Towards Few-shot Object Detection through Dual Calibration

Ding Sheng Ong, and Yi Liu, and Jungong Han, *Senior Member, IEEE*

***Abstract*—Object detection is crucial in traffic scenes for accurately identifying multiple objects within complex environments. Traditional systems rely on deep learning models trained on large-scale datasets, but this approach can be expensive and impractical. Few-shot object detection (FSOD) offers a potential solution by addressing limited data availability. However, object detectors trained with FSOD frameworks often generalize poorly on classes with limited samples. Although most existing methods alleviate this problem by calibrating either the feature maps or prediction heads of the object detector, none of them, like this work, have proposed a unified, dual calibration strategy that operates in both the latent feature space and the prediction probability space of the object detector. Specifically, we propose to improve representation precision by reducing the variances of feature vectors using highly adaptive centroids learned from ensembles of training features in the latent space. These centroids are employed to calibrate the features and reveal the underlying structure of the latent feature space. Moreover, we further exploit the association between the query and support features to calibrate inaccurate predictions resulting from overfitting or underfitting when fine-tuned with few training samples and low training iterations. Through visualization, we demonstrate that our method produces more discriminative high-level features, ultimately improving the precision of an object detector's predictions. To validate the effectiveness of our approaches, we conduct comprehensive experiments on well-known benchmarks, including PASCAL VOC and MS-COCO, showing considerable performance gains compared to existing works. The training codes can be found at https://github.com/dingsheng-ong/fsod-dc.**

***Index Terms*—Few-shot learning, object detection, calibration, transfer learning.**

I. Introduction

OBJECT detection [1], [2] is a key computer vision technology that provides robust perception capabilities, particularly in the context of intelligent vehicle applications, such as Advanced Driver Assistance Systems (ADAS) and Automated Driving Systems (ADS). Designing a vision-based system capable of accurately identifying various objects in traffic scenes, such as pedestrians, vehicles, and traffic signs is crucial for the development of modern ADS and ADAS systems. This includes components like traffic sign recognition [3], pedestrian detection [4], obstacle detection [5], and navigation. However, traditional deep learning models may suffer from the vast diversity of objects encountered in complex traffic scenes, as traffic conditions can vary significantly across different geographic locations, influenced by cultural differences, laws, and regulations. This diversity complicates system design, as it is impractical to collect extensive samples covering the wide variety of objects and scenarios encountered. Fortunately, the emerging field of few-shot object detection (FSOD) addresses this challenge by enabling visual systems to detect previously unseen objects with limited samples (e.g., $1-30$ instances).

Although FSOD extends from few-shot learning (FSL), which predominantly focuses on classification tasks, it introduces additional complexities by simultaneously addressing both classification and localization. This challenge is compounded by the presence of multiple base and novel objects within a single image, making it difficult to generate high-precision feature maps and accurate predictions with limited training samples. Most FSOD methods, despite employing diverse learning strategies such as meta-learning [6]–[14] and transfer learning [15]–[22], share common goals of enhancing representations and improving prediction accuracy. These methods often address the challenge of generalization with limited learning samples by leveraging support information to calibrate inaccurate representations or predictions. Typical approaches include adding extra branches and using aggregation modules, such as attention mechanisms [23], for feature map calibration [6], [8], [10], [11], [13], [14], [18], [22], or incorporating support sets as auxiliary information to improve prediction accuracy [17], [20], [21], [24].

As stated above, previous FSOD calibration techniques can be broadly categorized into two groups: representation calibration and prediction calibration. Representation calibration focuses on generating meaningful features from extractors while mitigating overfitting. In contrast, prediction calibration aims to correct inaccuracies in predictions due to undertrained classifiers in transfer learning scenarios. Although these two perspectives are distinct and their contributions largely do not overlap, it is surprising that previous research has not explored integrating both types of calibration into a unified framework. Besides, both approaches also face challenges that limit their effectiveness in FSOD. Representation calibration methods often struggle to generate high-precision feature maps, leading to significant noise when learning the feature space for novel categories. The feature extractor tends to overfit, memorizing specific characteristics, including noise, from the limited samples rather than capturing the generalizable features essential for few-shot categories. Meanwhile,

This work was supported in part by the UKRI AIMLAC CDT under grant EP/S023992/1, and in part by the National Natural Science Foundation of Jiangsu Province under Grant BK20221379, in part by the Changzhou University CNPC-CZU Innovation Alliance under Grant CCIA2023-01, and in part by the Changzhou Leading Innovative Talent Introduction and Cultivation Project under Grant 20221460. (*Corresponding authors: Yi Liu; Jungong Han.*)

Ding Sheng Ong and Jungong Han are with the Department of Computer Science, Aberystwyth University, Penglais, Aberystwyth SY23 3DB, UK.

Yi Liu is with the School of Computer Science and Artificial Intelligence, the Aliyun School of Big Data, and the School of Software, Changzhou University, Changzhou, Jiangsu 213000, China.

Jungong Han is also with the Department of Computer Science from The University of Sheffield, Regent Court, Sheffield S1 4DP, UK.

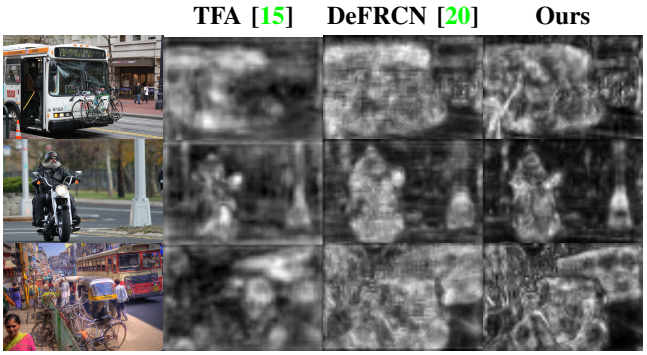


Fig. 1. Comparison of the feature maps generated by the backbone and the performance of the object detectors in the MS-COCO 1-shot setting. Note the variation in noise level and level of detail between the feature maps of previous works and our approach, which incorporates feature calibration.

prediction calibration techniques frequently require additional data or involve complex computations, as seen in studies such as [17], [21].

In response to these challenges, we propose a dual calibration framework that integrates both feature and logit calibration modules into a unified system. This framework aims to enhance FSOD accuracy and robustness by incorporating the strengths of both calibration approaches, which share a common structure involving references or prototypes and calibration algorithms. We also strive to overcome the limitations of existing calibration approaches and improve the overall effectiveness of FSOD models. Specifically, to reduce noise in feature maps, we introduce a lightweight module with learnable centroids. These centroids partition the feature space and adjust feature vectors towards their respective centroids, thereby reducing variance and improving precision. This leads to more discriminative features, as demonstrated in Fig. 1, and results in enhanced performance compared to state-of-the-art methods [15], [20]. Our approach integrates seamlessly into the existing feature extraction network without requiring additional branches or supervision. It also incorporates exponential moving averages in centroid learning to better control update momentum and mitigate overfitting on few-shot samples. Our study further validates the generalizability of the proposed calibration modules to novel categories through comprehensive quantitative (See Section IV-D) and qualitative (See Section V) evaluations.

