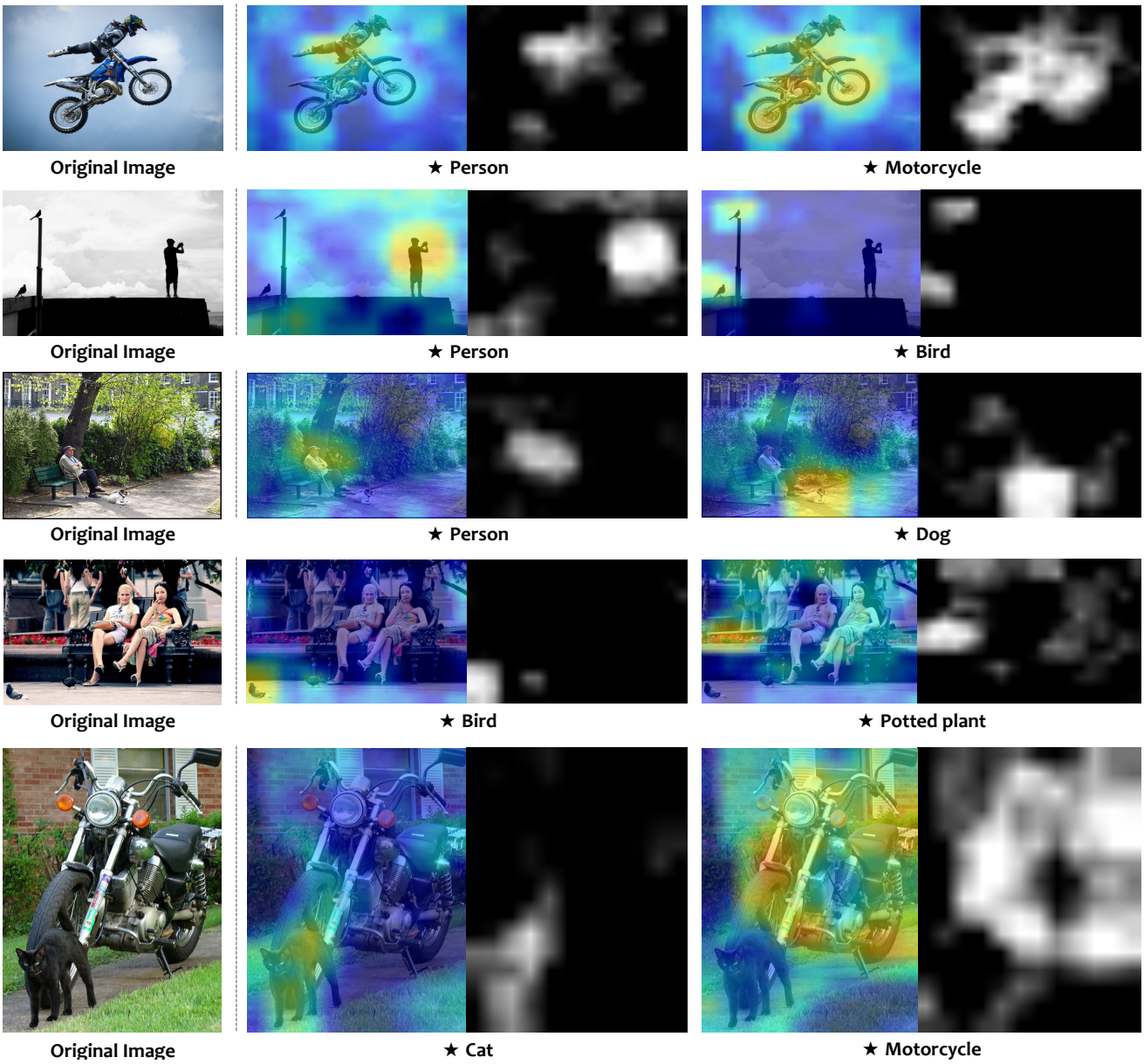

Figure 1: The visualization of PCB on COCO val set. Through different kinds of prototypes, which are calculated by K-shot samples ($K = 10$), distinct areas of the same picture are activated. The symbol ★ indicates that it is some kind of prototypes.

