Beyond Max-Margin: Class Margin Equilibrium for Few-shot Object Detection

Bohao Li[1*] Boyu Yang[1*] Chang Liu[1] Feng Liu[1] Rongrong Ji[2,3,4] Qixiang Ye[1,4†]
PriSDL, EECE, University of Chinese Academy of Sciences, 100049, China.[1]
MAC, Department of Artificial Intelligence, School of Informatics, Xiamen University, 361005, China.[2]
Institute of Artificial Intelligence, Xiamen University, 361005, China.[3]
Peng Cheng Laboratory, Shenzhen, China.[4]
{libohao20, yangboyu18, liuchang615, liufeng20}@mails.ucas.ac.cn rrji@xmu.edu.cn
qxye@ucas.ac.cn

Abstract

Few-shot object detection has made substantial progress by representing novel class objects using the feature representation learned upon a set of base class objects. However, an implicit contradiction between novel class classification and representation is unfortunately ignored. On the one hand, to achieve accurate novel class classification, the distributions of either two base classes must be far away from each other (max-margin). On the other hand, to precisely represent novel classes, the distributions of base classes should be close to each other to reduce the intra-class distance of novel classes (min-margin). In this paper, we propose a class margin equilibrium (CME) approach, with the aim to optimize both feature space partition and novel class reconstruction in a systematic way. CME first converts the few-shot detection problem to the few-shot classification problem by using a fully connected layer to decouple localization features. CME then reserves adequate margin space for novel classes by introducing simple-yet-effective class margin loss during feature learning. Finally, CME pursues margin equilibrium by disturbing the features of novel class instances in an adversarial min-max fashion. Experiments on Pascal VOC and MS-COCO datasets show that CME significantly improves upon two baseline detectors (up to 3 $\sim$ 5% in average), achieving state-of-the-art performance. Code is available at https://github.com/Bohao-Lee/CME.

1. Introduction

In the past few years, we witnessed the great progress of visual object detection [16, 17, 3, 18]. This is attributed to the availability of large-scale datasets with precise annotations and convolutional neural networks (CNNs) capable of absorbing the annotation information. However, annotating a large amount of objects is expensive and laborious [22, 23, 33, 34]. It is also not consistent with cognitive learning, which can build a precise model using few-shot supervisions [20].

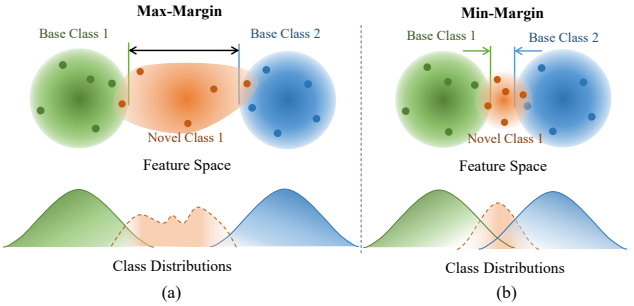


Figure 1. The contradiction of representation and classification in few-shot object detection. (a) To separate the classes with each other, either two base classes requires to be far away from each other (max-margin), which aggregates the intra-class distance of novel classes. (b) To precisely represent novel classes, the distributions of base classes should be close to those of novel classes (min-margin), which improves the difficulty of classification.

Few-shot detection, which simulates the way that human learns, has attracted increasing attention. Given base classes of sufficient training data and novel classes of few supervisions, few-shot detection trains a model to simultaneously detect objects from both base and novel classes. To this end, a majority of works divided the training procedure to two stages: base class training (representation learning) and novel class reconstruction (meta training). In representation learning, the sufficient training data of base classes are used to train a network and constructs a representative feature space. In meta training, the network is finetuned so that the novel class objects can be represented within the feature space. Among the earliest work, Kang *et al*. [7] pro-

*Equal Contribution.
†Corresponding Author.

posed applying channel-attended feature re-weighting for semantic enforcement. In the two-stage framework, Wang *et al*. [27] and Yan *et al*. [30] contributed early few-shot detection methods. Wang *et al*. [24] and Wu *et al*. [28] proposed freezing the backbone network and reconstructing the novel classes using the classifier weights during detector finetuning.

Despite the substantial progress, the implicit contradiction between representation and classification is unfortunately ignored. To separate the classes, the distributions of two base classes requires to be far away from each other (max-margin), which however aggregates the diversity of novel classes, Fig. 1(a). To precisely represent novel classes, the distributions of base classes should be close to each other (min-margin), which causes the difficult of classification, Fig. 1(b). How to simultaneously optimize novel class representation and classification in the same feature space remains to be elaborated.

In this paper, we propose a class margin equilibrium (CME) approach, with the aim to optimize feature space partition for few-shot object detection with adversarial class margin regularization. For the object detection task, CME first introduces a fully connected layer to decouple localization features which could mislead class margins in the feature space. CME then pursues a margin equilibrium to comprise representation learning and feature reconstruction. Specifically, during base training, CME constructs a feature space where the margins between novel classes are maximized by introducing class margin loss. During network finetuning, CME introduces a feature disturbance module by truncating gradient maps. With multiple training iterations, class margins are regularized in an adversarial min-max fashion towards margin equilibrium, which facilities both feature reconstruction and object classification in the same feature space.