At the logit level, our objective is to enhance prediction accuracy by addressing suboptimal prediction heads that produce inaccurate probability logits in transfer learning setting. Training the prediction head from scratch with limited data, lower learning rates, and fewer training iterations result in a low-accuracy classifier for novel categories. To tackle this issue while taking inspiration from [25] and [26], we propose a module that calibrates prediction logits by leveraging support-query associations and cosine similarity between features. Additionally, we observe a lack of discriminability in the region of interest (RoI) features extracted from support samples due to similarities across different object categories (Figs. 7 and 8). For instance, vehicles often have very similar visual appearances, and mammals like cats and dogs share common features. Thus, we introduce a mechanism to enhance RoI feature discriminability by computing a linear transformation matrix that has been optimized to maximize the inter-class distance between features from different object categories. Our method offers significant advantages over previous approaches as it does not necessitate additional unlabeled images for fine-tuning, distinguishing it from the work of Kaul *et al.* [21], thereby eliminates the need for extra data acquisition. Additionally, our logit calibration module outperforms existing methods, such as the prototypical calibration block (PCB) [20], by addressing critical issues such as negative probability scores.

In summary, our work pursues to advance FSOD by integrating calibration mechanisms at multiple levels within the object detection framework. Both of our calibration modules employ a similar structure involving references or prototypes and a calibration procedure. At the feature level, we use learnable centroids as references, with the calibration procedure involves shifting feature vectors towards these centroids to enhance precision. Similarly, at the logit level, we generate prototypes from the support set and aggregate logits based on the similarity between region-of-interest (RoI) features and the support queries. Our contributions can be outlined as follows:

(i) We introduce a novel dual calibration strategy that simultaneously calibrates at both the feature and prediction levels. By using a unified structure for both modules, we eliminate potential conflicts and achieve highly effective calibration.

(ii) We present a feature calibration module that enhances feature precision by leveraging learnable centroids to partition the feature space and reduce variance in the feature maps produced by the backbone.

(iii) We develop a logit calibration mechanism that improves prediction accuracy by using cosine similarity to assess the relationship between query and support features, where the discriminability of the support features are further enhanced using a pre-computed transformation matrix.

(iv) We evaluate our proposed approach on the well-established FSOD benchmarks and achieve the state-of-the-art performance.

II. Related Works

A. Object Detection

Object detection is a computer vision problem that has been extensively studied and for which numerous approaches have been proposed. We will thus only describe a subset of the approaches by grouping them into one-stage [27]–[31] and two-stage [32]–[34] detectors. The primary distinction between these two approaches is that the one-stage detector directly predicts class labels and bounding boxes. In contrast, the two-stage detector predicts class-agnostic proposals, then predicts the class labels for each proposal in the second stage. While the conventional approach for localization is via a collection of predefined anchor boxes [27]–[29], [32], [33], there has lately been a subset of methods, notably anchor-free methods [35]–[38], that do not rely on the anchor boxes

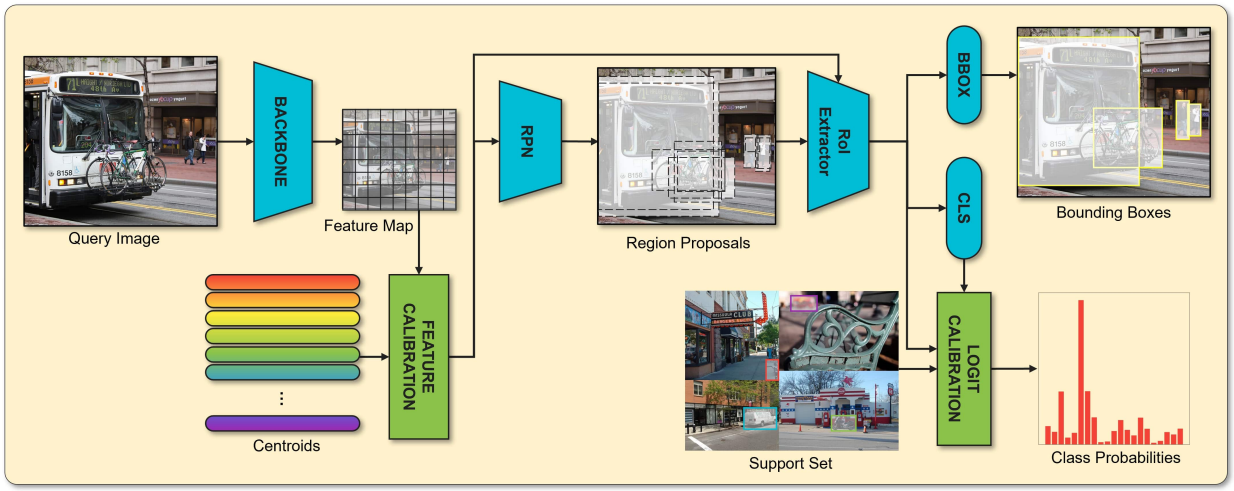


Fig. 2. Overview of the proposed dual calibration framework. Blue components represent the Faster R-CNN components, while green components denote the proposed modules, namely the feature-level calibration and logit-level calibration. Logit calibration is exclusively utilized in the novel fine-tuning phase.

to predict the position. While there are exceptions, such as YOLO [6], [18] and DETR [12], employed to study FSOD, Faster R-CNN remains the preferred base object detector for studying FSOD.