Split	# shots	Method	Overall *#20*			Base *#15*	Novel *#5*
			AP	AP50	AP75	AP	AP
Split 1	1	FSRW [4]	27.6±0.5	50.8 ±0.9	26.5±0.6	34.1±0.5	8.0±1.0
		FRCN+ft [11]	30.2±0.6	49.4±0.7	32.2±0.9	38.2±0.8	6.0±0.7
		TFA [8]	40.6±0.5	64.5±0.6	44.7±0.6	**49.4±0.4**	14.2±1.4
		DeFRCN	**42.0±0.6** (+1.4)	**66.7±0.8** (+2.2)	**45.5±0.7** (+0.8)	48.4±0.4 (-1.0)	**22.5±1.7** (+8.3)
	2	FSRW [4]	28.7±0.4	52.2±0.6	27.7±0.5	33.9±0.4	13.2±1.0
		FRCN+ft [11]	30.5±0.6	49.4±0.8	32.6±0.7	37.3±0.7	9.9±0.9
		TFA [8]	42.6±0.3	67.1±0.4	47.0±0.4	**49.6±0.3**	21.7±1.0
		DeFRCN	**44.3±0.4** (+1.7)	**70.2±0.5** (+3.1)	**48.0±0.6** (+1.0)	49.1±0.3 (-0.5)	**30.6±1.2** (+8.9)
	3	FSRW [4]	29.5±0.3	53.3±0.6	28.6±0.4	33.8±0.3	16.8±0.9
		FRCN+ft [11]	31.8±0.5	51.4±0.8	34.2±0.6	37.9±0.5	13.7±1.0
		TFA [8]	43.7±0.3	68.5±0.4	48.3±0.4	**49.8±0.3**	25.4±0.9
		DeFRCN	**45.3±0.3** (+1.6)	**71.5±0.4** (+3.0)	**49.0±0.5** (+0.7)	49.3±0.3 (-0.5)	**33.7±0.8** (+8.3)
	5	FSRW [4]	30.4±0.3	54.6±0.5	29.6±0.4	33.7±0.3	20.6±0.8
		FRCN+ft [11]	32.7±0.5	52.5±0.8	35.0±0.6	37.6±0.4	17.9±1.1
		TFA [8]	44.8±0.3	70.1±0.4	49.4±0.4	**50.1±0.2**	28.9±0.8
		DeFRCN	**46.4±0.3** (+1.6)	**73.1±0.3** (+3.0)	**50.4±0.4** (+1.0)	49.6±0.3 (-0.5)	**37.3±0.8** (+8.4)
	10	FRCN+ft [8]	33.3±0.4	53.8±0.6	35.5±0.4	36.8±0.4	22.7±0.9
		TFA [8]	45.8±0.2	71.3±0.3	50.4±0.3	**50.4±0.2**	32.0±0.6
		DeFRCN	**47.2±0.2** (+1.4)	**74.0±0.3** (+2.7)	**51.3±0.3** (+0.9)	49.9±0.2 (-0.5)	**39.8±0.7** (+7.8)
Split 2	1	FSRW [4]	28.4±0.5	51.7±0.9	27.3±0.6	35.7±0.5	6.3±0.9
		FRCN+ft [11]	30.3±0.5	49.7±0.5	32.3±0.7	38.8±0.6	5.0±0.6
		TFA [8]	36.7±0.6	59.9±0.8	39.3±0.8	45.9±0.7	9.0±1.2
		DeFRCN	**40.7±0.5** (+4.0)	**64.8±0.7** (+4.9)	**43.8±0.6** (+4.5)	**49.6±0.4** (+3.7)	**14.6±1.5** (+5.6)
	2	FSRW [4]	29.4±0.3	53.1±0.6	28.5±0.4	35.8±0.4	9.9±0.7
		FRCN+ft [11]	30.7±0.5	49.7±0.7	32.9±0.6	38.4±0.5	7.7±0.8
		TFA [8]	39.0±0.4	63.0±0.5	42.1±0.6	47.3±0.4	14.1±0.9
		DeFRCN	**42.7±0.3** (+3.7)	**67.7±0.5** (+4.7)	**45.7±0.5** (+3.6)	**50.3±0.2** (+3.0)	**20.5±1.0** (+6.4)
	3	FSRW [4]	29.9±0.3	53.9±0.4	29.0±0.4	35.7±0.3	12.5±0.7
