Supplementary Material of "DiGeo"

1. Approach

1.1. Simplex ETF & Neural Collapse

The neural collapse (NC) phenomenon is revealed by [11] in the fully-supervised learning, *i.e.*, an extremely simple mathematical structure on the last-layer features. In particular, when the model is well-trained on a balanced dataset, the N_f features $\{\mathbf{x}_{c,i}\}_{i=1}^{N_f}$ for class c will converge to its class mean $\bar{\mathbf{x}}_c = \frac{1}{N_f}\sum_i \mathbf{x}_{c,i}$ where the class means $\{\bar{\mathbf{x}}_c\}_{c\in\mathcal{C}}$ together the class centers $\{\mathbf{w}_c\}_{c\in\mathcal{C}}$ will collapse to the simplex equiangular tight frame (Simplex ETF). Meanwhile, though the optimization objective of class mean and class centers (classifier weights) are not exactly the same, the class mean and class centers will still converge to each other.

Simplex Equiangular Tight Frame denotes a collection of vectors $\mathbf{W}^* = \{\mathbf{w}'_i\}_{i=1}^{N_c} \in \mathcal{R}^{d\times N_c}$ that

$$\mathbf{W}^* = \sqrt{\frac{N_C}{N_C-1}}\mathbf{U}\big(\mathbf{I}_{N_C} - \frac{1}{N_c}\mathbf{1}_{N_C}\mathbf{1}_{N_C}^T\big) \tag{1}$$

where each vector $\mathbf{w}'_i \in \mathcal{R}^d$ and $\|\mathbf{w}'_i\|_2 = 1$ for $1 \le i \le N_C$, $\mathbf{I}_{N_C} \in \mathcal{R}^{N_C\times N_C}$ and $\mathbf{1}_{N_C} \in \mathcal{R}^{N_C}$ denote the identity matrix and all-ones vector respectively. The rotation matrix $\mathbf{U} \in \mathcal{R}^{d\times N_C}$ satisfies $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{N_C}$ and $d \ge N_C - 1$. In this way, for all vectors in a simplex ETF, their pair-wise angles are identical, *i.e.*,

$$\mathbf{w}_i'^T\mathbf{w}'_j = -\frac{1}{N_C-1}, \forall i,j \in [1, N_C] \text{ and } i \ne j, \tag{2}$$

where the angle $arccos(-\frac{1}{N_C-1})$ shown in [11] ans is the maximal equiangular angle of N_C vectors in the feature space.

Algorithm 1 Iterative algorithm for obtaining Simplex ETF)

Input: Number of classes N_C, feat. dim d where $d \le N_C - 1$ maximum iterations T, stop threshold δ, learning rate τ.

1: Initialization:
  Randomly initialize W = np.random.normal(size=(N_C, d)),
2: for $t = 1$ to T
  l_2-normalize the vector $\mathbf{w}$ in each row W = normalize(W),
  calculate pair-wise l_2 distance $W_{inner} \in \mathcal{R}^{N_C\times N_C}$ and set diagonal value as max infinity
  calculate minimum distance for each class (except for the diagonal value) $idx = np.argmin(W_{inner}, axis = 1)$
  calculate objective funcdtion $obj = np.sum(W_{inner}[range(N_C), idx])$
  calculate gradient $grad = (W - W[nn_index, :]) * 2$
  update the weight $W' = W + grad * \tau$
  l_2-normalize the vector $\mathbf{w}'$ in each row W' = normalize(W'),
  update the weight $W' = W$,
  if $obj < \tau$:
    Early stop, set T as t
  end for

Output: $\mathcal{X}^{(T+1)}$

Note that the equation in Eq. 1 is a *closed-form* for obtaining an ETF but it is only used when $d \le N_C$. When $d = N_C + 1$, we can then use an *iterative algorithm* to obtain the ETF. Specifically, we randomly initialize the values in W and use the Eq.1 in the main paper to update the weight values. We provide the python-stype pseudo-code below. Note, since we want to maximize the objective function, we apply $+$ in the weight updating part.

