σ-Adaptive Decoupled Prototype for Few-Shot Object Detection

Jinhao Du[1*], Shan Zhang[2*], Qiang Chen[1], Haifeng Le[3],
Yanpeng Sun[4], Yao Ni[2], Jian Wang[1], Bin He[1], Jingdong Wang[1†]
[1]Baidu VIS, [2]Australian National University
[3]Beijing Union University, [4]Nanjing University of Science And Technology
[†]wangjingdong@baidu.com, [*]{dujinhao02@gmail.com, shan.zhang@anu.edu.au}

Abstract

Meta-learning-based few-shot detectors use one K-average-pooled prototype (averaging along K-shot dimension) in both Region Proposal Network (RPN) and Detection head (DH) for query detection. Such plain operation would harm the FSOD performance in two aspects: 1) the poor quality of the prototype, and 2) the equivocal guidance due to the contradictions between RPN and DH. In this paper, we look closely into those critical issues and propose the σ-Adaptive Decoupled Prototype (σ-ADP) as a solution. To generate the high-quality prototype, we prioritize salient representations and deemphasize trivial variations by accessing both angle distance and magnitude dispersion (σ) across K-support samples. To provide precise information for the query image, the prototype is decoupled into task-specific ones, which provide tailored guidance for 'where to look' and 'what to look for', respectively.

Beyond that, we find our σ-ADP can gradually strengthen the generalization power of encoding network during meta-training. So it can robustly deal with intra-class variations and a simple K- average pooling is enough to generate a high-quality prototype at meta-testing. We provide theoretical analysis to support its rationality. Extensive experiments on Pascal VOC, MS-COCO and FSOD datasets demonstrate that the proposed method achieves new state-of-the-art performance. Notably, our method surpasses the baseline model by a large margin – up to around 5.0% AP_{50} and 8.0% AP_{75} on novel classes.

1. Introduction

In recent years, object detectors based on deep learning have achieved impressive performance [36, 37, 14] due to a large amount of human-annotated data. However, humans can observe novel objects with limited instances. Thus, few-shot object detection (FSOD) comes to rescue.

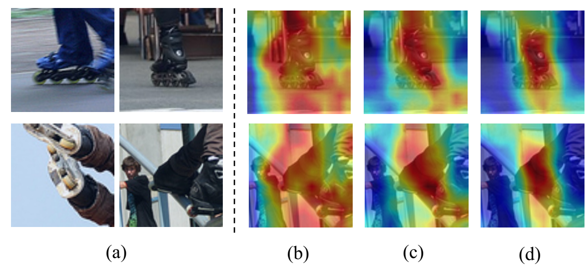


Figure 1: The visualization of attention maps for a novel class of 'walking shoe'. The attention maps are generated by common practice (average-pooling along K-shot dimension) *vs*. weighted pooling (weights deriving from either angle distance or a combination with magnitude deviation), depicted as (b), (c) and (d), respectively. Compared to other operations, σ-ADP (d) is able to active salient regions and suppress trivial features among K support samples, resulting in the robust class representation. We have only shown the two most representative samples for comparison, as the activation map is the same for K samples.

In general, there are two main categories of FSOD approaches : fine-tuning and meta-learning based methods. The fine-tuning approaches [46, 41, 48, 50], without considering the class-level representations, may produce negative transfer when the differences among categories are obvious. Other meta-learning-based methods [18, 19, 49, 8, 54, 56, 55] are designed to acquire class-level meta-knowledge and improve model generalization to novel classes through feature re-weighting. Currently, FSOD meta-detectors use episodic training with inputs of K-support images and a query image. A class-level prototype, generated from K-support images, re-weights the query image and guides the learner for final detection results. Therefore, two main factors directly affect FSOD performance: **1)** *the quality of prototype* and **2)** *the precision of guidance information*.

For the first one, most meta-learning-based methods employ some form of class prototypes (globally semantic-rich or locally spatially-aware) from a set of support samples.

[*]Equal contribution. Author ordering determined by coin flip.
[†]Corresponding author.

For example, methods [8, 54, 56, 29] form vectorial prototypes via global average-pooled features, bilinearly-pooled second-order representations, kernelized descriptors and condition-coupled information respectively. Other works [15, 55, 53] delve into the spatially-aware prototypes. They treat different support samples equally (averaging along K-shot dimension). Such plain operation struggles to capture the salient regions, overwhelming by the non-target objects [1] and the intra-class variations, as shown in Figure 1 (b).

Generally, assigning weights for K-support features based on the angle distance (measured by cosine similarity) helps to reduce the intra-class variance [46], *e.g.*, the cosine similarity between per sample to their average-pooled points. This way, the refined prototype can target the relevant objects, while discarding the most irrelevant regions across K-shot support images. We observe that accounting for cosine similarity alone is insufficient(Figure 1(c)). As a vector is represented by its angle and magnitude (or length), it makes sense to re-evaluate K support features based on the magnitude deviation. And this deviation can be captured by σ, which is a statistical measurement for measuring the dispersion of a set of points.

In short, we first propose a novel σ-Adaptive Prototype (σ-AP) to provide a high-quality class-level representation. Specifically, the σ is power normalized [22] to properly update the cosine-similarity-refined prototype, enhancing the significance of descriptors that are similar to intrinsic representations. The activations produced by our method highlight salient features across support samples, leading to improved class-level representation, as shown in Figure 1 (d).

For the second one, as analysed by the DeFRCN [33], there are potential contradictions between the Region Proposal Network (RPN) and the Detection Head (DH), which may lead to reduced FOSD power. DeFRCN, a fine-turning based method, alleviates conflicts through a gradient decoupled layer. For our meta-learning-based detector, we decouple the prototype into task-specific ones in the spirit of divide-and-conquer. The task-agnostic prototype is divided, and each one plays a specific role in conquering the *where to look* and *what to look for*. This allows our prototype learning to purposefully target inconsistent goals, resulting in precise guidance information.

Beyond satisfying those two requirements, we find our model gradually allows the encoding network (EN) to focus on generic features and factor out outliers across a set of support samples during the training stage. So, the generalization power of EN is strengthened and we can directly utilize the basic average-pooled prototype at the inference stage. We theoretically analyse prioritizing the samples with small σ can speed up the process of prototype learning. And the EN's generalization power is strengthened by raising the lower bound of the optimal prototype. Extensive experiments demonstrate that our model achieves state-of-the-art results, especially on the FSOD dataset without meta fine-tuning on novel classes, which conforms to our model's generalization ability.

In summary, we propose the σ-Adaptive Decoupled Prototype, which includes (i) a novel σ-Adaptive Prototype for robust class-level representations, and (ii) the decoupled task-specific prototypes to provide precise guidance for query detection. We call our approach σ-ADP and its resultant network σ-ADP Net.