The contributions of this study include:

- We unveil the representation-classification constriction hidden in few-shot object detection, and propose a feasible way to alleviate the constriction from the perspective of class margin equilibrium (CME).
- We design the max-margin loss and feature disturbance module to implement class margin equilibrium in an adversarial min-max fashion.
- We convert the few-shot detection problem to a few-shot classification problem by filtering out localization features, We improve the state-of-the-art with significant margins upon both one-stage and two-stage baseline detectors.

2. Related Works

2.1. Object Detection

CNN-based methods have significantly improved the performance of object detection. While the one-stage methods, $e.g.$, YOLO [16, 17] and SSD [14], have higher detection efficiency, the two-stage methods, $e.g.$, Faster R-CNN [3, 18] and FPN [10] report higher performance, usually. Relevant CNN-based detectors provided fundamental techniques, $e.g.$, RoI pooling [3] and multi-scale feature aggregation [10], which benefit few-shot object detection. However these methods generally require large amounts of training data, which hinders their applications in practical scenarios.

2.2. Few-shot Learning

Existing few-shot learning methods can be broadly categorized as either: metric learning [21, 19, 4, 32, 31, 12, 13, 9], meta-learning [26, 15, 2, 6], or data augmentation [5, 25]. Metric learning methods train networks to predict whether two images/regions belong to the same category. Meta-learning approaches specify optimization or loss functions which force faster adaptation of parameters to new categories with few examples. The data argumentation methods learn to generate additional examples for unseen categories. In the metric learning framework, prototypical models converted the spatial semantic information of objects to convolutional channels. In existing studies, it was observed that class margin has a great impact to classifiers when required to guarantee model discriminability under few supervisions. Li *et al*. [8] proposed adaptive margin loss to improve model generalization ability. They further developed a class-relevant additive margin loss considering the semantic similarity between image pairs. However, solely pursuing max-margin could be infeasible because the novel classes required to be reconstructed with the base classes and large margin would improve the diversity of novel class samples. Liu *et al*. [11] introduced negative class margin to benefit representation of novel classes. Existing studies inspire us to re-think the max-margin principle in few-shot settings, to comprise discriminability and representation capability of features.

2.3. Few-shot Object Detection

Following the meta learning methods, Kang *et al*. [7] contributed an early few-shot detection method, which fully exploited training data from base classes while quickly adapting the detection prediction network to predict novel classes. Yan *et al*. [30] proposed meta-learning over RoIs, enabling Faster R-CNN be a meta-learner for few-shot detection. Wu *et al*. [28] proposed positive sample refinement to enrich object scales for few-shot detection. Despite of the progress, the discriminability and representation equi-

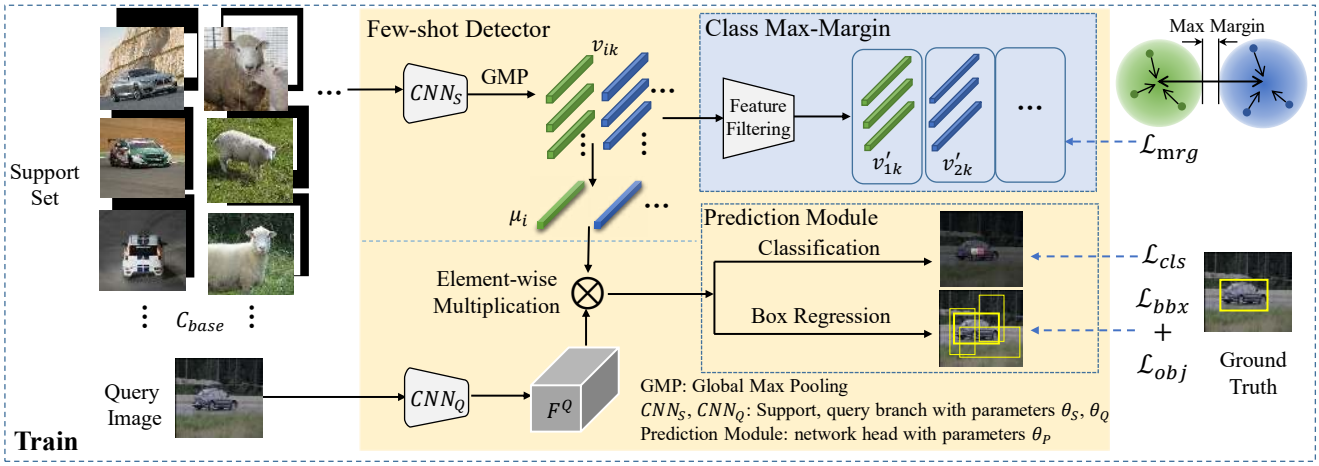


Figure 2. Framework of the proposed few-shot detection which consists of a support branch and a query branch. This figure only illustrates base class training driven by detection loss and max-margin loss.

librium between novel and base classes remain unsolved. Furthermore, most existing methods treat few-shot detection as a few-shot classification problem, ignoring the role of features for object localization.