B. Few-shot Learning

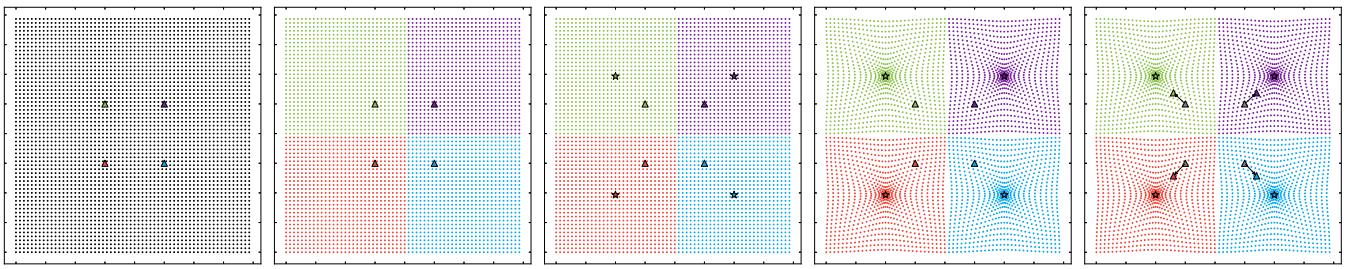
FSL is closely related to FSOD and focuses on recognizing objects from novel classes using only a few examples. In FSL research, classification tasks are commonly used as benchmarks to evaluate new methods. Although many FSL techniques, including meta-learning and metric-learning approaches, have significantly influenced FSOD methodologies, they often fall short when applied directly to object detection. This is due to the increased complexity of FSOD, which involves not only classification but also localization, and the challenge of detecting multiple base and novel objects within a single image, obfuscating the prediction of their structural labels. Meta-learning approaches [39]–[42], such as MAML [43], restructure training into episodic mini-tasks with small training and testing sets, enabling the model to rapidly adapt to new tasks with minimal iterations. In contrast, metric-learning methods [25], [26], [44] focus on learning a generalizable metric to measure image similarity within the base dataset, which is then used to classify novel examples.

C. Few-shot Object Detection

FSOD involves recognizing novel objects from limited samples while simultaneously performing localization and recognition tasks. In recent literature, meta-learning [6]–[14] and transfer-learning [15]–[22] are the most prevalent learning strategies employed to solve FSOD problems. The meta-learning methods closely resemble FSL, in which the FSOD dataset is divided into episodes of few-shots tasks. On the other hand, transfer-learning strategy [15] is proposed as an alternative to the meta-learning approach, where the parameters of an object detector trained on a large base dataset are fine-tuned using a few-shot novel dataset. Regardless of the learning paradigm, most methods introduce different calibration modules that either calibrate the low-quality representation or less accurate predictions. For example, many two-branch works [6], [8], [10]–[12], [14], [18], [22] aim to improve the representation quality by aggregating the support set through mechanisms like attention, concatenation, etc. Additionally, some calibration methods focus on enhancing predictions instead of representations, often leveraging additional data [17], [21] or feature extraction branches [20] to compute alternative predictions based on similarity scores. We observe the performance gains from these works on FSOD precision and agree they are effective in improving FSOD performance. However, to the best of our knowledge, no work has proposed a calibration framework that applies unified structure calibration modules at both representation and prediction levels.

D. Few-shot Segmentation

Few-shot segmentation (FSS) extends beyond FSOD by requiring more detailed and precise labels. Instead of simply identifying objects with bounding boxes, FSS necessitates dense labeling, where each pixel in the image is classified into a specific category. This task involves segmenting the object from the background with a high level of detail, demanding finer granularity in the labeling process. Although FSS and FSOD share similarities, such as the use of prototypes or reference representations to guide predictions (*e.g.* [19], [20] in FSOD, [45], [46] in FSS), and often rely on learning frameworks designed for FSL like meta-learning (*e.g.* [47], [48] in FSS), their implementations differ significantly. For instance, both FSS and FSOD have recently adopted commonality distillation methods to address few-shot learning challenges, with MFDC [49], SD-FSOD [24] being prominent examples in FSOD and PCNet by Lang *et al.* [50] being used in FSS. The core distinction lies in their output requirements: FSOD focuses on coarse localization with bounding boxes, whereas FSS demands fine-grained, pixel-level segmentation. This difference in granularity reflects the varying complexity and precision needed for each task.



(i) $\bullet = \boldsymbol{x}_i$; $\blacktriangle = \boldsymbol{v}_j$ (ii) Eq. (1) (iii) $\bigstar = \boldsymbol{\mu}_j$ (Eq. (2)) (iv) Eq. (3) (v) Eq. (4)

Fig. 3. Proposed feature calibration mechanism in a simulated 2D feature space. (i) Feature vectors, $\boldsymbol{x}_i$ are represented as circles ($\bullet$), learnable centroids, $\boldsymbol{v}_j$ as triangles ($\blacktriangle$). (ii) Feature vectors are partitioned based on similarities to centroids (as described in Equation (1)), each partition visually represented by the color associated with its respective centroid. (iii) New centroids, $\boldsymbol{\mu}_j$ (represented by $\bigstar$) for each partition are computed by averaging the vectors within the respective partition (as per Equation (2)). (iv) Calibration of the feature map by applying proximity-based adjustments (as per Equation (3)) to shift the feature vectors, $\boldsymbol{x}_i$ towards the re-computed centroids, $\boldsymbol{\mu}_j$. (v) Centroids, $\boldsymbol{v}_j$ ($\blacktriangle$) are updated by shifting them towards the re-computed centroids, $\boldsymbol{\mu}_j$ ($\bigstar$), as described in Equation (4).

III. Method

In line with existing research [8], [10], [14]–[21], we have adopted the widely-used Faster R-CNN [33] as the foundation for our object detection model. Our primary objective is to enhance the performance of few-shot object detection by calibrating the outputs of the object detector at two distinct levels: the feature level and the logit level. To achieve this, we have developed two independent modules that operate at these respective levels. Firstly, we introduce the feature level calibration module, which effectively partitions the feature space based on the proximity of feature vectors and subsequently adjusts them towards their corresponding centroids. Additionally, we present the logit level calibration module, where we leverage the support set to calibrate the logits predicted by the RCNN, thereby rectifying any inaccurate predictions. In Fig. 2, we illustrate an overview of our proposed approaches and designs for addressing the FSOD problem. The following sections describe the implementation details of both components in further detail.

Algorithm 1 Feature calibration module.

Require: feature map, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^T \in \mathbb{R}^{M \times d_1}$
Require: centroids, $V = \left\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N : \boldsymbol{v}_j \in \mathbb{R}^{d_1}\right\}$
Require: hyperparameter, α
  for $i = 1, \ldots, M$ **do**
    $a_i \leftarrow \arg\max_{j \in \{1,\ldots,N\}} S_C\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right)$ ▷ assign partitions
  end for
  for $j = 1, \ldots, N$ **do**
    $\boldsymbol{\mu}_j \leftarrow \dfrac{\sum_{i=1}^{M} \boldsymbol{x}_i [\![a_i = j]\!]}{\sum_{i=1}^{M} 1[\![a_i = j]\!]}$
    $\boldsymbol{v}_j \leftarrow (1-\alpha)\, \boldsymbol{v}_j + \alpha \boldsymbol{\mu}_j$ ▷ update centroids
  end for
  for $i = 1, \ldots, M$ **do**
    $\Delta \boldsymbol{x}_i \leftarrow \sum_{j=1}^{N} \left(\boldsymbol{\mu}_j - \boldsymbol{x}_i\right) [\![a_i = j]\!]$
    $\boldsymbol{x'}_i \leftarrow \boldsymbol{x}_i + \exp\left(-\dfrac{\|\Delta \boldsymbol{x}_i\|^2}{d_1}\right) \Delta \boldsymbol{x}_i$ ▷ refine feature
  end for

A. Problem Definition

In our study, we employ identical problem settings from previous works [6], [15]. In particular, the object detection dataset $\mathcal{D}$ comprises the base $\mathcal{D}_{\text{base}}$ and novel $\mathcal{D}_{\text{novel}}$ (a.k.a. support set) datasets: the former has large training samples with associated annotations, and the latter only contains K (*e.g.* $K = 1, 2, 3, 5, 10, 30$) annotated samples. This study aims to learn an object detector that localizes and classifies the objects in the provided samples (a.k.a. query images) from novel categories by exploiting both the large base and scarce novel datasets.

