σ-Adaptive Decoupled Prototype for Few-Shot Object Detection

Jinhao Du\(^1\)*, Shan Zhang\(^2\)*, Qiang Chen\(^1\), Haifeng Le\(^3\), Yanpeng Sun\(^1\), 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.

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.
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 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 features. 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 $\sigma$, which is a statistical measurement for measuring the dispersion of a set of points.

In short, we first propose a novel $\sigma$-Adaptive Prototype ($\sigma$-AP) to provide a high-quality class-level representation. Specifically, the $\sigma$ 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-tuning 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 analyze prioritizing the samples with small $\sigma$ 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 $\sigma$-Adaptive Decoupled Prototype, which includes (i) a novel $\sigma$-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 $\sigma$-ADP and its resultant network $\sigma$-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-
explored 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 $\sigma$. 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 $D_b$, containing abundant examples of base classes $C_b$, and a novel dataset $D_n$ comprising only a handful of examples of novel classes $C_n$. The two sets of classes do not overlap, i.e., $C_b \cap C_n = \emptyset$. Formally, $D_b = \{(x, y) | y = \{(c_i, b_i)\} , c_i \in C_b\}, D_n = \{(x, y) | y = \{(c_i, b_i)\} , c_i \in C_n\}$, where $x \in I$ 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 $I$, 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 $\sigma$-ADP Net and then elaborate on $\sigma$-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 $\sigma$-AP and decoupled task-specific prototypes.

As a plug-and-play module, we implement $\sigma$-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 $\sigma$-ADP Net is illustrated in Figure 2. Specifically, given a set of $K$ support crops $\{X_k\}_{k \in I_k}$ ($I_k$ stands for the index set of $K$-shot) and a query image $X^*$ per episode, we use the EN (e.g., ResNet-101) with shared weights to extract feature map $\Phi \in \mathbb{R}^{C \times N}$ per image (of $N = W \times H$ spatial size and $C$ channel dimension) from query and support images. Then, taking as the inputs $\{\Phi_k\}_{k \in I_k}$ and their $K$-shot average-pooled feature $\bar{\Phi}$, $\sigma$-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. $\sigma$-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-
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 \( \{\Phi_k\}_{k \in I_k} \) and \( K \)-shot average-pooled prototype \( \bar{\Phi} \), we transform them into matrices with size of \( \{\Phi_k\}_{k \in I_k} \in \mathbb{R}^{K \times N \times C} \) and \( \{\Phi\} \in \mathbb{R}^{1 \times N \times C} \), 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:

\[
\Gamma(\Phi, \Phi_k) = \frac{\Phi \bullet \Phi_k}{\|\Phi\|_2 \|\Phi_k\|_2},
\]

where \( \bullet \) indicate matrix multiplication, and the size of \( \Gamma \) is \( \mathbb{R}^{K \times N \times N} \).

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

\[
\sigma_k(\Phi, \Phi_k) = \sqrt{\frac{1}{C} \sum_{i} (\phi_i^k - \bar{\phi}_i)^2},
\]

where lowercase symbols \( \phi_i^k \) and \( \bar{\phi}_i \) denote vectors, e.g.,

\[\Phi_k = \{\phi_i^k \in \mathbb{R}^{N \times 1}\}_{i \in I_G}, \quad \bar{\Phi} = \{\bar{\phi}_i \in \mathbb{R}^{N \times 1}\}_{i \in I_G}. \]

For brevity, we define \( \Sigma = \{\sigma_k\}_{k \in I_k} \in \mathbb{R}^{K \times N \times N} \).