2. Related work

Below, we describe popular object detection and few-shot learning algorithms followed by a short discussion on few-shot object detection.

Object Detection. A classical problem of object detection in Computer Vision (CV) performs localization of bounding boxes of objects and recognition of their classes. Historically, object detection relied on sliding windows and hand-crafted features [5, 12, 45]. Deep learning approaches include one-stage detectors which directly regress images to bounding box annotations [35, 36, 26, 28]. Two-stage detectors, inspired by R-CNN [37], generate class-agnostic region proposals which are then classified into class concepts by another network [37, 14, 25, 21]. Two-stage approaches can filter unrelated locations by the Region Proposal Network and outperform one-stage methods [39]. Object detectors are trained on large-scale datasets and do not scale well to novel classes in the low-sample training regime. Few-shot Learning (FSL) described below is better at adaptation to novel classes.

Few-shot Learning. FSL has been heavily explored in CV, with the prominent older shallow approaches [1, 9, 23] and recent convolutional neural network (CNN) based approaches [20, 44, 40, 10, 42, 52]. FSL approaches can be divided into metric learning and meta-learning approaches. The aim of FSL with the underlying metric-learning mechanism [20, 38, 42] is to capture the similarity between training images sufficiently enough to provide good generalization during testing with novel classes. Koch *et al*. [20] employ Siamese networks for one-shot image classification. Prototypical Networks [40] learns a model that computes distances between a datapoint and prototype representations of each class. Meta-learning approaches [11, 13] contain two optimization loops, with the outer loop finding a meta-initialization, from which the inner loop can efficiently learn new tasks. Ravi and Larochelle [34] propose an LSTM-based meta-learner that is trained to attain a quick convergence on new tasks. These classification methods do not scale to detection that requires object localization and recognition.

Few-shot Object Detection. FSOD is an emerging less ex-

[1]The regions are not part of the support object.

plored problem than few-shot classification. Recent methods can be categorized into fine-tuning and meta-learning based models. Firstly, the fine-tuning-based frameworks [3, 46, 50, 33] learn to transfer knowledge from base categories to novel categories via fine-tuning. A Low-Shot Transfer Detector (LSTD) [3] leverages a rich source domain to construct a target domain detector with few training samples. TFA [46] shows that only fine-tuning the last layers of detection head on novel classes can significantly improve the FSOD performance. NP-RepMet [50] introduces a negative- and positive- representative learning framework via triplet losses that bootstrap the classifier. On the other hand, meta-learning-based methods [49, 8, 54, 56] learn a class-agnostic detector by performing an exemplar search at the instance level given K support images. Those approaches can generalize better to novel classes. Recently, FSOD-ARPN [8], PNSD [54] and KFSOD [56] focus on the prototype generation paradigm, that is, they generate a vectorial prototype using different strategies or multiple high-order features. Then support prototypes and query feature maps are matched by channel-wise attention. Other approaches [15, 55, 53] delve into the spatially-aware prototype by K-shot average pooling in the process of prototype learning. But this basic prototype is intra-class biased due to the photometric and geometric variations across a set of support images. This would affect the feature re-weighting with the query image and thus significantly harms the performance of FSOD. We thus revalue K support features based on the reliable weights which are adjusted for both angle similarity and the magnitude deviation σ. Besides, previous meta-learning-based methods utilize the task-agnostic prototype and fail to handle the contradictions between Region Proposal Network and Detection Head. Our method draws on the spirit of divide-and-conquer while proposing task-specific prototypes.

3. Problem Setting

The FSOD operates on L-way K-shot episodes which are formed by sampling a query image containing multiple objects, and K support crops per each of L sampled classes. Specifically, we have a base dataset $\mathcal{D}_b$ containing abundant examples of base classes $\mathcal{C}_b$, and a novel dataset $\mathcal{D}_n$ comprising only a handful of examples of novel classes $\mathcal{C}_n$. The two sets of classes do not overlap, *ie*., $C_b \cap C_n = \emptyset$. Formally, $\mathcal{D}_b = \{(x, y)|y = \{(c_i, b_i)\}, c_i \in \mathcal{C}_b\}$, $\mathcal{D}_n = \{(x, y)|y = \{(c_i, b_i)\}, c_i \in \mathcal{C}_n\}$, where $x \in \mathcal{I}$ is an input image, and $y \in \mathcal{Y}$ is the corresponding annotation; c_i and b_i are the class label and bounding box coordinates of i^{th} image of $\mathcal{I}$, respectively. The goal is to detect objects in the query image for novel classes using few-shot support crops.

4. The Proposed Approach

In this section, we first introduce the architecture of our σ-ADP Net and then elaborate on σ-Adaptive Prototype and Decoupled Task-specific Prototypes. Finally, we provide a brief discussion about rationality.

4.1. Overview

FSOD relies on limited support information to detect objects of novel classes, and there are two important aspects that determine its performance: 1) the quality of the prototype, and 2) the precision of the guidance information for the query detection. These two factors motivate our designs of σ-AP and decoupled task-specific prototypes.

As a plug-and-play module, we implement σ-ADP in two architectures [8, 17] to demonstrate that the prototypes generated by our method are both spatially-aware and semantic-rich. Generally, both architectures consist of an Encoding Network (EN), Support-Query Aggregation (S-QA), Region Proposal Network (RPN) and detection head (DH), but the S-QA and DH designs differ.

The overall architecture of our σ-ADP Net is illustrated in Figure 2. Specifically, given a set of K support crops $\{\boldsymbol{X}_k\}_{k\in\mathcal{I}_k}$ ($\mathcal{I}_k$ stands for the index set of K-shot) and a query image $\boldsymbol{X}^*$ per episode, we use the EN (*e.g.*, ResNet-101) with shared weights to extract feature map $\boldsymbol{\Phi} \in \mathbb{R}^{C\times N}$ per image (of $N = W \times H$ spatial size and C channel dimension) from query and support images. Then, taking as the inputs $\{\boldsymbol{\Phi}_k\}_{k\in\mathcal{I}_k}$ and their K-shot average-pooled feature $\bar{\boldsymbol{\Phi}}$, σ-ADP aims to build the task-specific prototypes. They are individually applied in the subsequent two units: 1) S-QA which matches the prototype with query features to activate co-existing features, passed into the traditional RPN [37] to generate region proposals, and 2) DH with the inputs of proposal-prototype pairs to learn localization and classification for the query image.