B. Feature Level Calibration

Unlike previous approaches [6], [8], [10], [12], [14], [18], [22] that use the support set to guide the backbone encoding the feature map, our method focuses on discovering the underlying structure in the latent feature space by learning the centroids that partition the feature space based on proximity. The centroids are then re-computed by averaging the feature vectors in the same partition, which are subsequently leveraged to shift the feature vectors toward the centroids. Concretely, the update strategy for the centroids features an exponential moving average, with the update momentum exclusively defined by a hyper-parameter, α. It enables the centroids to adapt to the novel dataset without overfitting since the hyper-parameter allows us to fully regulate the extent to which the centroids deviate from the base set centroids. Furthermore, we produce a higher-precision feature map due to the relocation of the feature vectors closer to the centroid, reducing the variances between the feature vectors. Henceforth, we referred to this procedure as feature calibration.

Given a d_1-dimensional feature map with dimensions $h \times w \times d_1$, wherein h and w represent the feature map's height and width, we seek to generate a refined feature map with the exact dimensions. To be precise, we refer to the flattened feature map as $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^T \in \mathbb{R}^{M \times d_1}$, where M is the total number of pixels ($M = h \times w$) and $V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ is a collection of N centroids. Note that the centroids, $\boldsymbol{v}_j$, and the feature vectors, $\boldsymbol{x}_i$, have the same dimension, d_1 ($\boldsymbol{x}_i \in \mathbb{R}^{d_1}$ and $\boldsymbol{v}_j \in \mathbb{R}^{d_1}$). First, each feature vector, $\boldsymbol{x}_i$, is assigned to its

corresponding partition by comparing it to the centroids, $\boldsymbol{v}_j$,

$$a_i = \arg \max_{j \in \{1,\dots,N\}} S_C\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right) \quad \text{for } i = 1,\dots,M, \tag{1}$$

where $S_C\left(\boldsymbol{u}, \boldsymbol{v}\right) = \dfrac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$ computes the cosine similarity between two vectors. Intuitively, we assign the feature vectors into N partitions according to their similarities to the respective centroids. The purpose of the following step is to recalculate the centroid, $\boldsymbol{\mu}_j$, in the j-th partition using the feature vectors assigned to the same cluster. Specifically, the re-computed centroid, $\boldsymbol{\mu}_j$, is evaluated as the mean of the feature vectors that belong to the same partition,

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{M} \boldsymbol{x}_i [\![a_i = j]\!]}{\sum_{i=1}^{M} 1[\![a_i = j]\!]} \quad \text{for } j = 1,\dots,N, \tag{2}$$

where $1[\![\cdot]\!]$ represents the Iverson bracket, in which the term is evaluated as 1 if the statement written in the bracket is true and 0 if it is false. The step of re-evaluating the centroids is necessary when relocating the feature vectors to reduce the variance of feature vectors within the same partition. Then, the next step would be shifting each feature vector, $\boldsymbol{x}_i$, to its respective centroid, $\boldsymbol{\mu}_j$. The magnitude of shifting is proportional to the distance between the centroids and the feature vectors, with the closer feature vectors being brought closer to the centroids:

$$\begin{aligned} \Delta \boldsymbol{x}_i &= \sum_{j=1}^{N} \left(\boldsymbol{\mu}_j - \boldsymbol{x}_i\right) [\![a_i = j]\!] \quad \text{for } i = 1,\dots,M, \\ \boldsymbol{x'}_i &= \boldsymbol{x}_i + \exp\left(-\frac{\|\Delta \boldsymbol{x}_i\|^2}{d_1}\right) \Delta \boldsymbol{x}_i. \end{aligned} \tag{3}$$

The computed shifting vectors, $\Delta \boldsymbol{x}_i$, indicate the direction of each feature vector relative to its centroid, and the ℓ_2-norm of such a vector provides the Euclidean distance between them. The shifting magnitude increases as the feature vector is positioned closer to its center. Due to the discrete nature of the partition assignment, gradient descent could not be used to learn the centroids, $\boldsymbol{v}_j$. Thus, we could only update the centroids by computing the exponential moving average and adding all the re-computed centroids, $\boldsymbol{\mu}_j$,

$$\boldsymbol{v}_j = (1 - \alpha)\boldsymbol{v}_j + \alpha \boldsymbol{\mu}_j, \tag{4}$$

where α determines the weight of the recent re-computed centroids over the centroids computed from past training samples. In other words, the hyper-parameter, α, may also be interpreted as the update rate of the centroids, with a greater value indicating that the centroids update at a higher rate in each iteration. Accordingly, the centroids, $\boldsymbol{v}_j$, can be considered an ensemble of partition centroids, $\boldsymbol{\mu}_j$, determined from the whole training set. Algorithm 1 provides a summary of the complete implementation of the feature calibration and Fig. 3 illustrates the proposed feature calibration mechanism applied to a simulated 2D feature space.

C. Logit Level Calibration

In general, the weights of class-specific prediction layers are not transferable in a transfer learning setting where the weights learned on the base dataset are fine-tuned on a few-shot novel dataset. Therefore, they are randomly initialized during the fine-tuning process and are usually trained using fewer iterations and a lower learning rate. These variables cause the classifier to overfit or underfit the few-shot training samples, resulting in inaccurate predictions with arbitrary confidence.

1) Logit Calibration: To overcome the problem mentioned above, we design a logit calibration module to refine the logit predicted by the classifier using pairwise cosine similarity between the query and support features to calibrate the confidence level of the prediction during the few-shot fine-tuning stage.