		FRCN+ft [11]	31.1±0.3	50.1±0.5	33.2±0.5	38.1±0.4	9.8±0.9
		TFA [8]	40.1±0.3	64.5±0.5	43.3±0.4	48.1±0.3	16.0±0.8
		DeFRCN	**43.5±0.3** (+3.4)	**68.9±0.4** (+4.4)	**46.6±0.4** (+3.3)	**50.6±0.3** (+2.5)	**22.9±1.0** (+6.9)
	5	FSRW [4]	30.4±0.4	54.6±0.5	29.5±0.5	35.3±0.3	15.7±0.8
		FRCN+ft [11]	31.5±0.3	50.8±0.7	33.6±0.4	37.9±0.4	12.4±0.9
		TFA [8]	40.9±0.4	65.7±0.5	44.1±0.5	48.6±0.4	17.8±0.8
		DeFRCN	**44.6±0.3** (+3.7)	**70.2±0.5** (+4.5)	**47.8±0.4** (+3.7)	**51.0±0.2** (+2.4)	**25.8±0.9** (+8.0)
	10	FRCN+ft [8]	32.2±0.3	52.3±0.4	34.1±0.4	37.2±0.3	17.0±0.8
		TFA [8]	42.3±0.3	67.6±0.4	45.7±0.3	49.4±0.2	20.8±0.6
		DeFRCN	**45.6±0.2** (+3.3)	**71.5±0.3** (+3.9)	**49.0±0.3** (+3.3)	**51.3±0.2** (+1.9)	**29.3±0.7** (+8.5)
Split 3	1	FSRW [4]	27.5±0.6	50.0±1.0	26.8±0.7	34.5±0.7	6.7±1.0
		FRCN+ft [11]	30.8±0.6	49.8±0.8	32.9±0.8	39.6±0.8	4.5±0.7
		TFA [8]	40.1±0.3	63.5±0.6	43.6±0.5	**50.2±0.4**	9.6±1.1
		DeFRCN	**41.6±0.5** (+1.5)	**66.0±0.9** (+2.5)	44.9±0.6 (+1.3)	49.4±0.4(-0.8)	**17.9±1.6** (+8.3)
	2	FSRW [4]	28.7±0.4	51.8±0.7	28.1±0.5	34.5±0.4	11.3±0.7
		FRCN+ft [11]	31.3±0.5	50.2±0.9	33.5±0.6	39.1±0.5	8.0±0.8
		TFA [8]	41.8±0.4	65.6±0.6	45.3±0.4	**50.7±0.3**	15.1±1.3
		DeFRCN	**44.0±0.4** (+2.2)	**69.5±0.7** (+3.9)	**47.7±0.5** (+2.4)	50.2±0.2 (-0.5)	**26.0±1.3** (+10.9)
	3	FSRW [4]	29.2±0.4	52.7±0.6	28.5±0.4	34.2±0.3	14.2±0.7
		FRCN+ft [11]	32.1±0.5	51.3±0.8	34.3±0.6	39.1±0.5	11.1±0.9
		TFA [8]	43.1±0.4	67.5±0.5	46.7±0.5	**51.1±0.3**	18.9±1.1
		DeFRCN	**45.1±0.3** (+2.0)	**70.9±0.5** (+3.4)	**48.8±0.4** (+2.1)	50.5±0.2 (-0.6)	**29.2±1.0** (+10.3)
	5	FSRW [4]	30.1±0.3	53.8±0.5	29.3±0.4	34.1±0.3	18.0±0.7
		FRCN+ft [11]	32.4±0.5	51.7±0.8	34.4±0.6	38.5±0.5	14.0±0.9
		TFA [8]	44.1±0.3	69.1±0.4	47.8±0.4	**51.3±0.2**	22.8±0.9
		DeFRCN	**46.2±0.3** (+2.1)	**72.4±0.4** (+3.3)	**50.0±0.5** (+2.2)	51.0±0.2 (-0.3)	**32.3±0.9** (+9.5)
	10	FRCN+ft [11]	33.1±0.5	53.1±0.7	35.2±0.5	38.0±0.5	18.4±0.8
		TFA [8]	45.0±0.3	70.3±0.4	48.9±0.4	**51.6±0.2**	25.4±0.7
		DeFRCN	**47.0±0.3** (+2.0)	**73.3±0.3** (+3.0)	**51.0±0.4** (+2.1)	51.3±0.2 (-0.3)	**34.7±0.7** (+9.3)