3. The Proposed Approach

3.1. Few-shot Detection Framework

Problem Definition. Given base classes C_{base} of sufficient training data and novel classes C_{novel} of few supervisions, few-shot detection aims to train a model that can simultaneously detect objects from both base and novel classes. As shown in Fig. 2, a detection network is first trained with C_{base} to construct feature representation. The network is then finetuned with both C_{base} and C_{novel} to represent the few-shot instances from novel classes. In what follows, we describe the proposed method by using meta YOLO [7] as the baseline detector. Our approach can be applied to two-stage few-shot detectors [28] in a plug-and-play fashion.

For both detector training and finetuning, the dataset $\mathcal{D}$ (either base classes or novel classes) is divided to a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. $\mathcal{D} = \mathcal{S} \cup \mathcal{Q} = \{I^S, M^S\} \cup \{I^Q, M^Q\}$, where I^S denotes support images with a mask annotations M^S, which are generated according to object bounding-boxes. I^Q denotes the query images with ground-truth bounding boxes M^Q. Given N classes, each of which has K annotated instances, $\mathcal{S}$ can be further denoted as $\{\{I^S_{ik}, M^S_{ik}\}, i = 1, ..., N, k = 1, ..., K\}$, where i indexes the class and k indexes instance, and $I^S_{ik} \in \mathbb{R}^{W \times H \times 3}$.

The network consists of a support branch (Fig. 2 (upper)) a query branch (Fig. 2 (lower)). On the support branch, the support images I^S and their bounding-boxes M^S are fed to the CNN to extract convolutional feature maps. With a global max pooling (GMP) operation, the feature maps are squeezed to prototype vectors $v_{ik} = f_{\theta_S}(I^S \oplus M^S)$, where $f_{\theta_S}(\cdot)$ denotes the network of the support branch with parameters θ_S and $\oplus$ the concatenate operation. The mean prototypes for the i-th class is calculated by $\mu_i = \frac{1}{K}\sum_{k=1}^{K} v_{ik}$, indicating the semantics of the object class. On the query branch, convolutional features $F^Q = f_{\theta_Q}(I^Q)$ are extracted for the query images I^Q, where θ_Q denotes the query branch network parameters. The features are activated by multiplying with the prototype vectors $\{\mu_i\}$ through a pixel-wise multiple operation. The activated features are fed to a prediction (classification and box regression) module and output $\mathcal{P}_{\theta_P}(F^Q \otimes \mu_i)$, where θ_P denotes the prediction module parameters, $\otimes$ means element-wise multiplication. For the general object detection task, the target of the prediction results are expected to match the ground-truth bounding box area M^Q, through minimizing the following loss

$$\arg\min_{\theta} \mathcal{L}_{det}(\mathcal{P}_{\theta_P}(F^Q \otimes u_i), M^Q), \tag{1}$$

where $\theta = \theta_S \cup \theta_Q \cup \theta_P$. $\mathcal{L}_{det}$ is the object detection loss, defined as $\mathcal{L}_{det} = \mathcal{L}_{cls} + \mathcal{L}_{bbx} + \mathcal{L}_{obj}$, where $\mathcal{L}_{cls}$, $\mathcal{L}_{reg}$, and $\mathcal{L}_{obj}$ respectively denote the classification, regression and anchor confidence loss [7, 17].

Feature Filtering. For object detection, the convolutional features incorporate both localization features and classification features. While the classification features are class dependent, the localization features are independent to object classes, and therefore tend to perturb class margins. To filtering out the localization features, a fully connected layer is used to decouple localization features, as $v'_{ik} = \mathcal{FC}(v_{ik})$, to convert the few-shot detection problem to a pure few-shot classification problem, Fig. 2. Driven by the max-margin loss, the localization features are filtered out during detector training.

3.2. Base Training: Class Max-margin

Max-Margin Loss. In the base training stage, the sufficient data of base classes are used to train the network and construct a representative feature space. As the object detection network is a discriminative model, the whole feature space will be divided into multiple sub-spaces each of which is occupied with a class. In the finetuning stage, novel classes will be embedded to the feature space, often to the margin space between base classes. To avoid aliasing, the margin space between base classes should be big enough to accommodate novel classes, Fig. 1(a), $i.e.$, class max-margin.

To pursue class max-margin, prototype vectors of the base classes require to be close to their mean prototypes ($i.e.$, minimum intra-class variance) while those of different classes be far away from each other ($i.e.$, maximum inter-class distance). Given the prototype vector $v_{ik}^{'}$ for the k-th instance, the mean prototype vector of the i-th class is calculated as $\mu_i^{'} = \frac{1}{K}\sum_{j=0}^{k-1} v_{ij}^{'}$, which represents the semantics of the class. The intra-class distance is $D_i^{Intra} = \sum_{j=0}^{K-1} ||v_{ij}^{'} - \mu_i^{'}||_2^2$. The inter-class margin distance is calculated as $D_i^{Inter} = \min_{j,j\neq i} ||\mu_j^{'} - \mu_i^{'}||_2^2$. Generally speaking, margin is defined as the distance between the decision boundary and the sample of shortest distance to the boundary. For the feature space constructed by CNN, it is hard to directly calculate the margin $\mathcal{M}_{i,i'}$ between two classes. As an approximation, we first calculate the upper and lower bounds of $\mathcal{M}_{i,i'}$ and have

$$D_i^{Inter} - D_i^{Intra} - D_{i'}^{Intra} \leq \mathcal{M}_{i,i'} \leq D_i^{Inter}, \quad (2)$$

which indicates that the upper bound of margin is the inter-class distance while the lower bound is the inter-class distance subtracting intra-class distance. According to Eq. 2, max-margin can be approximated by maximizing the upper and lower bounds of margins, as

$$\arg\max_{\theta} \mathcal{M}_{i,i'} \simeq \arg\max_{\theta} \mathcal{L}_{mrg} = \frac{\sum_i^N D_i^{Intra}}{\sum_i^N D_i^{Inter}}, \quad (3)$$

where N denotes the class number.