4.2. σ-Adaptive Prototype

Motivations: 1) Extracting discriminative and salient features can help create high-quality representations. The classic K-shot average pooling is detrimental to the prototype's quality which is mainly affected by the intra-class variations and non-target objects. While cosine similarity can reduce intra-class variances by measuring angle distances, it may not be sufficient as it does not take into account modulus changes. In order to produce high-quality prototypes, our aim is to exploit the underlying variability by additionally capturing magnitude deviations within the same class. 2) For the dynamic aggregation of support features in a spatially-aware way, FCT [16] uses the transformer [43] architecture for dynamic support feature aggregation, while DAnA [4] generates weights with stacked FC layers. These methods introduce more parameters, which can be detri-

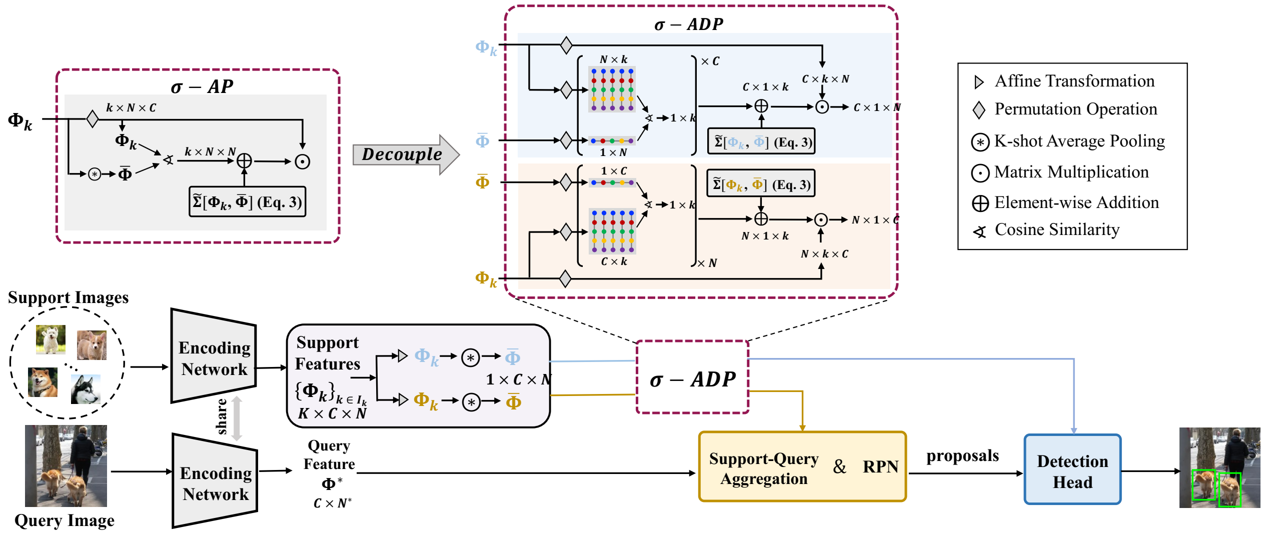


Figure 2: The framework of σ-ADP Net. Compared to the baseline, the model of σ- Adaptive Decoupled Prototype is inserted into the pipeline to provide the task-specific prototypes, performing 'where to look' and 'what to look for' in RPN and Detection Head (as depicted by boxes in yellow and blue for clear reference), respectively. We first employ the affine transformation for feature decoupling, and then capture distance of angle & magnitude, either spatially-wise or channel-wise, for space decoupling. During meta-testing, we remove the σ-ADP model. Best viewed in color and zoomed in.

mental to FSOD by increasing model complexity, thereby reducing network generalization.

In short, we first intend to revalue the representations per sample by measuring both angle and magnitude distances. This filtering process will remove the variability and ensure a robust prototype. In addition, the proposed model should be parameterless. We propose a conceptually simple but practically powerful paradigm in prototype learning.

Given a set of support features $\{\mathbf{\Phi}_k\}_{k\in\mathcal{I}_k}$ and K-shot average-pooled prototype $\bar{\mathbf{\Phi}}$, we transform them into matrices with size of $\{\mathbf{\Phi}_k\}_{k\in\mathcal{I}_k} \in \mathbb{R}^{K\times N\times C}$ and $\{\bar{\mathbf{\Phi}}\} \in \mathbb{R}^{1\times N\times C}$, where K, N, C represent the number of support images, the number of pixels and the channel dimension, respectively.

First, the cosine similarity between the prototype and support samples is formulated, as follows:

$$\tilde{\Gamma}(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) = \frac{(\bar{\mathbf{\Phi}}) \bullet (\mathbf{\Phi}_k)^T}{\left\|\bar{\mathbf{\Phi}}\right\|_2 \bullet \left\|\mathbf{\Phi}_k\right\|_2}, \tag{1}$$

where '$\bullet$' indicate matrix multiplication, and the size of Γ is $\mathbb{R}^{K\times N\times N}$.

Herein, the magnitude dispersion of K support samples is measured by the standard deviation σ_k per shot sample, defined as:

$$\boldsymbol{\sigma}_k(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) = \sqrt{\frac{1}{C}\sum_{i}^{C}(\phi_i^k - \bar{\phi}_i^T)^2}, \tag{2}$$

where lowercase symbols ϕ_i^k and $\bar{\phi}_i$ denote vectors, *e.g.*, $\mathbf{\Phi}_k \equiv \{\phi_i^k \in \mathbb{R}^{N\times 1}\}_{i\in\mathcal{I}_C}$ and $\bar{\mathbf{\Phi}} \equiv \{\bar{\phi}_i \in \mathbb{R}^{N\times 1}\}_{i\in\mathcal{I}_C}$. For brevity, we define $\mathbf{\Sigma} \equiv \{\boldsymbol{\sigma}_k\}_{k\in\mathcal{I}_k} \in \mathbb{R}^{K\times N\times N}$.

In order to positively affect the angle similarity, the Σ^{-1} should be used. However, since Σ is in the denominators, the gradient may sometimes explode at the beginning of training. To avoid this issue, we turn to the Spectral Power Normalization, a so-called SigmE PN function [22] which transforms the inputs into the range $[0, 1]$. Then, we use one minus power normalization results in practice, defined by:

$$\tilde{\Sigma} = 1 - \mathcal{G}_{\text{SigmE}}(\mathbf{\Sigma}; \eta) = \frac{2}{e^{\eta\mathbf{\Sigma}} + 1}, \tag{3}$$

where $1 \leq \eta \approx N$ depicts the number of features in[22], while it plays a different role in our method. η controls the large dispersion features to be filtered out, leading to better adaptation. Refer to the §5 and Supplementary Material §C for further analyses.

We combine the above steps to form the following formulation:

$$\Gamma(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) = \tilde{\Gamma}(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) + \tilde{\Sigma}(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k). \tag{4}$$

Then, K support features are re-weighted, as follows:

$$\Gamma(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) \bullet \mathbf{\Phi}_k. \tag{5}$$

The final robust prototype $\bar{\mathbf{\Phi}}'$ is K- summation across re-evaluated support features.