Table 1: Generalized few-shot object detection (*G-FSOD*) performance on PASCAL VOC dataset. For each metric, we report the average and 95% confidence interval computed over 30 random samples. All comparison results refer from [8].

# shots	Method	Overall *#80*			Base *#60*	Novel *#20*
		AP	AP50	AP75	AP	AP
1	FRCN+ft [11]	16.2±0.9	25.8±1.2	17.6±1.0	21.0±1.2	1.7±0.2
	TFA [8]	**24.4±0.6**	**39.8±0.8**	26.1±0.8	**31.9±0.7**	1.9±0.4
	DeFRCN (Ours)	24.0±0.4 (-0.4)	36.9±0.6 (-2.9)	**26.2±0.4** (+0.1)	30.4±0.4 (-1.5)	**4.8±0.6** (+2.9)
2	FRCN+ft [11]	15.8±0.7	25.0±1.1	17.3±0.7	20.0±0.9	3.1±0.3
	TFA [8]	24.9±0.6	**40.1±0.9**	27.0±0.7	**31.9±0.7**	3.9±0.4
	DeFRCN (Ours)	**25.7±0.5** (+0.8)	39.6±0.8 (-0.5)	**28.0±0.5** (+1.0)	31.4±0.4 (-0.5)	**8.5±0.8** (+4.6)
3	FRCN+ft [11]	15.0±0.7	23.9±1.2	16.4±0.7	18.8±0.9	3.7±0.4
	TFA [8]	25.3±0.6	40.4±1.0	27.6±0.7	32.0±0.7	5.1±0.6
	DeFRCN (Ours)	**26.6±0.4** (+1.3)	**41.1±0.7** (+0.7)	**28.9±0.4** (+1.3)	**32.1±0.3** (+0.1)	**10.7±0.8** (+5.6)
5	FRCN+ft [11]	14.4±0.8	23.0±1.3	15.6±0.8	17.6±0.9	4.6±0.5
	TFA [8]	25.9±0.6	41.2±0.9	28.4±0.6	32.3±0.6	7.0±0.7
	DeFRCN (Ours)	**27.8±0.3** (+1.9)	**43.0±0.6** (+1.8)	**30.2±0.3** (+1.8)	**32.6±0.3** (+0.3)	**13.6±0.7** (+6.6)
10	FRCN+ft [11]	13.4±1.0	21.8±1.7	14.5±0.9	16.1±1.0	5.5±0.9
	TFA [8]	26.6±0.5	42.2±0.8	29.0±0.6	32.4±0.6	9.1±0.5
	DeFRCN (Ours)	**29.7±0.2** (+3.1)	**46.0±0.5** (+3.8)	**32.1±0.2** (+3.1)	**34.0±0.2** (+1.6)	**16.8±0.6** (+7.7)
30	FRCN+ft [11]	13.5±1.0	21.8±1.9	14.5±1.0	15.6±1.0	7.4±1.1
	TFA [8]	28.7±0.4	44.7±0.7	31.5±0.4	34.2±0.4	12.1±0.4
	DeFRCN (Ours)	**31.4±0.1** (+2.7)	**48.8±0.2** (+4.1)	**33.9±0.1** (+2.4)	**34.8±0.1** (+0.6)	**21.2±0.4** (+9.1)

Table 2: Generalized few-shot object detection (*G-FSOD*) performance on COCO dataset. For each metric, we report the average and 95% confidence interval computed over 10 random samples. All comparison results refer from [8].