Detector Training. In base training, a support set and a query set are constructed for base classes by randomly selecting training samples, $\mathcal{S} \cup \mathcal{Q} \subseteq \mathcal{D}_{C_{base}}$. The detection network is trained by optimizing both the object detection loss and the max-margin loss, as

$$\arg\min_{\theta} \mathcal{L}_{trn} = \mathcal{L}_{det} + \lambda \mathcal{L}_{mrg}, \quad (4)$$

where $\lambda = 1.0$ is an experimentally determined regularization factor to balance the two loss functions.

3.3. Finetuning: Margin Equilibrium

Finetuning refers to a meta learning procedure, which uses few-shot novel class data to update network parameters. However, without sufficient data, the novel classes cannot significantly change the feature representation, so that novel classes required to be represented by the features learned upon base classes [11]. According to Eq. 3, the margins between base classes in the feature space are required to be large, which improves the D^{Intra} and preserve sufficient margin space for novel classes. However, when the margin between base classes is large, the samples/features from novel classes can be of large diversity, Fig. 1(b), which aggregates the difficulty to train the detection model for novel classes. To solve this problem, we propose the margin equilibrium strategy based on feature disturbance.

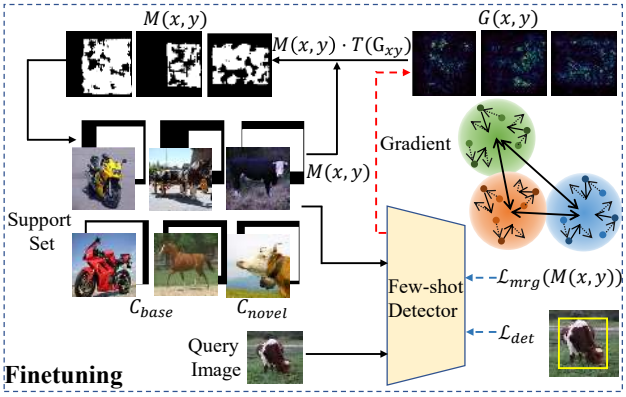


Figure 3. Network finetuning with feature disturbance. Feature disturbance is implemented by truncating the gradient maps and re-sampling the training images.

Feature Disturbance. Feature disturbance defines an online data augmentation procedure according to the gradient maps of samples. During finetuning, images of base and novel classes are simultaneously fed to the network detector training driven by the margin loss and detection loss, Fig. 3. During the back-propagation procedure, the gradient map of a support image is calculated by $G(x,y) = ||\frac{\partial \mathcal{L}_{ftn}}{\partial I(x,y)^S}||$, where $G(x,y) \in \mathbb{R}^{W\times H}$ and $||\cdot||$ denotes the norm operation. $\mathcal{L}_{ftn}$ denotes the finetuning loss defined on the detection loss and margin loss. (x,y) is the pixel location. According to the characteristics of CNN, the pixels of larger gradients correspond object parts of larger discrimination ability and contribute more to reduce the finetuning loss. During detector training, the disturbance procedure is carried out to truncate the pixels of large gradient and disturb the finetuned features. This is implemented by re-sampling the ground-truth mask according to the gradient map, as

$$T(G(x,y)) = \begin{cases} 0 & G(x,y) \geq \tau(G(x,y)) \\ 1 & otherwise \end{cases}, \quad (5)$$

where τ is a threshold which controls the ratio of feature disturbance. In experiments, τ is set to be a dynamic threshold so that top-15% pixels of large gradient are set to 0. For

Algorithm 1 Detector training and finetuning with CME

Input:
Support set $\mathcal{S} = \{I^S, M^S\}$, query set $\mathcal{Q} = \{I^Q, M^Q\}$;
Output:
Network parameters $\theta = \theta_S \cup \theta_Q \cup \theta_P$;
Training:
for $(I^S, M^S, I^Q, M^Q$ in $\mathcal{D}_{C_{base}})$ **do**
Predict detections and calculate detection loss $\mathcal{L}_{det}$ by Eq. 1;
Calculate margin loss $\mathcal{L}_{mrg}$ by Eq. 3
Update θ to minimize the training loss $\mathcal{L}_{trn}$ by Eq. 7;
end for
Finetuning:
for (each I^S, M^S, I^Q, M^Q in $\mathcal{D}_{C_{base}} \cup D_{C_{novel}}$) **do**
for (iteration **do**
Predict detections and calculate the detection loss $\mathcal{L}_{det}$ by Eq. 1;
Calculate margin loss $\mathcal{L}_{mrg}$ by Eq. 3
Update θ to minimize the finetuning loss $\mathcal{L}_{trn}$ by Eq. 7 with back-propagation;
Update support mask M^S according to Eq. 6.
end for
end for