2. Experiment

2.1. Implementation Details

The Faster-RCNN system we are using consists of a ResNet-101 feature backbone, a RPN network, and a detection module. The detection module is used to extract features for each region proposal, a linear classifier and a regression for localization. As mentioned in the main paper, since penultimate layer in the classification module is followed by a ReLU activation [3], the proposal features are constrained to have non-negative entries and its distance to weights in W^* are lower-bounded, and we thus add a linear layer (projector) on top of the extractor of proposal feature. Meanwhile, as highlighted in Sec. 5.3 in the main paper, we do not need to pretrained the detector on the base set, but directly training everything from scratch, however, we will still use the ImageNet-pretrained model to initialize the feature extractor.

The dimension of proposal feature in Faster-RCNN is $d = 1024$ by default. As such, for experiments on MSCOCO and Pascao VOC, we set the projector with the same input and output dimension. However, for experiments on LVIS, since it has 1230 classes in v0.5 and 1203 classes in v1.0, we set the output dimension of projector as 1280.

During distillation, as we mainly focus on the learning of detector. As such, we fix the ResNet-101 feature backbone and the RPN network, and only distill the detection module. Also, during distillation, we do not apply any distillation strategy on the layer for localization. Then, we first use the fixed margin $-\log(p_c)$ in the loss and train the whole network. Then, during distillation, to fasten the training process, we can choose to also initialize the detector module with the pretrained teacher model.

RFS implementation details. We directly call the "RepeatFactorTrainingSample" as the training sampler function and send rfs parameter (0.01 for VOC & COOC and 0.001 for LVIS) to the variable "SAMPLER TRAIN"

2.2. Full experiment on Pascao VOC

We summarize the performance of novel detection in Table M1. Comparing with baseline TFA, over all 15 experiments on PASCAL VOC, the Prior baseline has already outperformed TFA by 3.6 gain in nAP_{50} and 1.9 gain in bAP_{50} on average. By performing the self-distillation to adjust margins for all classes $\mathcal{C}$ adaptively, our full approach DiGeo can further improve the detection score, *e.g.*, comparable nAP_{50} with MPSR [19] but maintaining high *base* detection precision (81.3 Vs. 68.1). As reported in Table 2 in the main paper, comparing with Retentive RCNN [1], a state-of-the-art (SOTA) approach in GFSOD, besides maintaining precise base detection, our approach also improves the novel detection score (43.9 Vs. 41.1). Meanwhile, the superior performance by Retentive RCNN on split 1 when K is $\{1, 2\}$ cannot be generalized to other splits. However, our approach achieves stable and consistent gain. Meanwhile, when more training data are provided, *i.e.*, $K \geq 3$, the advantage of our DiGeo is better explored and achieve 3.76 nAP_{50} gain on average.

Table M1. Performance comparison of nAP_{50} on the PASCAL VOC dataset.

Approach	split 1					split 2					split 3					Avg.
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10	
FRCN-ft-full [21]*	15.2	20.3	29	25.5	28.7	13.4	20.6	28.6	32.4	38.8	19.6	20.8	28.7	42.2	42.1	27.1
TFA w/ fc [18]	36.8	29.1	43.6	55.7	57	18.2	29	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2	38.7
TFA w/ cos [18]	39.8	36.1	44.7	55.7	56	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8	39.9
MPSR [19]	**42.8**	43.6	48.4	55.3	61.2	**29.8**	28.1	41.6	**43.2**	47.0	**35.9**	40.0	43.7	48.9	51.3	**44.0**
Meta RCNN [21]*	16.8	20.1	20.3	38.2	43.7	7.7	12.0	14.9	21.9	31.1	9.2	13.9	26.2	29.2	36.2	22.8
FSRW [12]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	39.2	19.2	21.7	25.7	40.6	41.3	27.3
FsDetView [20]*	25.4	20.4	37.4	36.1	42.3	22.9	21.7	22.6	25.6	29.2	32.4	19.0	29.8	33.2	39.8	29.2
Retentive R-CNN [1]	42.4	**45.8**	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1	41.1
Prior	35.8	37.9	45.7	56.4	61.0	22.7	28.4	39.8	41.4	48.8	30.8	36.4	45.4	52.4	53.8	42.4
DiGeo	37.9	39.4	**48.5**	**58.6**	**61.5**	26.6	**28.9**	**41.9**	42.1	**49.1**	30.4	**40.1**	**46.9**	**52.7**	**54.7**	**44.0**

*: results reported by Retentive R-CNN [1] and TFA [18].†: Model ensembling. Full tables can be found in Supp.