Method	w / PCB	# shots					
		1	2	3	5	10	30
FRCN-ft [11]	✗	1.0	1.8	2.8	4.0	6.9	11.0
	✓	2.4 (+1.4)	4.1 (+2.3)	5.2 (+2.4)	6.6 (+2.6)	9.9 (+3.0)	14.0 (+3.0)
TFA [8]	✗	4.4	5.4	6.0	7.7	9.0	13.4
	✓	6.7 (+2.3)	7.6 (+2.2)	9.0 (+3.0)	10.4 (+2.7)	11.8 (+2.8)	15.5 (+2.1)
MPSR [9]	✗	5.1	6.7	7.4	8.7	9.8	14.5
	✓	6.7 (+1.6)	8.9 (+2.2)	9.7 (+2.3)	10.9 (+2.2)	11.9 (+2.1)	15.5 (+1.0)
DeFRCN (Ours)	✗	7.9	10.9	13.4	14.6	16.9	21.0
	✓	9.3 (+1.4)	12.9 (+2.0)	14.8 (+1.4)	16.1 (+1.5)	18.5 (+1.6)	22.6 (+1.6)

Table 3: Effectiveness of Prototypical Calibration Block with different approaches. We evaluate *FSOD* performance (mAP) on COCO dataset with $K = 1, 2, 3, 5, 10, 30$ shots over multiple runs. All experimental results are reproduced by us. The term w/PCB indicates whether the method uses the PCB module. Note that the α in PCB is set to 0.5 in all experiments.

Method	Model	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
DeFRCN	✗	47.0	48.8	52.3	57.1	55.6	22.5	31.9	42.1	45.6	42.3	42.5	48.7	48.9	51.1	52.2
	IN-SwAV [1]	48.7	52.4	54.5	60.2	56.3	26.9	34.6	44.6	48.1	44.7	41.8	50.1	50.5	53.4	55.1
	IN-1K [3]	53.6	57.5	61.5	64.1	60.8	30.1	38.1	47.0	53.3	47.9	48.4	50.9	52.3	54.9	57.4
	IG-WSL [5]	62.3	64.5	66.6	69.3	68.2	37.5	44.7	53.5	57.6	54.7	54.7	57.2	59.0	60.9	62.0

Table 5: Experimental results of employing different pre-trained model in PCB on PASCAL VOC dataset. All reported results are averaged over 30 random samples. IN-SwAV, IN-1K and INS-WSL denote the different pre-trained models from ImageNet self-supervised learning, conventional supervised learning and weakly-supervised learning separately.

Cityscapes → FoggyCityscapes										
Method	Backbone	Bus	Bicycle	Car	Motor	Person	Rider	Train	Truck	mAP
Faster-RCNN [12]	VGG16	25.0	26.8	30.6	15.5	24.1	29.4	4.6	10.6	20.8
Faster-RCNN *	ResNet-101	31.5	**39.3**	45.2	24.7	**35.3**	41.2	8.8	18.7	30.0
+ GDL	ResNet-101	**32.9** (+1.4)	38.4 (-0.9)	**47.3** (+2.1)	**26.6** (+1.9)	34.3 (-1.0)	**41.4** (+0.2)	**17.3** (+8.5)	**24.3** (+7.6)	**32.8** (+2.8)

Table 6: The performance of conventional cross-domain object detection. All results in the first line refer from [12] for brief comparison. Note that the Faster R-CNN model trained on the source domain only without any other information (denoted as "Source Only" in other papers). The symbol * indicates the model is re-implemented by us.

Figure 2: The visualization results of our 10-shot object detection on COCO dataset. We visualize the bounding boxes with score larger than 0.7. The green and red box shows the success and failure cases of our DeFRCN respectively.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020. 1, 5

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 5

[4] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8420–8429, 2019. 4

[5] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision*, 2018. 1, 5

[6] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 2

[7] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2

[8] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, 2020. 1, 4, 5

[9] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Proceedings of the European Conference on Computer Vision*, pages 456–472. Springer, 2020. 1, 5

[10] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 1

[11] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9577–9586, 2019. 1, 4, 5

[12] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13766–13775, 2020. 2, 6