feature disturbance, the support mask M^S are updated according to the gradients map, as

$$M^S(x,y) \leftarrow M^S(x,y) \cdot T(G(x,y)). \tag{6}$$

Margin Equilibrium. With the above defined feature disturbance strategy, the novel classes C_{novel} are combined with the base classes C_{base} for network parameter finetuning. Given a batch of support and query images, the network parameters are updated for a few iterations. The iteration number relies on the number (K) of training images in each novel class. In each finetuning iteration, the support mask is re-sampled, and the prototype vector is calculated by $v_{ik} = f_{\theta_S}(I^S \oplus M^S)$ guided by the re-sampled mask M^S. Accordingly, detector finetuning is performed by minimizing the loss function

$$\mathcal{L}_{ftn} = \mathcal{L}_{det} + \lambda \mathcal{L}_{mrg}(M^S(x,y)). \tag{7}$$

Meanwhile, according to Eq. 6, the features are disturbed so that prototype vectors within the feature space are re-sampled to occupy the class margin.

During back-propagation, network parameters are updated to maximize the class margins $\mathcal{M}_{i,i'}$ by optimizing Eq. 3. In the procedure, D_i^{Inter} increases while D_i^{Intra} decreases. During forward propagation, the support mask M^S is updated according to Eq. 6 and the support image is re-sampled. In this way, the discriminative pixels on the image/features are erased so that the discrimination power of prototype vectors v_{ik} generated by the re-sampled features decreases. As a result, the upper bound D^{Inter} decreases and so does the margin $\mathcal{M}_{i,i'}$. This actually defines an adversarial learning procedure for min-max margin, as

$$\begin{cases} \arg\max_{\theta} \mathcal{M}_{i,i'}, & \text{Back Propagation} \\ \arg\min_{M^S(x,y)} \mathcal{M}_{i,i'}, & \text{Forward Propagation} \end{cases}, \tag{8}$$

which pursues class margin equilibrium for base classes and embedded novel classes, detailed in Algorithm 1.

Table 1. Ablation study of CME modules for few-shot object detection on Pascal VOC novel classes (split-1). "MM" denotes max-margin, "FF" feature filtering, "FD" feature disturbance and "avg. Δ" average performance improvements.

Module			Shot					avg. Δ
MM	FF	FD	1	2	3	5	10	
			14.8	15.5	26.7	33.9	47.2	
✓			13.5	21.9	28.5	40.2	47.0	+2.6
✓	✓		13.2	23.4	29.9	43.1	**49.8**	+4.3
✓	✓	✓	**17.8**	**26.1**	**31.5**	**44.8**	47.5	**+5.9**

4. Experiments

4.1. Experimental Setting

Datasets. The proposed CME approach for few-shot object detection is evaluated on Pascal VOC 2007, VOC 2012 and MS COCO, following the settings in Meta YOLO [7]. The object categories in the datasets are divided into two groups: base classes with adequate annotations and novel classes with K-shot annotated instances. During base training process, the network is optimized upon using the training data of base classes. During finetuning, the network is optimized by K-shot instances of each novel and base classes. For Pascal VOC, the whole dataset is partitioned into 3 splits for cross validation. In each split, 5 classes are selected as novel classes and the rest of the 15 classes are base classes. The number of annotated instances K is set as 1, 2, 3, 5, and 10. For MS COCO, 20 classes are selected as novel ones and the remaining 60 classes are set as base ones.

Implementation Details. As a plug-and-play module, CME is fused with the one-stage detector (Meta YOLO [7]) and the two-stage detector (MPSR [28]) for evaluation. In what follows, the experimental analysis and ablation study are based on the Meta YOLO detector, which is implemented with PyTorch 1.0 and run on Nvidia Tesla V100 GPUs. During training, four data augmentation strategies are used, including size normalization, horizontal flipping, random cropping, and random resizing. The network is optimized by the SGD algorithm with initial learning rate of

Table 2. Ablation study of number of output channels in the feature filtering module on Pascal VOC Novel class (split-1).

Num. \ Shots	1	2	3	5	10	avg. Δ
W/O FF	13.5	21.9	28.5	40.2	47.0	
1024	13.6	19.9	27.5	36.0	48.2	-1.2
512	13.2	**23.5**	**29.9**	**43.1**	**49.8**	**+2.9**
256	**16.3**	22.6	27.3	37.7	47.9	+1.3

Table 3. Ablation study of self-disturbance on Pascal VOC novel classes (split-1).

Method		Shots					avg. Δ
C_{Novel}	C_{Base}	1	2	3	5	10	
		13.2	23.4	29.9	43.1	**49.8**	
✓		13.0	22.7	29.5	43.5	49.5	-0.2
	✓	**17.8**	**26.1**	**31.5**	**44.8**	47.5	**+1.7**
✓	✓	16.0	24.9	31.0	43.9	49.5	+1.2

Table 4. Comparison of feature disturbance strategies on Pascal VOC novel classes (split-1). "trun." denotes truncation.