In order to positively affect the angle similarity, the \( \Sigma^{-1} \) should be used. However, since \( \Sigma \) 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:

\[
\tilde{\Sigma} = 1 - G_{\text{SigmE}}(\Sigma; \eta) = \frac{2}{e^{\eta \Sigma} + 1},
\]

where \( 1 \leq \eta \approx N \) depicts the number of features in\([22]\), while it plays a different role in our method. \( \eta \) 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:

\[
\Gamma(\Phi, \Phi_k) = \Gamma(\Phi, \Phi_k) + \tilde{\Sigma}(\Phi, \Phi_k).
\]

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

\[
\Gamma(\Phi, \Phi_k) \bullet \Phi_k.
\]

The final robust prototype \( \Phi' \) 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
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, i.e., $\Phi(1 \times N \times C)$ and $\Phi^k(K \times N \times C)$ 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 $\Gamma^1(\cdot)$ and $\Gamma^2(\cdot)$, 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 $1 \times 1$ 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), $\Phi_1(\Phi_k)$ is rearranged to the size of $N \times K \times C$, while for channel-wise relationships, the size of permuted $\Phi_2(\Phi_k)$ 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 ‘Perm($\Phi_1(\Phi_k)$)’ and ‘Perm($\Phi_2(\Phi_k)$)’, which replace the inputs of Eq. 4. The final spatially-wise $\Phi^{1\dagger}$ and channel-wise $\Phi^{2\dagger}$ are obtained by weighted summation where the weights are from $\Gamma^1(\cdot)$ and $\Gamma^2(\cdot)$, respectively:

$$\Phi^{1\dagger} = \Gamma^1(\Phi, \Phi_k) \cdot \Phi_k, \quad \Phi^{2\dagger} = \Gamma^2(\Phi, \Phi_k) \cdot \Phi_k.$$  

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 $\sigma$-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 $\sigma$ between support samples and the basic prototype in parallel, then resulted in $\sigma$-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 1st 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 $\sigma$-ADP uses only the magnitude deviation ($\sigma$) 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 re-weight 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 2nd 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, $\sigma$ 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 ($\Phi$) should be robust enough to represent $K$ support samples ($\Phi_k$). 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 ($\Phi_L \equiv \{\Phi_l\}_{l \in \mathcal{L}}$). This process is formulated as:

$$\max \mathbb{E}_{\Phi} [\mathbb{E}_{\Phi_k} [\cos(\Phi_l, \Phi_k)]]$$

**Proposition 1** Approximating the optimal prototype is equivalent to minimizing the variance $\mathbb{D}[\phi^{k\dagger}]$. $\Phi_k \equiv \Phi$
\{\phi_i^k\}_{i \in I_C} : \ 
E_{\Phi} [E_{\Phi_h} [\text{Cos}(\Phi, \Phi_h)]] \geq \frac{\sum_{i=1}^{C} E[\phi_i^k]^2}{\sum_{i=1}^{C} \mathbb{D}[\phi_i^k] + \sum_{i=1}^{C} 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 \(\sigma\) 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 (AP50). 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], i.e., \(L_{det} = L_{rpn} + L_{cls} + L_{reg}\), where \(L_{rpn}\) aims to refine the region proposals generated from RPN, \(L_{cls}\) is the binary cross-entropy loss for the box classifier, and \(L_{reg}\) is a smooth \(\ell_1\) 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 x 320 pixels. We keep all hyper-parameters the same across all three datasets, 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, \(\sigma\)-ADP still benefits FSOD performance as it allows for self-refinement within samples. Moreover, \(\sigma\)-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\(^{\text{up}}\) [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 AP50 of the novel classes on the three data splits with \(K\) training shots. \(\sigma\)-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 \(\sigma\)-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 AP50 (%) 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.