2.3. Long-Tail Object Detection

LVIS [4] is derived from COCO17 [9] and has two versions of annotations. The version v1.0 contains ∼1.3M training instances of 1203 classes while the version v0.5 has ∼0.7M training instances of 1230 classes. The one reported in the main paper is of v0.5. According to the number of training instances, the classes are divided into three groups, rare (1-10), common

Table 2. Detailed Performance of MS COCO dataset.

Approach	10-shot								30-shot							
	AP	bAP	nAP	nAP_{50}	nAP_{75}	nAPs	nAPm	nAPl	AP	bAP	nAP	nAP_{50}	nAP_{75}	nAPs	nAPm	nAPl
Prior	31.5	38.8	9.6	17.8	9.2	3.8	9.3	16.5	32.5	38.8	13.6	24.3	13.3	4.7	11.9	21.4
DiGeo	32.0	39.2	10.3	18.7	9.9	4.5	10.0	16.8	33.1	39.4	14.2	26.2	14.8	5.3	13.1	23.9

(11-100), and frequent (>100). Following [18], apart from the precision for all classes (AP) on the validation set, we also report the precision for each group, *i.e.*, AP_r, AP_c, and AP_f. Meanwhile, following a common setup, we try two different backbones ResNet50 and ResNet101.

Here we try two different baseline, TFA and ACSL. We do acknowledge other related research on LVIS such as EFL [6] and LOCE [2]. However, these approaches are developed on Mask-RCNN framework, *i.e.*, both object detection and object segmentation are trained. Since object segmentation introduces extra supervision signals, while our focus is main on object detection, we thus choose ACSL as the baseline.

Comparing with ACSL, TFA also focus on object detection only but ACSL 1) applies a two step training strategy and 2) use the model pretrained on MSCOCO as initialization. In contrast, TFA only uses ImageNet-pretrained model to initialize the feature extractor. Meanwhile, it follow the configuration regarding learning rate and training epochs in the 1x Baseline but apply it on base training stage. As such, we consider both of these two setups. As such, we follow the training steps ACSL and use model pre-trained on MS COCO as initialization. From the Table M2, DiGeo can achieve consistent gain on two cases.

Table M2. Performance comparison of LVIS dataset (Full Table)

Approach	ResNet-50				ResNet-101			
	AP	APr	APc	Apf	AP	APr	APc	Apf
V0.5								
1x Baseline	22.7	10.6	22.0	28.0	24.5	13.1	23.9	30.0
TFA w/ fc [18]	24.1	14.9	23.9	27.9	-	-	-	-
TFA w/ cos [18]	24.4	16.9	24.3	27.7	-	-	-	-
DiGeo	24.9	17.3	24.6	28.5	26.8	18.5	26.8	30.1
RFS [4]	24.9	14.4	24.5	29.5	-	-	-	-
Focal Loss [8]	22.0	10.5	22.4	25.9	-	-	-	-
EQL [15]	25.1	11.9	26.0	29.1	26.1	11.5	27.1	30.5
BAGS [7]	26.0	17.7	25.8	29.5	26.4	16.8	25.8	30.9
ACSL [17]	26.4	18.6	26.4	29.4	27.5	19.3	27.6	30.7
DiGeo	26.7	18.9	27.0	29.0	27.9	19.5	28.0	31.0
V1.0								
1x Baseline	19.3	6.4	17.1	27.6	21.1	10.1	21.7	25.8
DiGeo	22.5	12.4	20.6	26.8	24.4	16.6	22.8	28.0

The configuration of 1x Baseline can be found in the TFA official repo.

3. Discussion

3.1. Decoupling localization from classification.

Consistent with the observation in [11], by enhancing inter-class separation and intra-class compactness, the detection scores are improved. However, the features for localization should still be class-independent (*e.g.*, bus and elephant has similar shape). From the implementation details, a projector is set where its input & output are used for localization & classification separately. Then, sharing the features for localization and classification will lead to slight performance drop (*i.e.*, AP_{50} 74.0, nAP_{50} 55.6). As such, it is important to decouple the features for localization and classification and employing a simple linear projector has been shown to be userful.

Table M3. Ablation study of Background Design.

Idx	N_-	Fixed	AP_{50}	bAP_{50}	nAP_{50}
1	1	✓	74.9	81.0	56.4
2	5	✓	73.5	81.3	50.2
3	1		74.9	81.0	56.4
4	5		74.9	80.9	56.7
5	10		74.9	80.9	57.0
6	20		74.6	81.1	55.2

3.2. Design of Background class

An object detector should reject the background and not recognize it as any foreground object. As such, a background class c_- is set as a placeholder and is trained to have high similarity with background proposals. Different from foreground objects, as background proposals can be diverse, we considered different strategies in designing the background class center.