Manner \ Shots	1	2	3	5	10	avg. Δ
w/o disturbance	13.2	23.4	29.9	43.1	**49.8**	
Random sample	15.2	23.2	31.4	42.2	48.8	+0.3
Random crop	15.7	21.9	**32.5**	43.9	46.6	+0.2
Feature trun.	14.5	23.1	31.8	43.6	48.8	+0.5
Gradient trun.	**17.8**	**26.1**	31.5	**44.8**	47.5	**+1.7**

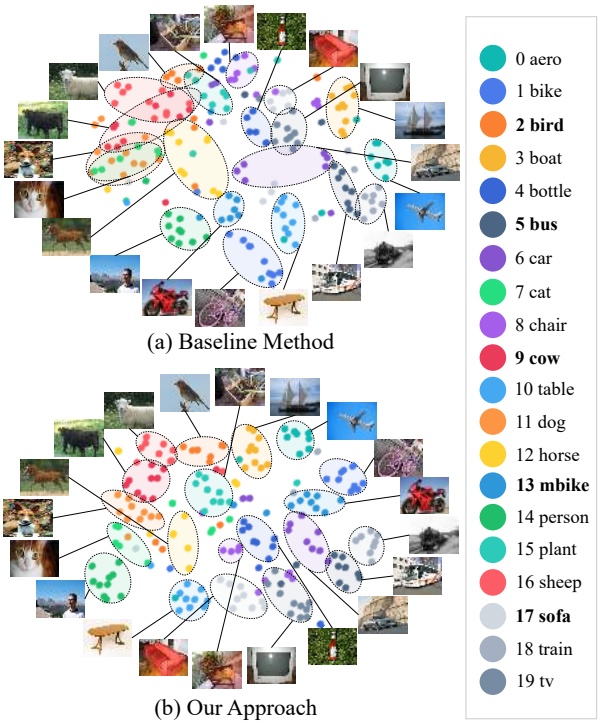
0.001, momentum of 0.9 for 80,200 iterations in base training and 2000 iterations in finetuning. There are 64 query images per batch and 2 support images for each class.

4.2. Ablation Study

Table 1 shows the efficacy of the main components of CME for few-shot object detection with different shot settings on Pascal VOC novel classes (split-1). With the max-margin loss, the average performance gain is 2.6% compared with baseline method. By using feature filtering, the performance gain increases to 4.3%. With the feature disturbance for class margin equilibrium, the performance gain increases to 5.4%. It shows that CME achieves significant improvement over the baseline method.

Max-Margin. From Table 1, we can find that max-margin makes effect in 2,3,5 shots setting while being invalid in 1-shot setting. It validates that given limit training data, the increase of class margins is worthless for which is not conducive to the reconstruction of novel class.

Feature Filtering. In Table 2, experiments are conducted to determine the number of output feature channels in the feature filtering module. It reveals that 512 channels reports the best result. 1024 output channels is redundant that makes the margin loss invalid. 256 output channels is insufficiency because of the depression of the feature representation. That is to say that the FC layer requires a slightly smaller output channel number (compared with 1024 input channels) to filter out localization related features.



(a) Baseline Method

(b) Our Approach

Figure 4. t-SNE visualization of prototypes produced by the baseline method [7] and our proposed CME approach. The novel classes are in bold font.

Feature Disturbance. In Table 3 and Table 4, ablation studies are carried out to compare the feature disturbance strategies. Table 3 shows that it is better to disturb the prototypes of base class C_{Base} without C_{Novel}. According to Eq. 8, the margin equilibrium is implemented by feature disturbance. The disturbance of base classes depresses margins between base classes which benefit the representation of novel class. Conversely, the disturbance of novel classes may degenerate the representation discrimination as the margin space turns limited.

Table 4 validates that gradient truncation significantly outperforms feature truncation and feature crop strategies among those of feature disturbance method since it is an adversarial min-max margin manner against the gradient rather than a simple data augmentation strategy.

Table 5. Detection performance comparison on the Pascal VOC dataset.

Framework	Method \ Shots	Novel set 1					Novel set 2					Novel set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
YOLO	LSTD [1]	8.2	11.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3
	Meta YOLO [7]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	**21.3**	25.6	28.4	42.8	45.9
	MetaDet [27]	17.1	19.1	28.9	35.0	**48.8**	**18.2**	**20.6**	25.9	30.6	**41.5**	20.1	22.3	27.9	41.9	42.9
	CME (Ours)	**17.8**	**26.1**	**31.5**	**44.8**	47.5	12.7	17.4	**27.1**	**33.7**	40.0	15.7	**27.4**	**30.7**	**44.9**	**48.8**
F-RCNN	MetaDet [27]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
	Meta R-CNN [30]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
	Viewpoint [29]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
	TFA w/cos [24]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	**49.5**	49.8
	MPSR [28]	**41.7**	42.5	**51.4**	52.2	**61.8**	24.4	29.3	39.2	39.9	**47.8**	**35.6**	**41.8**	42.3	48.0	49.7
	CME (Ours)	41.5	**47.5**	50.4	**58.2**	60.9	**27.2**	**30.2**	**41.4**	**42.5**	46.8	34.3	39.6	**45.1**	48.3	**51.5**