4.3. Decoupled Task-specific Prototypes

Inspirations: Our pipeline is based on a two-stage architecture called Faster R-CNN, which comprises a Region Proposal Network (RPN) for generating query proposals and a

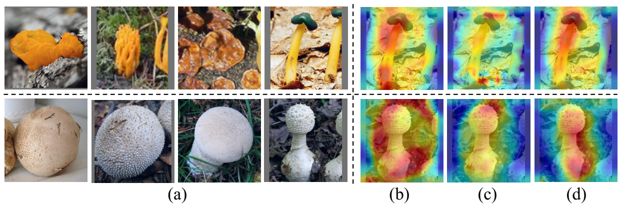


Figure 3: The attention maps in (b), (c), and (d) are generated by different variants trained using either angle distance, magnitude deviation, or both.

Detection Head (DH) for performing classification and localization. Both these sub-networks are guided by the support prototype. There is an inconsistency among these sub-networks [33]. To address this issue, we design spatially-wise and channel-wise prototypes, one for RPN of 'where to look', and the other for DH of 'what to look for'. This is important because an entangled task-agnostic prototype may provide imprecise guidance for individual tasks, as it needs to balance the different needs of both tasks.

When computing $\Gamma(\cdot)$ in Eq.4, the relation map (RM) would be entangled in both spatial and channel dimensions, *ie*., $\bar{\mathbf{\Phi}}(1 \times N \times C)$ and $\mathbf{\Phi}_{\boldsymbol{k}}(K \times N \times C)$ are measured along the second and third dimensions. The size of RM is $K \times N \times N$. We intend to divide $\Gamma(\cdot)$ to $\Gamma^{\ddagger}(\cdot)$ and $\Gamma^{\dagger}(\cdot)$, each with the size of $N \times 1 \times K$ and $C \times 1 \times K$ for capturing the similarity& discrepancy along spatial and channel modes. The two types of RM are obtained by analogy with the same operations as in $\Gamma(\cdot)$, but different input dimensions via the operation of permutation. Specifically, we begin by using two 1×1 convolutional kernels to map K-support features to different representations (feature decoupling), both with weights of $C \times C$, similar to the affine transformation layer [33]. Such outputs are then permuted: for measuring spatially-wise similarity& discrepancy (space decoupling), $\text{Conv}_1(\mathbf{\Phi}_k)$ is rearranged to the size of $N \times K \times C$, while for channel-wise relationships, the size of permuted $\text{Conv}_2(\mathbf{\Phi}_k)$ is $C \times K \times N$. The corresponding prototypes are, of course, K-average pooled across those transformed support features, respectively. The above processes are marked as '$\text{Perm}(\text{Conv}_1(\mathbf{\Phi}_k))$' and '$\text{Perm}(\text{Conv}_2(\mathbf{\Phi}_k))$', which replace the inputs of Eq. 4. The final spatially-wise $\bar{\mathbf{\Phi}}^{\ddagger}$ and channel-wise $\bar{\mathbf{\Phi}}^{\dagger}$ are obtained by weighted summation where the weights are from $\Gamma^{\ddagger}(\cdot)$ and $\Gamma^{\dagger}(\cdot)$, respectively:

$$\bar{\mathbf{\Phi}}^{\ddagger} = \Gamma^{\ddagger}(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) \bullet \mathbf{\Phi}_k, \quad \bar{\mathbf{\Phi}}^{\dagger} = \Gamma^{\dagger}(\bar{\mathbf{\Phi}}, \mathbf{\Phi}_k) \bullet \mathbf{\Phi}_k. \tag{6}$$

4.4. Discussion

Readers may understandably ask about the rationale behind the form of Eq. 4. We provide an initial overview of this design and discuss its crucial factor by providing qualitative results based on the attention map of the prototype. We also give a theoretical analysis of σ-Adaptive Prototype and its impact on EN's generalization ability.

'Refine once' and 'Refine twice' perform similarly. One common method is to refine the basically K-averaged pooled prototype in a step-by-step process. First, the prototype is updated by aggregating K weighted features based on their cosine similarity to the mean. Then, the K features are re-evaluated again based on their dispersion around the first refined prototype for final one. In short, the basic prototype is refined twice. As a serial 'Refine twice' is cumbersome, we try to use a parallel structure by computing cosine similarity and σ between support samples and the basic prototype in parallel, then resulted in σ-adapted cosine similarity for weighting K features. The final prototype is obtained by refining the basic prototype at once instead of one by one ('Refine once'). We examine two designs from a theoretical standpoint, as provided in the Supplementary Material §B, supported by the empirical evidence in §5. And we can safely make the ***1**st **observation*** that 'Refine once' and 'Refine twice' perform similarly.

A residual link is crucial for ensuring 'Refine once' works properly. We perform training where the σ-ADP uses only the magnitude deviation (σ) during meta-training. The resulting attention map is shown in the top row of Figure 3(c), where the prototype features are represented by a few trivial variations. This prototype cannot precisely reweight query features for detection task, resulting in lower FSOD results. If there are large differences in appearance or photometry, it can be hard to capture common features based on the sample' dispersion. However, cosine similarity measures the angle distances between two sets of vectors and is not affected by the magnitude of the vectors being compared (the top row of Figure 3 (b)). Therefore, it is better to first use angle distance following a residual sample' dispersion. We obtain the ***2**nd **observation*** that a residual link is crucial for ensuring 'Refine once' works properly.

These two ***observations*** explain why two statistical representations are combined by a residual link (element addition) in Eq 4. Even in the worst case, σ wouldn't impede prototype learning; instead, it would enhance it (Figure 3 (d)).

Theoretical Analysis: For meta-learning-based detectors, the high quality class-level prototype ($\bar{\mathbf{\Phi}}$) should be robust enough to represent K support samples ($\mathbf{\Phi}_{\boldsymbol{k}}$). In other words, an optimal prototype should be similar to all samples within the same class, as indicated by maximum expectation of cosine similarity among them, and also across L classes ($\bar{\mathbf{\Phi}}_{\boldsymbol{L}} \equiv \{\bar{\mathbf{\Phi}}_{\boldsymbol{l}}\}_{l \in \mathcal{I}_L}$). This process is formulated as:

$$\max \mathbb{E}_{\bar{\mathbf{\Phi}}_{\boldsymbol{l}}}[\mathbb{E}_{\mathbf{\Phi}_{\boldsymbol{k}}}[Cos(\bar{\mathbf{\Phi}}_{\boldsymbol{l}}, \mathbf{\Phi}_{\boldsymbol{k}})]], \tag{7}$$

Proposition 1 *Approximating the optimal prototype is equivalent to minimizing the variance* $\mathbb{D}[\phi_i^k]$, $\mathbf{\Phi}_k \equiv$

$\{\phi_i^k\}_{i \in \mathcal{I}_C}$:

$$\mathbb{E}_{\bar{\Phi}_l}[\mathbb{E}_{\Phi_k}[Cos(\bar{\Phi}, \Phi_k)]] \geq \frac{\sum_{i=1}^{C} \mathbb{E}[\phi_i^k]^2}{\sum_{i=1}^{C} \mathbb{D}[\phi_i^k] + \sum_{i=1}^{C} \mathbb{E}[\phi_i^k]^2} \tag{8}$$

We give the proof in the Supplementary Material §A. The Eq.8 presents a learning task with adjustable speed, where the speed gradually increases as the variance becomes smaller. Therefore, if the meta-learner is trained with low-dispersion points, it can speed up the process of approaching the expected prototypes. Thus, we emphasize low-dispersion descriptors and deemphasize trivial variations by explicitly using σ in prototype learning.