<table>
<thead>
<tr>
<th>Method / Shots</th>
<th>Venue</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSRW</td>
<td>ICCV 2019</td>
<td>25.3</td>
<td>36.4</td>
<td>42.1</td>
<td>47.9</td>
<td>52.8</td>
<td>18.3</td>
<td>27.5</td>
<td>30.9</td>
<td>40.8</td>
<td>43.1</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
<td>45.6</td>
</tr>
<tr>
<td>MetaDet</td>
<td>ICCV 2019</td>
<td>37.8</td>
<td>39.2</td>
<td>31.7</td>
<td>37.3</td>
<td>49.4</td>
<td>41.6</td>
<td>41.3</td>
<td>43.2</td>
<td>46.5</td>
<td>45.8</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
<td>46.3</td>
</tr>
<tr>
<td>TFA w/ fc</td>
<td>ICML 2020</td>
<td>44.7</td>
<td>43.8</td>
<td>51.4</td>
<td>61.9</td>
<td>63.4</td>
<td>27.3</td>
<td>29.5</td>
<td>43.5</td>
<td>44.2</td>
<td>50.2</td>
<td>37.2</td>
<td>41.9</td>
<td>47.5</td>
<td>54.6</td>
<td>58.5</td>
<td>58.5</td>
<td>58.5</td>
<td>58.5</td>
<td>58.5</td>
<td>58.5</td>
<td>58.5</td>
</tr>
<tr>
<td>TFA w/ cos</td>
<td>ICML 2020</td>
<td>41.5</td>
<td>47.5</td>
<td>50.4</td>
<td>58.2</td>
<td>60.9</td>
<td>27.2</td>
<td>30.2</td>
<td>41.4</td>
<td>42.5</td>
<td>46.8</td>
<td>34.3</td>
<td>39.6</td>
<td>45.1</td>
<td>48.3</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
</tr>
<tr>
<td>Xiao et al.</td>
<td>ECCV 2020</td>
<td>50.3</td>
<td>54.8</td>
<td>54.2</td>
<td>59.3</td>
<td>63.2</td>
<td>30.6</td>
<td>35.0</td>
<td>40.3</td>
<td>42.8</td>
<td>48.0</td>
<td>45.7</td>
<td>49.7</td>
<td>49.1</td>
<td>55.0</td>
<td>59.6</td>
<td>59.6</td>
<td>59.6</td>
<td>59.6</td>
<td>59.6</td>
<td>59.6</td>
<td>59.6</td>
</tr>
<tr>
<td>MPRC</td>
<td>ECCV 2020</td>
<td>31.1</td>
<td>35.7</td>
<td>39.2</td>
<td>50.7</td>
<td>59.4</td>
<td>22.9</td>
<td>28.4</td>
<td>32.1</td>
<td>35.4</td>
<td>42.7</td>
<td>24.3</td>
<td>29.1</td>
<td>35.2</td>
<td>46.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
<td>51.5</td>
</tr>
<tr>
<td>FSOD</td>
<td>CVPR 2020</td>
<td>34.1</td>
<td>–</td>
<td>50.2</td>
<td>52.0</td>
<td>60.7</td>
<td>24.0</td>
<td>–</td>
<td>40.5</td>
<td>40.2</td>
<td>47.4</td>
<td>33.3</td>
<td>–</td>
<td>38.9</td>
<td>43.7</td>
<td>51.4</td>
<td>51.4</td>
<td>51.4</td>
<td>51.4</td>
<td>51.4</td>
<td>51.4</td>
<td>51.4</td>
</tr>
<tr>
<td>Ours+DCNet</td>
<td>ICCV 2021</td>
<td>53.6</td>
<td>57.5</td>
<td>61.5</td>
<td>64.1</td>
<td>60.8</td>
<td>30.1</td>
<td>38.1</td>
<td>47.0</td>
<td>53.3</td>
<td>47.9</td>
<td>48.4</td>
<td>50.9</td>
<td>52.3</td>
<td>54.9</td>
<td>57.4</td>
<td>57.4</td>
<td>57.4</td>
<td>57.4</td>
<td>57.4</td>
<td>57.4</td>
<td>57.4</td>
</tr>
</tbody>
</table>

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% AP50 and 32.8% AP75 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 mini-val set (3b). RED/BLUE denote the best/the second best. ‘-‘ denotes results not provided.