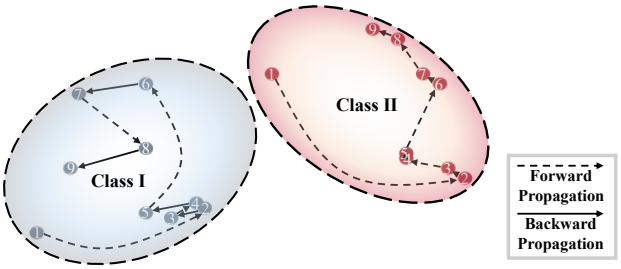


Figure 5. t-SNE visualization of the feature prototype evolution of two object classes during the finetuning stage. While the dashed curves denote feature disturbance routes (Forward Propagation), the solid line segments denote the routes driven by the fine-turning loss with max-margin regularization (Backward Propagation). In the finetuning stage, the instance features of novel classes form sub-spaces in the feature space learned on base classes.

4.3. Model Analysis

In Fig. 4, we compare the distributions of feature prototypes learned by the baseline method and our CME approach. One can see that CME optimizes novel class embeddings by reserving adequate margin space for novel classes when learning the feature representation. Furthermore, CME optimizes feature space partition by pursuing margin equilibrium in an adversarial min-max fashion when finetuning the network with the novel classes. While the baseline method is confused with the novel class "Cow" and the base classes "Cat", "Dog", "Sheep", and "Horse", CME clearly distinguishes them and reduces the overlap between classes. It proves that the CME can improve the representation capacity of the feature space for better object detection.

Fig. 5 visualizes the evolution of two prototypes in the feature space during the finetuning stage. With an adversarial min-max margin way (defined as Eq. 8), during forward propagation, the feature prototypes disturb to minimize the margin which is implemented by re-sampling the support mask. During backward propagation, the feature prototypes move to maximum the margin which is driven by the fine-tuning loss (Eq. 7). With multiple forward-backward propagation iterations, the samples span a feature sub-space for each object class.

Fig. 6 shows the detection result of the baseline method and the proposed CME approach. With the max-margin loss, the few-shot detector can reduce the false detection results because the margin between each class is increased which benefit the discrimination of the classifier. However, the max-margin is not conductive to the feature reconstruction with the raise of missing object. By margin-equilibrium, our approach balanced the contradiction between classification and representation. It shows that CME can precisely detect more objects with fewer false positives.

4.4. Performance Comparison

Pascal VOC In Table 5, we compare CME with the one-stage few-shot detectors including LSTD [1], Meta YOLO [7], and MetaDet [27], which are based on the YOLO detector. The proposed CME detector demonstrates great advantages over the compared detectors. Specifically, for Novel Set 1, CME respectively achieves 0.7%(17.8% vs. 17.1%) on 1-shot setting, 7.0%(26.1% vs. 19.1%) on 2-shot setting, 2.6%(31.5% vs. 28.9%) on 3-shot setting, 9.8%(44.8% vs. 35.0%) on 5-shot setting. The average improvement is 3.8%, which is s significant margin for the challenging task. The average performance improvements are respectively 0.7% for novel set 3.

We also compare the proposed approach with two-stage detectors including MetaDet [27], Meta RCNN [30], TFA [24] Viewpoint Estimation [29], and MPSR [28], which are based on the Faster-RCNN framework. One can see that in most settings CME outperforms the compared

Table 6. Performance comparison on the MS COCO dataset.

Shots	Method	AP	AP50	AP75	APS	APM	APL	AR1	AR10	AR100	ARS	ARM	ARL
10	LSTD [1]	3.2	8.1	2.1	0.9	2.0	6.5	7.8	10.4	10.4	1.1	5.6	19.6
	Meta YOLO [7]	5.6	12.3	4.6	0.9	3.5	10.5	10.1	14.3	14.4	1.5	8.4	28.2
	MetaDet [27]	7.1	14.6	6.1	1.0	4.1	12.2	11.9	15.1	15.5	1.7	9.7	30.1
	Meta R-CNN [30]	8.7	19.1	6.6	2.3	7.7	14.0	12.6	17.8	17.9	**7.8**	15.6	27.2
	TFA w/cos [24]	10.0	-	9.3	-	-	-	-	-	-	-	-	-
	Viewpoint [29]	12.5	**27.3**	9.8	2.5	13.8	19.9	**20.0**	**25.5**	**25.7**	7.5	**27.6**	38.9
	MPSR [28]	9.8	17.9	9.7	3.3	9.2	16.1	15.7	21.2	21.2	4.6	19.6	34.3
	CME (Ours)	**15.1**	24.6	**16.4**	**4.6**	**16.6**	**26.0**	16.3	22.6	22.8	6.6	24.7	**39.7**
30	LSTD [1]	6.7	15.8	5.1	0.4	2.9	12.3	10.9	14.3	14.3	0.9	7.1	27.0
	Meta YOLO [7]	9.1	19.0	7.6	0.8	4.9	16.8	13.2	17.7	17.8	1.5	10.4	33.5
	MetaDet [27]	11.3	21.7	8.1	1.1	6.2	17.3	14.5	18.9	19.2	1.8	11.1	34.4
	Meta R-CNN [30]	12.4	25.3	10.8	2.8	11.6	19.0	15.0	21.4	21.7	**8.6**	20.0	32.1
	TFA w/cos [24]	13.7	-	13.4	-	-	-	-	-	-	-	-	-
	Viewpoint [29]	14.7	**30.6**	12.2	3.2	15.2	23.8	**22.0**	**28.2**	**28.4**	8.3	**30.3**	42.1
	MPSR [28]	14.1	25.4	14.2	4.0	12.9	23.0	17.7	24.2	24.3	5.5	21.0	39.3
	CME (Ours)	**16.9**	28.0	**17.8**	**4.6**	**18.0**	**29.2**	17.5	23.8	24.0	6.0	24.6	**42.5**