Besides, the intra-class variations can significantly affect the generalization ability of CNNs, resulting in lower generalization performance on new classes [32, 51]. Thus, EN should also take advantage of the robust prototypes. In Eq.8, when the objective of prototype learning is achieved, the lower boundary will be raised. This learning process can help EN to deal robustly with outliers and strengthen features that spread less from the salient features. This is empirically supported by Figure 4b, which shows the deviation between the estimated prototype and the optimal one.

5. Experiments

Datasets and Evaluation. We evaluate our method on three benchmark datasets: PASCAL VOC 2007/12 [7], MS COCO [27] and FSOD [8]. For PASCAL VOC 2007/12, we use three random splits, each consisting of 20 categories that are randomly divided into base/novel classes at a ratio of 15/5. Few-shot learning is performed on each novel category with $K \in \{1, 2, 3, 5, 10\}$ objects sampled from the combination of VOC07 and VOC12 train/val set. Following previous works [18, 8, 54, 24], we evaluate the detector performance using the mean Average Precision with intersection over union (IoU) with the threshold of 0.5 (AP_{50}). For MS COCO [27], we adopt 20 categories that overlap with PASCAL VOC as novel categories and utilize the remaining 60 categories as base classes, as done in [49]. During the few-shot fine-tuning step, we choose $K \in \{10, 30\}$ annotated samples for each category and the standard COCO-style AP metric is employed to evaluate our method. For the FSOD dataset [8], we divide its 1000 categories into base/novel classes at a ratio of 800/200, and report the detection performance using the commonly used metrics AP_{50} and AP_{75}.

Implementation Details. The proposed model is trained with a genetic detection loss that has been used by existing methods [8, 56, 54], *ie.*, $\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{reg}$, where $\mathcal{L}_{rpn}$ aims to refine the region proposals generated from RPN, $\mathcal{L}_{cls}$ is the binary cross-entropy loss for the box classifier, and $\mathcal{L}_{reg}$ is a smooth ℓ_1 loss for the bounding-box regression. The model is trained with the SGD optimizer (momentum 0.9, weight decay 1e-4, batch size 4) on 4 NVIDIA V100 GPUs. We follow the same training/fine-tuning iterations as [8, 54, 56]. Images are resized to have a shorter edge of 600 pixels and a maximum longer edge of 1000 pixels. Each support image is cropped based on ground-truth boxes, bilinearly interpolated and padded to 320×320 pixels. We keep all hyper-parameters the same across all three datesets, unless specified otherwise.

Training Framework. To transfer knowledge from base categories to novel categories, we adopt the typical two-step training scheme:

(1) Meta-learning on base classes. We leverage episode-based training on base classes with an encoder network (ResNet-101) pre-trained on ImageNet [6]. Each episode includes a single query image and K randomly sampled support instances per class. During the meta-testing step, we generalize the class-agnostic model to novel classes by simply calculating their class prototypes.

(2) Fine-tuning on novel classes (optimal step). For PASCAL VOC and MS COCO datasets, we fine-tune our model on novel classes using the same training strategy as meta-learning on base classes. For the FSOD dataset, we do not use fine-tuning.

5.1. Main Results

5.1.1 Comparisons with Main Baselines

We first show the effectiveness of our method by comparing it with two baselines. As shown in Table 1 and Table 2, Ours+FSOD and Ours+DCNet achieved significant improvements of ∼4.8% and 6.1%, respectively, over the main baselines on PASCAL VOC benchmark. Even in extremely low-shot scenarios, σ-ADP still benefits FSOD performance as it allows for self-refinement within samples. Moreover, σ-ADP consistently improves the performance of both baselines on the more challenging FSOD and COCO benchmarks, as demonstrated in Table 3a and Table 3b.

5.1.2 Comparisons with the State-of-the-Art

PASCAL VOC 2007/12. We compare our method to FADI [2], QSAM[24], FSOD^{up} [47], FSCE [41], TFA [46], MetaDet [49], NP-RepMet [50], MPSR[48], FSOD [8], PSND [54], KFSOD [56], MGHL [53], and DCNet [17]. Table 1 shows the AP_{50} of the novel classes on the three data splits with K training shots. σ-ADP outperforms TENET (the second-best) by a remarkable margin of ∼ 1.82–4.4%, highlighting the effectiveness of our designs. Moreover, Table 2 provides detailed class-wise results of each novel/base category under the (class split 1, 3-shot setting), which show that the proposed σ-ADP significantly boosts the detection performance for the base categories (69.5% and 72.2%) compared with the second-best method [56], indicating our method has better generalization ability and can alleviate the catastrophic forgetting issue when transferring the base knowledge to a novel domain.

Table 1: Comparison of different methods in terms of AP_{50} (%) under 3 different splits for 5 novel categories with K shots. **RED/BLUE** denote the best/the second best.* represents average results over multiple runs. '–': No reported results.