<table>
<thead>
<tr>
<th>Method</th>
<th>Venue</th>
<th>(AP_{50})</th>
<th>(AP_{75})</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTD</td>
<td>AAI18</td>
<td>23.0</td>
<td>12.9</td>
</tr>
<tr>
<td>FSOD</td>
<td>CVPR20</td>
<td>27.5</td>
<td>19.4</td>
</tr>
<tr>
<td>PNSD</td>
<td>ACCV20</td>
<td>29.8</td>
<td>22.6</td>
</tr>
<tr>
<td>QSAM</td>
<td>WACV22</td>
<td>30.7</td>
<td>25.9</td>
</tr>
<tr>
<td>KFOSD</td>
<td>CVPR22</td>
<td>33.4</td>
<td>29.6</td>
</tr>
<tr>
<td>TENET</td>
<td>ECCV22</td>
<td>35.4</td>
<td>31.6</td>
</tr>
</tbody>
</table>

Table 4: Evaluation on FSOD testset (5-shot protocol on novel classes) for the effectiveness of decoupled task-specific prototypes (\(\Phi^1\) and \(\Phi^\dagger\)) vs. an entangled task-agnostic prototype (\(\Phi^\ddagger\)).

<table>
<thead>
<tr>
<th>Aff. Trans.</th>
<th>(\Phi^1)</th>
<th>(\Phi^\dagger)</th>
<th>(\Phi^\ddagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>b</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Prototype Generation</th>
<th>Novel(5-shot)</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(mAP)</td>
<td>(AP_{50})</td>
<td>(AP_{75})</td>
</tr>
<tr>
<td>Keq</td>
<td>23.1</td>
<td>27.5</td>
<td>19.4</td>
</tr>
<tr>
<td>K-s</td>
<td>24.8</td>
<td>28.7</td>
<td>21.3</td>
</tr>
<tr>
<td>Re. once</td>
<td>28.1</td>
<td>30.5</td>
<td>26.4</td>
</tr>
<tr>
<td>Re. twice</td>
<td>28.0</td>
<td>30.7</td>
<td>26.1</td>
</tr>
</tbody>
</table>

Figure 4: Impact of varying the value of \(\eta\) in SigmE PN for both entangled-task-agnostic and decoupled-task-specific prototypes (4a). Comparison, w.r.t. the standard deviation (\(\sigma\)) 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.

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 (‘\(K\)-\(\sigma\)’). 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 ‘\(K\)-\(\sigma\)’ 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% \(mAP\).

**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 \(\sigma\)-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 \(\eta\) 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 \(\eta\) 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/AP50 over 5 runs).

<table>
<thead>
<tr>
<th>Method</th>
<th>1-crop</th>
<th>2-crop</th>
<th>3-crop</th>
<th>4-crop</th>
<th>5-crop</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSOD</td>
<td>23.1 /</td>
<td>19.4 /</td>
<td>22.5 /</td>
<td>22.0 /</td>
<td>21.5 /</td>
</tr>
<tr>
<td>Ours+FSOD</td>
<td>29.9 /</td>
<td>27.3 /</td>
<td>30.1 /</td>
<td>29.3 /</td>
<td>27.2 /</td>
</tr>
<tr>
<td></td>
<td>27.9 /</td>
<td>28.1 /</td>
<td>28.8 /</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 (AP50%), 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/AP50.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FSOD</td>
<td>56.8 / 27.5 / 18.6</td>
<td>57.1 / 28.6 / 19.1</td>
<td>56.3 / 28.3 / 18.4</td>
</tr>
<tr>
<td>Ours+FSOD</td>
<td>62.0 / 32.7 / 23.9</td>
<td>63.1 / 33.4 / 25.3</td>
<td>62.7 / 33.2 / 24.9</td>
</tr>
</tbody>
</table>

**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**


[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


[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