Meta Yolo

Meta Yolo with Max-margin

Our Method

Figure 6. Comparison of detection results of the baseline method and the proposed CME approach. Red boxes indicate false detection results, green boxes indicate true detection results and blue boxes indicate missed objects.

detectors. For novel set 1, CME respectively outperforms by 5%(47.5% vs. 42.5%) on 2-shot setting, 6%(58.2% vs. 52.2%) on 5-shot setting. The average improvement researches 1.2%. The average performance improvement for novel set 2 is 1.5% and 0.3% for novel set 3.

MS COCO Compared with Pascal Voc, MS COCO has more object categories and images, which imply that the margin equilibrium may benefit for much richer feature representation. Thereby, our approach achieves more significant relative improvement on MS COCO as shown in Table 6. For the 10-shot setting, CME improves AP upon the baseline method MPSR by 5.3% and for the 30-shot setting, it improves 2.8%.

5. Conclusion

We proposed a class margin equilibrium (CME) approach to optimize both feature space partition and novel class representation for few-shot object detection. During base training, CME preserves adequate margin space for novel classes by a simple-yet-effective class margin loss. During finetuning, CME pursues margin equilibrium by disturbing the instance features of novel classes in an adversarial min-max fashion. Extensive experiments validated the effectiveness of CME for alleviating the constriction of feature representation and classification in few-shot settings. As a plug-and-play module, CME improved both one-stage and two-stage few-shot detectors, in striking contrast to the state-of-the-arts. As a general method for feature representation learning and class margin optimization, CME provides a fresh insight for few-shot learning problems.

Acknowledgement. This work was supported by Natural Science Foundation of China (NSFC) under Grant 61836012 and 61771447, the Strategic Priority Research-Program of Chinese Academy of Sciences under Grant No. XDA27000000, CAAI-Huawei MindSpore Open Fund and MindSpore deep learning computing framework at https://www.mindspore.cn

References

[1] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A low-shot transfer detector for object detection. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 2836–2843, 2018. 7, 8

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017. 2

[3] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448. IEEE Computer Society, 2015. 1, 2

[4] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *IEEE ICCV*, pages 8460–8469, 2019. 2

[5] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE ICCV*, pages 3037–3046, 2017. 2

[6] Muhammad Abdullah Jamal and GUo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *IEEE ICCV*, pages 111719–111727, 2019. 2

[7] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *IEEE ICCV*, pages 8419–8428, 2019. 1, 2, 3, 5, 6, 7, 8

[8] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *IEEE CVPR*, pages 12573–12581, 2020. 2

[9] X. Li, F. Pu, R. Yang, R. Gui, and X. Xu. AMN: Attention Metric Network for One-Shot Remote Sensing Image Scene Classification. *Remote Sensing*, 12(24), 2020. 2

[10] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 2

[11] 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 *ECCV*, 2020. 2, 4

[12] Binghao Liu, Yao Ding, Jianbin Jiao, Ji Xiangyang, and Qixiang Ye. Anti-aliasing semantic reconstruction for few-shot semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[13] B. Liu, J. Jiao, and Q. Ye. Harmonic feature activation for few-shot semantic segmentation. *IEEE Transactions on Image Processing*, 30:3142–3153, 2021. 2

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, volume 9905, pages 21–37, 2016. 2

[15] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[16] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE CVPR*, pages 779–788, 2016. 1, 2

[17] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *IEEE CVPR*, pages 6517–6525, 2017. 1, 2, 3

[18] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 1, 2

[19] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE CVPR*, pages 1199–1208, 2018. 2

[20] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *IEEE ICCV*, pages 6372–6381, 2019. 1

[21] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016. 2

[22] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *IEEE CVPR*, pages 2199–2208, 2019. 1

[23] Fang Wan, Pengxu Wei, Zhenjun Han, Jianbin Jiao, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 41(10):2395–2409, 2019. 1

[24] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E. Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *CoRR*, abs/2003.06957, 2020. 2, 7, 8

[25] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *IEEE CVPR*, pages 7278–7286, 2018. 2

[26] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, pages 616–634, 2016. 2

[27] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *IEEE ICCV*, pages 9924–9933, 2019. 2, 7, 8

[28] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, pages 456–472, 2020. 2, 3, 5, 7, 8

[29] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. 7, 8

[30] 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 *ECCV*, pages 9576–9585. IEEE, 2019. 2, 7, 8

[31] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, volume 12353, pages 763–778, 2020. 2

[32] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE CVPR*, pages 12200–12210, 2020. 2

[33] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *NeurIPS*, pages 147–155, 2019. 1

[34] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *IEEE ICCV*, pages 1859–1868, 2017. 1