Method / Shots	Venue	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW	ICCV 2019	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	ICCV 2019	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
TFA w/ fc	ICML 2020	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/ cos	ICML 2020	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
Xiao *et al.*	ECCV 2020	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
MPSR	ECCV 2020	41.7	42.5	51.4	55.2	61.8	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
FSOD	CVPR 2020	37.8	43.6	51.6	56.5	58.6	22.5	30.6	40.7	43.1	47.6	31.0	37.9	43.7	51.3	49.8
PNSD	ACCV2020	38.4	44.1	51.3	57.2	59.1	25.2	33.2	43.3	45.4	49.3	32.8	38.7	45.6	52.9	52.4
NP-RepMet	NeurIPS20	37.8	39.2	31.7	37.3	49.4	41.6	41.3	43.4	47.4	49.1	33.3	35.6	39.8	41.5	44.8
SRR-FSD	CVPR 2021	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
FSCE	CVPR 2021	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
$FSOD^{up}$	ICCV 2021	43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5
MGHL	CVPR 2021	48.6	51.1	52.0	53.7	54.3	**41.6**	**45.4**	45.8	46.3	48.0	46.1	**51.7**	**52.6**	54.1	55.0
CME	CVPR 2021	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
FADI	NeurIPS21	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	**55.0**	**59.6**
QSAM	WACV2022	31.1	35.7	39.2	50.7	59.4	22.9	28.4	32.1	35.4	42.7	24.3	29.1	35.0	50.0	53.6
KFSOD	CVPR2022	44.6	–	54.4	60.9	**65.8**	37.8	–	43.1	48.1	**50.4**	34.8	–	44.1	52.7	53.9
DeFRCN	ICCV2021	**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
Ours+DCNet		**52.3**	**55.5**	**63.1**	**65.9**	**66.7**	**42.7**	**45.8**	**48.7**	**54.8**	**56.3**	**47.8**	**51.8**	**56.8**	**60.3**	**62.4**
TFA w/ cos*	ICML 2020	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
TIP*	CVPR 2021	27.7	36.5	43.3	50.2	59.6	22.7	**30.1**	33.8	**40.9**	46.9	21.7	30.6	38.1	**44.5**	50.9
DCNet*	CVPR 2021	33.9	**37.4**	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	**34.9**	**39.7**	42.6	50.7
TENET*	ECCV 2022	**34.1**	–	**50.2**	**52.0**	**60.7**	**24.0**	–	**40.5**	40.2	**47.4**	**33.3**	–	38.9	43.7	**51.4**
Ours+DCNet*		**35.9**	**40.3**	**49.8**	**56.8**	**65.1**	**25.6**	**30.3**	**41.7**	**41.8**	**50.3**	**33.9**	**35.6**	**43.5**	**47.1**	**55.9**

Table 2: Comparison with SOTA on the PASCAL VOC 2007 testing set for novel and base categories (class split 1, 3-shot protocol) in terms of AP_{50} (%). **RED/BLUE** denote the best/the second best. The FSÖD indicates that the results are reproduced by us.

Method	Venue	Novel: bird	bus	cow	mbike	sofa	mean	Base: aero	bike	boat	bottle	car	cat	chair	table	dog	horse	person	plant	sheep	train	tv	mean
FSRW	ICCV2019	26.1	19.1	40.7	20.4	27.1	26.7	73.6	73.1	56.7	41.6	76.1	78.7	42.6	**66.8**	72.0	77.7	68.5	42.0	57.1	74.7	**70.7**	64.8
Meta R-CNN	ICCV2019	30.1	44.6	50.8	38.8	10.7	35.0	67.6	70.5	59.8	50.0	75.7	81.4	44.9	57.7	76.3	74.9	76.9	34.7	58.7	74.7	67.8	64.8
FSÖD	CVPR2020	35.8	61.2	57.6	60.2	44.7	51.9	68.0	73.3	58.6	54.1	**79.5**	**81.7**	48.4	62.9	79.1	**83.6**	76.3	36.6	65.2	**75.4**	62.3	67.0
MPSR	ECCV2020	35.1	60.6	56.6	**61.5**	43.4	51.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	67.8
NP-RepMet	NeurIPS2020	12.9	60.5	39.9	43.1	**52.2**	41.7	**79.8**	**82.5**	**66.9**	**73.8**	71.6	57.6	**52.9**	64.1	49.6	70.7	71.8	**58.7**	74.2	55.0	69.5	66.6
KFSOD	CVPR2022	**39.8**	**61.9**	**59.6**	58.3	45.7	**53.1**	78.1	73.4	60.3	58.2	79.4	81.0	52.2	61.1	**83.3**	74.9	**80.8**	41.3	**76.6**	71.7	70.5	**69.5**
Ours+FSOD		40.2	65.3	65.7	**68.4**	**50.3**	57.9	69.7	75.8	63.7	56.8	82.5	**85.8**	**52.6**	**64.9**	81.1	**86.6**	78.6	38.9	68.4	**78.6**	65.7	69.8
Ours+DCNet		**49.5**	**68.8**	**70.5**	60.5	43.6	**63.1**	**79.2**	**78.3**	**65.1**	**59.4**	**85.9**	82.7	45.7	64.3	**88.7**	78.6	**82.3**	**45.1**	**75.0**	77.7	**75.1**	**72.2**

FSOD. Table 3a presents a comparison of σ-ADP with FSOD [8], PNSD [54], KFSOD [56], TENET [55] and LSTD (FRN) [3] under 5-shot protocol. Our method achieves the SOTA results of 36.9% AP_{50} and 32.8% AP_{75} on this setting, surpassing all other methods. Note that all methods in the table are directly applied to detect unseen categories without fine-tuning, except for LSTD (FRN), which transfers base knowledge to the novel domain.

MS COCO. We further 3b compare σ-ADP with FADI [2], FSCE [41], TFA [46], Meta R-CNN [49], KFSOD [56] and DeFRCN [33] on the MS COCO minival set (20 novel categories, 10/30-shot protocol), a more challenging dataset with more complex scenarios and larger data size. our model consistently outperforms recent SOTAs on the 10/30 shot protocol, achieving approximately 1.8% mAP improvement over the best method in the 10-shot regime. Notably, even without using advanced techniques such as gradient decoupled layers, our method still outperforms DeFRCN [33] in the 30-shot setting.

5.2. Ablation Analysis

In this section, we conduct a comprehensive ablation analysis to investigate the impact of each key component in our σ-ADP. To achieve this, we build our σ-ADP upon the strong baseline FSOD [8]. We report the ablation results on the 5-shot protocol for each novel category on the FSOD dataset without any further fine-tuning.

Prototype generation strategies. Herein, we investigate the effectiveness of our σ-ADP module to generate a good prototype. We compare three different strategies for

Table 3: Evaluation on the FSOD testset (3a) and MS COCO minival set (3b). **RED/BLUE** denote the best/the second best. '–' denotes results not provided.

Method	Venue	AP_{50} 5	AP_{75} 5
LSTD (FRN)	AAAI18	23.0	12.9
FSOD	CVPR20	27.5	19.4
PNSD	ACCV20	29.8	22.6
QSAM	WACV22	30.7	25.9
KFOSD	CVPR22	33.4	29.6
TENET	ECCV22	**35.4**	**31.6**
Ours+FSOD		32.7	27.3
Ours+DCNet		**36.9**	**32.8**

(a)

Method	Venue	mAP 10	mAP 30	AP_{75} 10	AP_{75} 30
MPSR	ECCV20	9.8	14.1	9.7	14.2
PNSD	ACCV20	10.3	15.5	10.7	14.8
SRR-FSD	CVPR21	11.3	14.7	9.8	13.5
FSCE	CVPR21	11.9	16.4	10.5	16.2
FADI	NeurIPS21	12.2	16.1	11.9	15.8
KFSOD	CVPR22	**18.5**	–	**18.7**	–
DCNet	CVPR21	12.8	18.6	11.2	**17.5**
DeFRCN	ICCV21	18.5	**22.6**	–	–
Ours+FSOD		16.2	20.8	16.9	18.0
Ours+DCNet		**20.3**	**22.8**	**20.8**	**23.6**

(b)

Table 4: Evaluation on FSOD testset (5-shot protocol on novel classes) for the effectiveness of decoupled task-specific prototypes ($\bar{\mathbf{\Phi}}^{\ddagger}$ and $\bar{\mathbf{\Phi}}^{\dagger}$) *vs.* an entangled task-agnostic prototype ($\bar{\mathbf{\Phi}}'$).