We first choose to separate the design of $W_b \cup W_n$ and $\mathbf{w}_{c_-}$, *i.e.*, deriving fixed offline weights for $\mathcal{C}_b \cup \mathcal{C}_n$ only but learn the weight $\mathbf{w}_{c_-}$. Then, we follow the open-set strategy [22] to set multiple background centers $W_- = \{\mathbf{w}_{c_-}^{(i)}\}_{i=1}^{N_-}$ where N_- is the number of background centers where the maximum logit, *i.e.*, $\max_{1 \le i \le N_-}(\mathbf{x}^T \mathbf{w}_{c_-}^{(i)})$, is used in classification. As compared in Table M3, having more learnable class centers can introduce trivial performance improvement but will drop clearly when N_- is too large. However, when we directly set the classifier for the all classes ,*i.e.*, $W_n \cup W_b \cup W_-$ as ETF , the performance drops when $N_- > 1$.

In practice, we observe all learnable negative weights W_- are trained to separate from the $W_b \cup W_n$ where the weights in W_- are still close to each other such that the diversity of background features are preserved indirectly. Instead, having all negative weights maximally separated from each other assume background features is very diverse and make the model hard to learn. As such, we choose to set $N_- = 1$ and adjust margins through self-distillation to maintain the diversity properly.

3.3. More Visualization

As shown in Fig. M1, we visualize the classifier centers by their pair-wise cosine similarity when they are learned from scratch. Fig. M1(a) is the same as the Fig. 3(b) in the main paper but the background class center is also included (the rightmost and the bottom one). We can then see that when we have both base and novel annotation in the train set, the class centers can be trained to distance from all of the background classes. However, when we only use novel classes during the adaptation stage (Fig. M1(b)), the negative class center can be close to the novel class centers. Meanwhile, when we use the full set for training from scratch, we can see that the applying either RFS or adding margins can help with separating the novel class centers from the background class centers, while adding margins is more important.

The foreground class names (sorted by decreasing order) are person, chair, car, bottle, dog, potted plant, cat, boat, sheep, aeroplane, bicycle, tv monitor, horse, dining table, train, motorbike, cow, bus, bird, sofa.

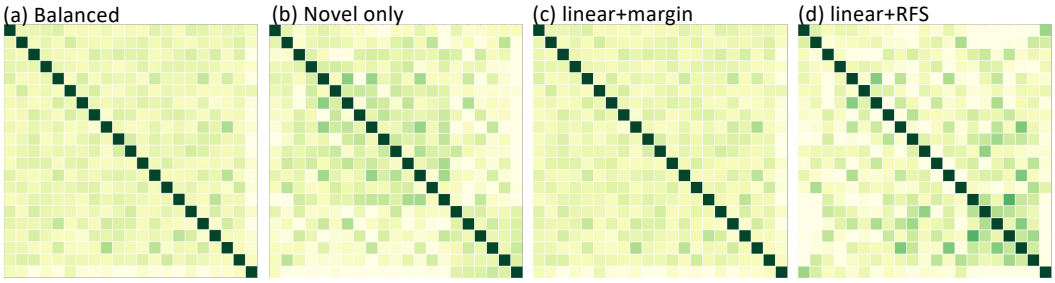

Figure M1. Visualization classifier centers.

4. Comparison with related work

In this section, we provide in-detail comparison with a few representation works to highlight our motivation and contribution. All of the approached listed below have been briefly mentioned in the sections of Related Work and Experiment.

CME [5] similarly employs a margin equilibrium strategy in the few-shot finetuning. The determination of the margin value is based on the degree of feature disturbance which is measured by the scale of gradient among augmented samples. Meanwhile, CME is motivated by the trade off between margins of base classes and the variance of novel classes.

However, we have used the geometric property of Simplex ETF to maximally separate the feature clusters for all classes. In this way, we decouple the learning for inter-class separation and intra-class compactness and only tighten the feature cluster to the corresponding class centers to reach a balanced distribution. As such, we can learn discriminative features for all of the classes even on an extremely imbalanced dataset.