Concretely, given the support set of a novel class, $S_c = \left\{\boldsymbol{f}_1^{(c)}, \dots, \boldsymbol{f}_K^{(c)}\right\}$ for $c \in C_{\text{novel}}$, with K-shot RoI features where each feature is a d_2-dimensional vector ($\boldsymbol{f}_k^{(c)} \in \mathbb{R}^{d_2}$), we first calculate the representation for a category by averaging the RoI features of the same class,

$$\overline{\boldsymbol{f}}^{(c)} = \frac{1}{|S_c|} \sum_{\boldsymbol{f}_k^{(c)} \in S_c} \boldsymbol{f}_k^{(c)}. \tag{5}$$

The cosine similarity score, π_c is then computed by comparing the support RoI feature for each class, $\overline{\boldsymbol{f}}^{(c)}$, to the query RoI feature, $\boldsymbol{f}$,

$$\pi_c = S_C\left(\boldsymbol{f}, \overline{\boldsymbol{f}}^{(c)}\right) = \frac{\boldsymbol{f} \cdot \overline{\boldsymbol{f}}^{(c)}}{\|\boldsymbol{f}\|\|\overline{\boldsymbol{f}}^{(c)}\|}, \tag{6}$$

where $\cdot$ denotes the dot product of two given vectors. To this end, we will have a vector of cosine similarity scores, $\boldsymbol{\pi} = [\pi_1, \dots, \pi_{|C_{\text{novel}}|}]^T$, for each proposal, which will be applied to calibrate the logit predicted by the classifier, $\boldsymbol{z}$. However, the magnitude and range of the prediction logits and the cosine similarity scores differ ($\boldsymbol{z} \in (-\infty, \infty)$, $\boldsymbol{\pi} \in [-1, 1]$), so we must normalize them before we can aggregate both vectors. Hence, the calibrated logit can be computed as follows:

$$\frac{1}{2}\left(\boldsymbol{z}\|\boldsymbol{\pi}\| + \boldsymbol{\pi}\|\boldsymbol{z}\|\right). \tag{7}$$

2) RoI Feature Transformation: Based on our observations (See Figs. 7 and 8), the quality of the support RoI features is inadequate for representing novel categories since some features have a high degree of similarity despite belonging to different categories, which will not give any performance gain if we introduce those features to calibrate the logits. Thus, it necessitates incorporating extra learning parameters, $\mathbf{W} \in \mathbb{R}^{d_2 \times d_2}$ for the linear transformation of RoI features to render them more discriminative. The learnable parameters are initialized using a matrix that minimizes the following:

$$\arg\min_{\mathbf{W}} \frac{1}{K^2} \sum_{i=1}^{K} \sum_{j=1}^{K} \left(S_C\left(\mathbf{W}^T \boldsymbol{f}, \mathbf{W}^T \overline{\boldsymbol{f}}^{(c)}\right) - 1[\![c_i = c_j]\!]\right)^2. \tag{8}$$

Essentially, the objective function above seeks a transformation matrix that maximizes the similarity of features from the same category while minimizing the similarity of features from different categories. Then, we apply the linear transformation on the RoI features, $\boldsymbol{f}$ and $\overline{\boldsymbol{f}}^{(c)}$, before computing the cosine

TABLE I
PERFORMANCE COMPARISON ON THREE NOVEL SPLITS OF THE PASCAL VOC DATASET (1, 2, 3, 5, 10 SHOTS). THE BEST PERFORMANCE IS IN **BOLD**.

Method	Backbone	Novel Split 1					Novel Split 2					Novel Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
Single Run																
FSRW [6]	YOLOv2	14.8	15.5	26.7	33.9	47.2	15.7	15.2	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
Meta R-CNN [8]	ResNet-101	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA w/ cos [15]	ResNet-101	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
MPSR [16]	ResNet-101	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
Fan *et al*. [51]	ResNet-101	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
SRR-FSD [17]	ResNet-101	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
CME [10]	ResNet-101	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
FSCE [52]	ResNet-101	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
TIP [18]	ResNet-101	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
FSOD-UP [19]	ResNet-101	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
DeFRCN [20]	ResNet-101	57.0	58.6	64.3	67.8	67.0	35.8	42.7	51.0	54.5	52.9	52.5	56.6	55.8	60.7	62.5
Meta-DETR [12]	ResNet-101	40.6	51.4	58.0	59.2	63.6	37.0	36.6	43.7	49.1	54.6	41.6	45.9	52.7	58.9	60.6
Kaul *et al*. [21]	ResNet-101	54.5	53.2	58.8	63.2	65.7	32.8	29.2	50.7	49.8	50.6	48.4	52.7	55.0	59.6	59.6
KFSOD [14]	ResNet-50	44.6	-	54.4	60.9	65.8	37.8	-	43.1	48.1	50.4	34.8	-	44.1	52.7	53.9
FCT [22]	PVTv2-B2-Li	49.9	57.1	57.9	63.2	67.1	27.6	34.5	43.7	49.2	51.2	39.5	54.7	52.3	57.0	58.7
Fan *et al*. [53]	ResNet-50	40.1	44.2	51.2	62.0	63.0	33.3	33.1	42.3	46.3	52.3	36.1	43.1	43.5	52.0	56.0
MFDC [49]	ResNet-101	63.4	66.3	67.7	69.4	68.1	42.1	46.5	53.4	55.3	53.8	56.1	58.3	59.0	62.2	63.7
MFE [54]	ResNet-101	55.0	55.5	59.2	-	59.7	34.7	38.2	44.1	-	46.4	49.5	44.2	47.3	-	55.4
Norm-VAE [55]	ResNet-101	62.1	64.9	67.8	69.2	67.5	39.9	46.8	54.4	54.2	53.6	58.2	60.3	61.0	64.0	65.5
SD-FSOD [24]	ResNet-101	64.6	67.1	67.4	69.0	70.7	42.4	48.3	52.7	55.4	56.0	57.0	59.7	60.4	63.5	64.6
Ours	ResNet-101	**66.9**	**73.5**	**73.7**	**75.4**	**76.4**	**43.5**	**48.4**	**54.7**	**57.0**	**59.6**	**62.0**	**63.9**	**62.1**	**70.2**	**70.6**
Multiple Runs																
TFA w/ cos [15]	ResNet-101	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
FSDetView [9]	ResNet-101	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
FSCE [52]	ResNet-101	32.9	44.0	46.8	52.9	59.7	23.7	30.6	38.4	43.0	48.5	22.6	33.4	39.5	47.3	54.0
DeFRCN [20]	ResNet-101	40.2	53.6	58.2	63.6	66.5	29.5	39.7	43.4	48.1	52.8	35.0	38.3	52.9	57.7	60.8
FCT [22]	PVTv2-B2-Li	38.5	49.6	53.5	59.8	64.3	25.9	34.2	40.1	44.9	47.4	34.7	43.9	49.3	53.1	56.3
Ours	ResNet-101	**53.2**	**65.9**	**68.8**	**72.7**	**74.1**	**36.6**	**46.5**	**50.2**	**54.7**	**58.7**	**49.3**	**59.4**	**61.3**	**66.4**	**69.2**

similarity scores in Equation (6), to make the representation more discriminative:

$$\pi_c = S_C\left(\mathbf{W}^T\boldsymbol{f}, \mathbf{W}^T\overline{\boldsymbol{f}}^{(c)}\right) = \frac{\mathbf{W}^T\boldsymbol{f}\cdot\mathbf{W}^T\overline{\boldsymbol{f}}^{(c)}}{\|\mathbf{W}^T\boldsymbol{f}\|\|\mathbf{W}^T\overline{\boldsymbol{f}}^{(c)}\|}. \quad (9)$$

IV. EXPERIMENTS

The following sections will describe the typical benchmark for a fair comparison of our work with the literature. Then we will provide our implementation details and the hyper-parameters used in this work (Section IV-B). In Section IV-C, we will compare our work to the existing state-of-the-art methods and, lastly, present ablation studies on the proposed modules to justify the advantages of the modules.