	$\bar{\mathbf{\Phi}}'$	$\bar{\mathbf{\Phi}}^{\ddagger}$	$\bar{\mathbf{\Phi}}^{\dagger}$	Aff. Trans.	Novel(5-shot) mAP	AP_{50}	AP_{75}
a	✓				28.0	29.8	25.7
		✓			27.3	29.1	25.4
			✓		27.6	29.4	25.9
		✓	✓		29.3	31.8	26.8
b	✓			✓	28.1	30.5	26.4
		✓	✓	✓	**29.9**	**32.7**	**27.3**

producing prototypes: training with (1) the basically K-average pooled prototype ('K-avg.'), (2) K-weighted summation based on cosine similarity only ('K-cos.'), and (3) K-weighted summation based on standard deviation only ('K-σ'). We also ablate the process for basic prototype refinement (Re. once and Re. twice). Table 5a presents the ablation results. The results show that 'K-cos.' and 'K-σ' are unable to provide high-quality class-level prototypes as both of them lead to a decrease in object detection performance. Also, the two processes of prototype refinement perform similarly and provide up to 5.0% mAP/7.0% AP_{75} gain over 'K-avg.' in novel classes (5-shot protocol).

Impact of task-specific prototypes. Designing task-specific prototypes for mismatched tasks in RPN and DH should help the detection of novel objects. To evaluate our claim, we conduct ablations and present the results in Table 4. We use (a) an affine transformation layer for feature decoupling and (b) the metrics of similarity&deviation along spatial and channel for space decoupling. In the setting of (a), the superior performance of the last row demonstrates space decoupling is more effective than using either prototype alone. Furthermore, under the 5-shot regime on novel categories, we observed a drop in detection performance by $\sim$1% mAP and $\sim$2% AP_{50} without using space decoupling. In the setting of (b), our ablations confirmed that feature decoupling brings slight benefits to object detection in mismatched tasks in RPN and DH. However, the most significant impact on FSOD performance was observed when both decoupling tactics were used together, increasing performance from 28.0%to 29.9% mAP.

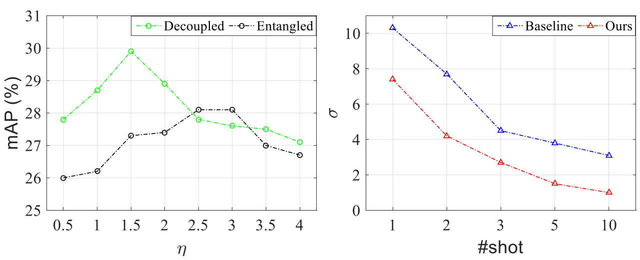
Table 5: Results on FSOD testset (5-shot protocol on novel classes) for applying different strategies of prototype generation (5a). Effect on the generalization ability of encoding network (EN) in (5b).

Prototype Generation	Novel(5-shot) mAP	AP_{50}	AP_{75}
K-avg.	23.1	27.5	19.4
K-cos.	24.2	28.3	20.7
K-σ	24.8	28.7	21.3
Re. once	**28.1**	30.5	**26.4**
Re. twice	28.0	**30.7**	26.1

(a)

Training	Testing	Novel(5-shot) mAP	AP_{50}	AP_{75}
K-avg.	K-avg.	23.1	27.5	19.4
σ-AP	K-avg.	28.1	30.5	26.4
	σ-AP	**28.7**	**30.6**	**26.9**

(b)



(a) (b)

Figure 4: Impact of varying the value of η in SigmE PN for both entangled task-agnostic and decoupled task-specific prototypes (4a). Comparison, w.r.t. the standard deviation (σ) of the estimated prototype from the expected one, is reported in (4b), where an expected prototype is a cluster center of all training support examples in the same class.

Effects on the generalization of EN. During meta-training and meta-testing, we apply hybrid strategies to examine how our method impacts the power of the encoding network (EN). Table 5b summarizes the results for novel classes. We have the following key observations: 1) prototype learning with k-average pooled entity leads to low generalization power, and 2) considering the similarity&deviation of features to their prototype during training improves EN's generalization, and 3) with generalized EN, the simple K-average pooling operation is sufficient for providing the high-quality prototypes and precise guidance for query detection during meta-testing. Moreover, as shown in Figure 4b, the prototype generated by σ-ADP is closer to the expected value (the cluster center of the support examples in the same class) than K-average pooling (our baseline). Note we report the average distance for all novel classes.

Hyper-parameter Analysis. We examine the influence of η in SigmE PN for both entangled and decoupled prototypes, which is responsible for filtering out the large dispersion features. This leads to a better combination with cosine similarity. We first vary η from 0.5 to 4. Figure 4a shows that our model performance is stable when $\eta \geq 2.5$. We further observe that 2.5/1.5 gives the best performance for entangled/decoupled prototypes.

Inference results for multi-support crops. For inference only, we randomly crop several patches based on the ground truth bounding box of the support object for prototype estimation. Table 6 shows the ablation on different numbers of crops (*vs*. the baseline FSOD). Increasing the number of features in the prototype estimation improves the performance, but exceeding 3 crops leads to performance degradation. The presence of valuable/positive support samples is crucial for achieving good results, as emphasized by the method [30] (Sec 4.1, 'Bigger is not necessarily better'). Our model exhibits robustness in handling noisy support crops (#crop>1) *vs*. baseline.

Table 6: Inference results for multi-support crops on FSOD testset (5-shot, averaged mAP/AP_{75} over 5 runs).

Method	1-crop	2-crop	3-crop	4-crop	5-crop
FSOD	23.1 / 19.4	23.1 / 19.5	22.5 / 18.1	22.0 / 17.2	21.5 / 16.7
Ours+FSOD	**29.9 / 27.3**	**30.1 / 27.9**	**30.3 / 28.1**	**29.3 / 27.2**	**28.8 / 26.7**

Generalization on transformer-extractor. We adopt the transformer-based extractor Swin-B, pretrained on ImageNet-22K, as the encoding network(EN) for PASCAL VOC, MS COCO, and FSOD datasets (AP_{50}%), following the architecture of FSOD. The results show in Table 7, where the superscript represents the window size. Importantly, σ-ADP consistently outperforms the baseline in transformer-based EN by 4.8–6.2%, showcasing its excellent compatibility. Refer to the Supplementary Material §F for details on applying σ-ADP to FCT [16].