In addition, CME is still trained on the balanced dataset $\mathcal{D}_b^- \cup \mathcal{D}_n$ and the so-called margin equilibrium is realized when the model is trained on a balanced set. Thus, CME may still forget the base knowledge. Instead, our margins are for all classes based on the prior of instance distribution and our approach is orthogonal to CME. Furthermore, the margin estimation strategy in CME can be used as an alternative of our self-distillation in margin adjustment.

Negative margin on few-shot classification is studied in [10] and reveals the trade-off of classification accuracy between base recognition and novel recognition. Namely, for a feature extractor pre-trained on *base* classes, if the model achieves better test accuracy on the base classification, the adaptation accuracy towards classification accuracy is then minimized. As such, a comprehensive study is provided in [10].

In contrast, we focus on few-shot object detection and aims to improve the few-shot adaptation efficiency without scarifying the performance of base detection. We always add positive class-specific margins to all classes where the margin values are adaptively learned during network training.

LOCE [2] is applied on long-tail object detection, which is a more general case of generalized few-shot object detection (*i.e.*, in GFSOD, the imbalance between base set and novel set is more significant and thus more challenging). A common problems discussed in LOCE and our paper is that the instance distribution of classes cannot be directly used to estimate the margins.

As such, LOCE discards the prior and introduce the Equilibrium loss to use the mean classification score to determine the margin. In addition, they proposed a complex memory-augmented feature sampling to facilitate the network training. In contrast, we clearly discuss and decouple the training objective for inter-class separation and intra-class compactness.

We consider the distribution of classifier weights in conventional training and use ETF as a fixed classifier. In this case, we used the assigned weights to guide the separation of feature clusters between different classes, and then apply different margins to push the features to the assigned centers. As we apply margins to facilitate the balanced distribution, we can use the instance distribution as prior and use a simple knowledge distillation to adjust the margins and facilitate training.

Margin modification techniques such as BALMS [13] and Seesaw loss [16] has been proposed. Specifically, BALMS considers the boundary shifting problem in long-tailed classification/segmentation and present a meta-sampling strategy to re-estiamte the boundary indicated in the Softmax function. Seasaw loss defines a compensation factor in vanilla cross entropy loss to balance the error for different classes. In both case, they in effect count on the real-time (online) distribution of selected samples during the training and then adjust the loss. Instead, we focus on the inter-class separation and intra-class compactness to guide the training of features, *i.e.*, re-arranging the feature distribution from the perspective of feature geometry. In addition, the margin modification techniques can be used as an alternative of our margin adjustment strategy for the intra-class compactness only.

Connection with FSCE In FSCE [14], the authors has provided a strong baseline by adjusting the hyper-parameters in RPN and proposal selection. We have tried to apply it in our framework but the performance drops. As such, we still follow the hyper-paramter setting in TFA. Meanwhile, it also demonstrates that the observation in FSCE is only available in the two-step based training strategy such as TFA, and cannot be generalized to a universal case.

Furthermore, FSCE proposed a contrastive encoding approach and treats the proposals as augmentation of the same instance. However, we have also add the contrastive loss in our approach and observed that it may help improve the nove

detection slightly but hurt the base detection significantly. We think the reason is that the data distribution is extremely imbalanced and and the contrastive loss cannot help.

References

[1] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4527–4536, 2021. 2

[2] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3417–3426, 2021. 3, 5

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning (adaptive computation and machine learning series). *Cambridge Massachusetts*, pages 321–359, 2017. 2

[4] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 2, 3

[5] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2021. 5

[6] Bo Li, Yongqiang Yao, Jingru Tan, Gang Zhang, Fengwei Yu, Jianwei Lu, and Ye Luo. Equalized focal loss for dense long-tailed object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6990–6999, 2022. 3

[7] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020. 3

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[10] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European conference on computer vision*, pages 438–455. Springer, 2020. 5

[11] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. 1, 3

[12] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020. 2

[13] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 5

[14] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7352–7362, 2021. 5

[15] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. 3

[16] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9695–9704, 2021. 5

[17] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3103–3112, 2021. 3

[18] Xin Wang, Thomas Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, pages 9919–9928. PMLR, 2020. 2, 3

[19] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *European conference on computer vision*, pages 456–472. Springer, 2020. 2

[20] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European conference on computer vision*, pages 192–210. Springer, 2020. 2

[21] 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/CVF International Conference on Computer Vision*, pages 9577–9586, 2019. 2

[22] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2021. 4