A. Evaluation Benchmarks

In accordance with widely recognized benchmarks, we assess our work on the PASCAL VOC [56] and MS-COCO [57] datasets, employing identical data split configurations as outlined in [15]. Specifically, we present two distinct settings: the first reports the results of a single repetition (corresponding to the metrics reported in [6]), while the second setting reports the average metrics across multiple repetitions (30 for PASCAL VOC, 10 for MS-COCO), each initiated with a unique random seed.

PASCAL VOC [56] is a 20 classes object detection dataset that is relatively small. Following the formulation of prior work, the benchmark is composed of three random combinations of base and novel categories, with 15 base classes and 5 classes allocated for novel fine-tuning in each configuration. We sample the K-shot images ($K = 1, 2, 3, 5, 10$) according to the list by [6] and report the average precision for IoU = 0.5 (nAP50) on the novel classes.

MS-COCO [57] comprises 80 categories, 20 of which overlap with the categories in the PASCAL VOC dataset. The 20 classes are utilized for novel fine-tuning in the FSOD benchmark setting, while the remaining 60 are labelled as base classes. We experiment with six different shot settings, *i.e.* $K = 1, 2, 3, 5, 10, 30$, and report the COCO-style mean average precision on novel classes (nAP), which is the standard metric used in this benchmark.

B. Implementation Details

As previously stated, we selected Faster R-CNN [33] as our primary object detector for the experiments. The two-stage detector consists of the backbone, RPN, and prediction head, with the backbone being a ResNet-101 [58] pre-trained on ImageNet [59]. The RPN and prediction head are randomly initialized CNN and MLP, respectively.

We utilize a widely adopted two-stage fine-tuning approach, initially proposed by Wang *et al*. [15], in our object detection framework. This approach entails training the base object detector with a base dataset, followed by fine-tuning only the prediction layers using a few-shot dataset. However, we have made modifications based on recent findings. In particular,

Sun *et al*. [52] emphasized the necessity of fine-tuning the Region Proposal Network (RPN) to enhance the recall rate for novel categories, thereby improving the overall detection performance. This aspect was further elaborated upon by the extensive study conducted by Kaul *et al*. [21], which confirmed the importance of updating the RPN's parameters during the fine-tuning stage.

The experiments are running on a machine with 4 NVIDIA A100 GPUs. All models are trained using the SGD optimizer with a momentum of 0.9 and weight decay of 5×10^{-5}. For the base-training phase, the initial learning rate was set to 0.02, which was then reduced to 0.01 for the fine-tuning phase. In contrast to previous approaches that freeze the parameters of the feature extractor, we employed a lower learning rate (*i.e*. 0.01 times lower than other components) for the backbone and RoI feature extractor. The number of training iterations in the few-shot fine-tuning phase varied depending on the dataset and the number of shots, denoted as K, utilized in the experiment.

The number of centroids, N, used for feature calibration is 24 for both PASCAL VOC and MS-COCO. The α involved in Equation (4) is set at 0.1 after careful selection from a limited set of values. The feature extractor used to construct the support set representation is the same ResNet-101 network pre-trained on ImageNet. We employ a standard SGD optimizer with a learning rate of 1.0 to help us determine the best transformation matrix, $\mathbf{W}$, that minimizes the objective function described in Equation (8).

C. Comparison to SotA

We have compiled a selection of recent publications that have been tested on the widely recognized benchmark datasets, PASCAL VOC and MS-COCO, to evaluate performance. While most studies adopt similar experimental settings for comparison, it is worth noting the distinctive methodologies employed by SRR-FSD [17] and the recent study by Kaul *et al*. [21]. SSR-FSD introduces a novel component that incorporates word embeddings [60], [61] of class labels into the object detection pipeline, utilizing a knowledge graph sampled from WordNet [62]. In contrast, the methodology proposed by Kaul *et al*. [21] requires an additional set of unlabeled images for pseudo-annotations, aimed at fine-tuning the object detector. Apart from the approaches mentioned above, most studies do not incorporate additional data into their experiments, opting solely to initialize the backbone weights using a pre-trained ImageNet classifier.

The PASCAL VOC results are summarized in Table I. In accordance with established standards, we report the n$AP50$ for each of the three novel split settings. Our approach consistently outperforms existing methods across all split/shot settings, exhibiting a substantial margin of improvement in both single run and the average metrics of multiple runs. On the other hand, the MS-COCO results are presented in Table II, where we report the nAP for each shot setting. While our method slightly underperforms compared to MFDC [49] in the 1 and 2 shot settings, we outperform all previous methods in the remaining settings. Additionally, our method outperforms all other approaches by a significant margin in the multiple run setting, which provides more consistent and reliable comparison results compared to single runs.

TABLE II
PERFORMANCE COMPARISON ON MS-COCO DATASET (1, 2, 3, 5, 10, 30 SHOTS). THE BEST PERFORMANCE IS IN **BOLD**.