Table 7: Results on PASCAL VOC, MS COCO and FSOD testset w.r.t. the generalization on Swin-B, measured by mAP/AP_{50}.

Method	ResNet-101			Swin-B^7			Swin-B^{12}		
	5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)	5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)	5-shot (VOC)	5-shot (FSOD)	10-shot (COCO)
FSOD	56.8	27.5	18.6	57.1	28.6	19.1	56.3	28.3	18.4
Ours+FSOD	**62.0**	**32.7**	**23.9**	**63.1**	**33.4**	**25.3**	**62.7**	**33.2**	**24.9**

6. Conclusions

We have proposed σ-ADP to generate high-quality prototypes tailored to each task in RPN and DH for FSOD. To factor out underlying intra-class variations within support samples, we consider both amplitude and angle distance of K-shot samples from the mean. Thus, we leverage a simple standard deviation formula (σ) to adaptively update the cosine similarity. Our theoretical analysis verifies that prioritizing the low-dispersion samples can speed up the process of prototype learning, and also benefit the EN's generalization power. Finally, we decouple the prototype into task-specific ones to conquer the contradicted tasks in RPN and DH. Extensive experiments on three few-shot benchmarks demonstrate its effectiveness.

References

[1] Evgeniy Bart and Shimon Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. *CVPR*, pages 672–679, 2005. 2

[2] Yuhang Cao, Jiaqi Wang, Ying Jin, Tong Wu, Kai Chen, Ziwei Liu, and Dahua Lin. Few-shot object detection via association and discrimination. *NeurIPS*, 34, 2021. 6, 7

[3] 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. 3, 7

[4] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, and Winston H. Hsu. Dual-awareness attention for few-shot object detection. *CoRR*, abs/2102.12152, 2021. 3

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893. IEEE Computer Society, 2005. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 6

[7] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 6

[8] Qi Fan, Wei Zhuo, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. *CoRR*, abs/1908.01998, 2019. 1, 2, 3, 6, 7

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006. 2

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70, pages 1126–1135. PMLR, 2017. 2

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70, pages 1126–1135. PMLR, 2017. 2

[12] David A. Forsyth. Object detection with discriminatively trained part-based models. *IEEE Computer*, 47(2):6–7, 2014. 2

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375. IEEE Computer Society, 2018. 2

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

[15] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster R-CNN: towards accurate few-shot object detection with attentive feature alignment. *CoRR*, abs/2104.07719, 2021. 2, 3

[16] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *CVPR*, pages 5321–5330, 2022. 3, 9, 14

[17] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, pages 10185–10194, 2021. 3, 6

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

[19] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *CVPR*, pages 5197–5206. CVF / IEEE, 2019. 1

[20] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 2

[21] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *CVPR*, pages 845–853. IEEE Computer Society, 2016. 2

[22] Piotr Koniusz and Hongguang Zhang. Power normalizations in fine-grained image, few-shot image and graph classification. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2, 4

[23] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. *CogSci*, 2011. 2

[24] Hojun Lee, Myunggi Lee, and Nojun Kwak. Few-shot object detection by attending to per-sample-prototype. In *WACV*, pages 1101–1110. IEEE, 2022. 6, 13

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

[26] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007. IEEE Computer Society, 2017. 2

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, volume 8693, pages 740–755. Springer, 2014. 6

[28] Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, volume 11215, pages 404–419. Springer, 2018. 2

[29] Xiaonan Lu, Wenhui Diao, Yongqiang Mao, Junxi Li, Peijin Wang, Xian Sun, and Kun Fu. Breaking immutable: Information-coupled prototype elaboration for few-shot object detection. *arXiv preprint arXiv:2211.14782*, 2022. 2

[30] Xu Luo, Hao Wu, Ji Zhang, Lianli Gao, Jing Xu, and Jingkuan Song. A closer look at few-shot classification again. *arXiv preprint arXiv:2301.12246*, 2023. 9

[31] Sebastian Nowozin. Optimal decisions from probabilistic models: the intersection-over-union case. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 548–555, 2014. 12

[32] Rafal Pilarczyk and Wladyslaw Skarbek. On intra-class variance for deep learning of classifiers, 2019. 6

[33] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *CVPR*, pages 8681–8690, 2021. 2, 3, 5, 7

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

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

[36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 1, 2

[37] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 2, 3

[38] Pranav Shyam, Shubham Gupta, and Ambedkar Dukkipati. Attentive recurrent comparators. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3173–3181. PMLR, 2017. 2

[39] Bharat Singh, Mahyar Najibi, and Larry S. Davis. SNIPER: efficient multi-scale training. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NeurIPS*, pages 9333–9343, 2018. 2

[40] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NeurIPS*, pages 4077–4087, 2017. 2

[41] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: few-shot object detection via contrastive proposal encoding. *CoRR*, abs/2103.05950, 2021. 1, 6, 7

[42] 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 *CVPR*, pages 1199–1208. IEEE Computer Society, 2018. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 3

[44] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NeurIPS*, pages 3630–3638, 2016. 2

[45] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518. IEEE Computer Society, 2001. 2

[46] Xin Wang, Thomas E. Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML 2020*, volume 119, pages 9919–9928. PMLR, 2020. 1, 2, 3, 6, 7

[47] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *ICCV*, pages 9567–9576, October 2021. 6

[48] 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*, volume 12361, pages 456–472. Springer, 2020. 1, 6

[49] 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 *ICCV*, pages 9576–9585. IEEE, 2019. 1, 3, 6, 7

[50] Yukuan Yang, Fangyun Wei, Miaojing Shi, and Guoqi Li. Restoring negative information in few-shot object detection. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020. 1, 3, 6

[51] Somayeh Danafar Zahra Sadeghi. Intra-class variations and their impact on transfer learning. In *Journal of Computational and Applied Mathematics*, 2021. 6

[52] Hongguang Zhang and Piotr Koniusz. Power normalizing second-order similarity network for few-shot learning. In *WACV*, pages 1185–1193. IEEE, 2019. 2

[53] Lu Zhang, Shuigeng Zhou, Jihong Guan, and Ji Zhang. Accurate few-shot object detection with support-query mutual guidance and hybrid loss. In *CVPR*, pages 14424–14432, June 2021. 2, 3, 6

[54] Shan Zhang, Dawei Luo, Lei Wang, and Piotr Koniusz. Few-shot object detection by second-order pooling. In *ACCV*, 2020. 1, 2, 3, 6, 7

[55] Shan Zhang, Murray Naila, Lei Wang, and Piotr Koniusz. Time-reversed diffusion tensor transformer: A new tenet of few-shot object detection. In *ECCV*, 2022. 1, 2, 3, 7, 13

[56] Shan Zhang, Lei Wang, Murray Naila, and Piotr Koniusz. Kernelized few-shot object detection with efficient integral aggregation. In *CVPR*, 2022. 1, 2, 3, 6, 7