Method	Backbone	Shot					
		1	2	3	5	10	30
Single Run							
FSRW [6]	YOLOv2	-	-	-	-	5.6	9.1
Meta R-CNN [8]	ResNet-101	-	-	-	-	8.7	12.4
TFA w/ cos [15]	ResNet-101	3.4	4.6	6.6	8.3	10.0	13.7
MPSR [16]	ResNet-101	2.3	3.5	5.2	6.7	9.8	14.1
Fan *et al*. [51]	ResNet-101	4.2	5.6	6.6	8.0	9.6	13.5
SRR-FSD [17]	ResNet-101	-	-	-	-	11.3	14.7
CME [10]	YOLOv2	-	-	-	-	15.1	16.9
FSCE [52]	ResNet-101	-	-	-	-	11.9	16.4
TIP [18]	ResNet-101	-	-	-	-	16.3	18.3
FSOD-UP [19]	ResNet-101	-	-	-	-	11.0	15.6
DeFRCN [20]	ResNet-101	6.5	11.8	13.4	15.3	18.6	22.5
DAnA [11]	ResNet-50	-	-	-	-	18.6	21.6
Meta-DETR [12]	ResNet-101	7.5	-	13.5	15.4	19.0	22.2
Kaul *et al*. [21]	ResNet-101	-	-	-	-	17.8	24.5
KFSOD [14]	ResNet-50	-	-	-	-	18.5	-
FCT [22]	PVTv2-B2-Li	5.6	7.9	11.1	14.0	17.1	21.4
MFDC [49]	ResNet-101	**10.8**	13.9	15.0	16.4	19.4	22.7
MFE [54]	ResNet-101	10.5	13.5	15.8	**17.9**	20.1	24.1
Norm-VAE [55]	ResNet-101	9.5	13.7	14.3	15.9	18.7	22.5
SD-FSOD [24]	ResNet-101	-	-	-	-	19.2	22.5
Ours	ResNet-101	**10.8**	**14.0**	**15.9**	17.8	**21.0**	**25.2**
Multiple Runs							
TFA w/ cos [15]	ResNet-101	1.9	3.9	5.1	7.0	9.1	12.1
FSDetView [9]	ResNet-101	4.5	6.6	7.2	10.7	12.5	14.7
FSCE [52]	ResNet-101	-	-	-	-	11.1	15.3
DeFRCN [20]	ResNet-101	4.8	8.5	10.7	13.6	16.8	21.2
FCT [22]	PVTv2-B2-Li	5.1	7.2	9.8	12.0	15.3	20.2
Ours	ResNet-101	**7.7**	**11.5**	**14.0**	**16.5**	**19.5**	**23.8**

Overall, our method shows better detection precision compared to state-of-the-art approaches. On top of that, our approach does not introduce supplementary data, such as the additional unlabeled images used in work [21], which is a significant advantage. Moreover, we achieve improved performance without introducing excessive complexity compared to other methods, such as FCT [22], MFDC [49], MFE [54], and SD-FSOD [24] which also utilizes DeFRCN as the baseline model.

1) Categorization of SotA: The methods we compared can be broadly categorized into two distinct groups: those derived from few-shot learning techniques and adapted for FSOD, and those specifically designed to address common problems in object detection tasks.

For instance, FSRW [6] employs meta-learning alongside a reweighting module to transform few-shot supports into global vectors, highlighting the relevance of meta features for detecting objects from given support classes. Similarly, many meta-learning approaches such as Meta R-CNN [8], Fan *et al*. [51], TIP [18], DAnA [11], Meta-DETR [12], and KFSOD [14], exploit the similarity between the support set and query set for FSOD, much like FSRW. On the other hand, SRR-FSD [17] leverages the semantic relationship between base and novel classes using word embeddings and applies the learned knowledge graph to build a robust object detector. Fan *et al*. [53] propose calibration modules at different levels to mitigate bias towards either base or novel classes. CME [10] enforces margin equilibrium between base classes using adversarial min-max optimization to accurately represent novel

classes, while FSOD-UP [19] learns a universal prototype that models intrinsic characteristics across different base categories and applies them to enhance features from both base and novel classes. Additionally, Norm-VAE [55] employs a variational autoencoder to generate data, addressing the issue of limited training samples for novel classes.

Conversely, some methods are specifically designed to tackle common object detection tasks. TFA [15], for instance, implements a two-stage fine-tuning strategy similar to other fine-tuning strategies for transfer learning, which is not necessarily exclusive to FSOD. MPSR [16] addresses the scale variation problem in object detection by enriching the scales of training samples using a proposed algorithm. FSCE [52] learns more discriminative feature representations by applying batch contrastive learning to explicitly model intra-class similarity and inter-class distinction, a technique that is also applicable to general object detection methods. MFE [54], similar to FSCE, aims to improve feature representation but from different perspectives, including spatial, task, and regularization levels.

There are also hybrid approaches that combine elements of both few-shot learning and object detection-specific techniques. DeFRCN [20], for example, features a decoupled layer to halt the propagation of gradients from the class-agnostic downstream task, effectively decoupling both tasks. This can be applied to other object detection methods as well. PCB improves prediction accuracy by exploiting the similarities of support and query sets, similar to other few-shot learning methods. Methods like MFDC [49] and SD-FSOD [24] are similar to DeFRCN but propose a distillation learning framework to transfer the learned distribution of features from base samples to the robust prediction of samples from few-shot novel classes. Our approach also fits into this hybrid category as we refine feature representation precision through our distinctive feature-level calibration algorithm, which uses learnable centroids to minimize representation variance, and apply a logit calibration module to adjust predictions by leveraging the support set, thus enhancing model accuracy for given queries.

Finally, Kaul *et al.* [21] does not fit into either category, functioning more like a semi-supervised method. It requires abundant unlabelled data to enrich the training samples, which does not strictly adhere to traditional few-shot learning settings. We summarize this section in Fig. 4 for a clearer overview and better insights.

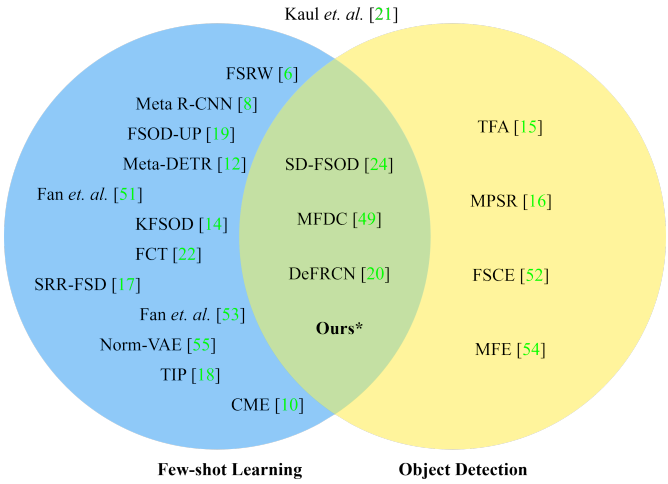


Fig. 4. Categorization of the state-of-the-art methods discussed and compared in this work.

D. Ablation Studies

This section will present our studies on each proposed component and explain how they would help the FSOD task. To begin, we will demonstrate the efficacy of each module by demonstrating the performance achieved by incorporating the component or combination of components. Following that, we will justify our hyper-parameter selections mentioned in Section III-B. The visualization of the latent feature space is then performed to investigate the effects of feature calibration. In addition, we will display the RoI feature to observe how the learned linear transformation helps render the RoI feature more discriminative and calibrates the prediction logit.

1) Effectiveness of different modules: We demonstrate the effectiveness of different modules by investigating how each module contributes to detection performance. In this experiment, we evaluate the performance gain by adding each module and combination of modules to the detection accuracy across all three split settings of PASCAL VOC. We report the n$AP50$ scores for different configurations in Table III. We can easily observe that the feature calibration module improves detection precision significantly. Notably, adding the logit calibration provides a slight gain in performance, but pairing the logit calibration with the learned linear transformation further improves the detection accuracy. The result is consistent with our observation of the poor discriminability of the support RoI features used in the logit calibration module as shown in Fig. 7.

2) Hyper-parameters selection: We carefully choose the number of centroids, N, and the update hyper-parameter, α, used in the feature calibration module by analyzing the performance of different configurations and ensuring they consistently perform best across all three split settings in the PASCAL VOC experiment. Table IV illustrates the results of our experiment on the PASCAL VOC dataset. Specifically, N determines the number of learnable centroids, with an excessive or insufficient number leading to overfitting or underfitting, respectively. α controls the update momentum of the centroids, functioning similarly to a learning rate but specifically for the centroids. These parameters serve different purposes: N relates to the number of parameters of a neural network, while α is similar to the learning rate for updating these parameters. To illustrate this relationship, we present the average performance for various combinations of N and α in the PASCAL VOC Novel Split 1 setting (See Table V). Our results indicate that an N of 24 is optimal, as deviations in either direction significantly reduce performance regardless of the α value. Although α is less sensitive, we found that 0.1 is the ideal value, where deviations from this can lead to slightly lower performance. Thus, we choose the best performing option, $N = 24$ and $\alpha = 0.1$.

TABLE III
FEW-SHOT OBJECT DETECTION PERFORMANCE ON THE PASCAL VOC DATASET. FC: SEC. III-B, LC: SEC. III-C, **W**: SEC. III-C2

Settings	FC	LC	**W**	Novel Split 1					Novel Split 2					Novel Split 3				
				1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
Baseline				48.5	48.4	52.1	61.1	64.3	31.3	39.4	46.4	49.2	51.8	40.2	50.8	51.7	59.2	61.4
LC only		✓		50.6	59.7	58.6	63.6	67.2	32.2	37.0	44.7	46.9	54.6	45.9	49.6	54.5	61.6	63.7
LC + **W**		✓	✓	56.7	62.4	62.9	68.5	69.9	38.2	42.7	50.5	52.7	55.7	44.6	52.3	58.4	64.0	64.1
FC only	✓			61.0	65.3	63.9	71.7	73.0	39.8	45.3	53.3	55.9	57.0	55.9	59.4	60.5	65.0	66.2
FC + LC	✓	✓		55.0	67.5	67.6	69.6	73.6	39.1	43.5	50.2	53.3	57.4	54.4	61.1	61.3	68.2	69.5
FC + LC + **W**	✓	✓	✓	**66.9**	**73.5**	**73.7**	**75.4**	**76.4**	**43.5**	**48.4**	**54.7**	**57.0**	**59.6**	**62.0**	**63.9**	**62.1**	**70.2**	**70.6**

TABLE IV
ABLATION STUDY ON THE HYPER-PARAMETERS OF FEATURE CALIBRATION MODULE.

	Novel Split 1					Novel Split 2					Novel Split 3				
N	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
16	50.2	53.2	57.2	67.7	68.6	37.8	41.3	51.1	53.0	55.8	51.7	56.9	57.8	63.9	66.9
20	54.8	61.3	59.6	67.3	69.3	39.5	41.9	50.7	55.4	54.2	52.3	56.9	53.7	63.3	**67.3**
24	**61.0**	**65.3**	**63.9**	**71.7**	**73.0**	**39.8**	**45.3**	**53.3**	**55.9**	**57.0**	**55.9**	**59.4**	**60.5**	**65.0**	66.2
28	55.1	60.6	59.6	69.7	69.9	33.6	41.5	51.4	53.4	54.4	52.8	56.6	56.1	62.9	65.8
32	57.1	61.4	60.0	69.5	70.6	39.3	44.0	53.1	54.8	53.8	48.4	56.8	54.3	63.4	67.1
α	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
1.0	58.7	61.8	62.4	70.1	71.0	35.3	41.4	50.5	54.3	56.8	50.5	57.2	56.9	63.5	67.9
0.5	55.1	60.5	58.3	68.7	69.9	37.3	40.2	45.4	54.1	53.8	50.9	54.6	57.5	63.6	67.0
0.1	**61.0**	**65.3**	**63.9**	**71.7**	**73.0**	**39.8**	**45.3**	**53.3**	**55.9**	**57.0**	**55.9**	**59.4**	**60.5**	**65.0**	66.2
0.05	57.8	62.3	61.6	70.2	69.9	37.5	40.2	50.5	54.0	55.1	52.9	56.3	57.1	64.9	**69.1**
0.01	60.0	65.6	62.8	70.8	71.4	39.6	36.1	50.4	53.6	55.1	47.0	51.2	54.5	62.2	65.8

TABLE V
PERFORMANCE ACROSS VARIOUS SETS OF N AND α COMBINATIONS. THE SCORES REPRESENT THE AVERAGE ACCURACY OVER $1-10$ SHOTS IN THE PASCAL VOC NOVEL SPLIT 1 SETTING.

N \ α	**0.01**	**0.05**	**0.1**	**0.5**	**1.0**
16	59.0	59.3	59.4	59.0	57.1
20	62.4	62.4	62.5	62.4	62.4
24	66.1	64.4	67.0	62.5	64.8
28	62.8	62.9	63.0	62.8	62.7
32	63.7	63.7	63.7	63.7	63.6

3) Logit calibration: Our logit calibration module and prototype calibration block (PCB) [20] have some notable similarities, especially the use of cosine similarity in the calibration process. However, our logit calibration module employs a different approach, where we calibrate the prediction logits rather than the predicted category score. As previously stated, PCB can produce negative probability scores, which defies the notion of probability and makes no sense for class prediction. While our method eliminates this concern, it also eliminates the need of additional feature extractor which increases time and memory complexity. Furthermore, we show the performance (n$AP50$) consistently drops across all three split settings in the PASCAL VOC experiment when our logit calibration module is replaced with PCB, while keeping the rest of the configurations unchanged. The experiment results are included in Table VI.

4) Similarity Metrics: In this work, we use similarity metrics in both our feature calibration and logit calibration modules for different purposes. In feature calibration, we use them to assign features to the nearest centroids, while in logit calibration, we use the score to calibrate the logits predicted by the classification head based on their similarity to the reference RoI features. We adopted cosine similarity due to its effectiveness in representing the similarity of feature vectors in terms of orientation. It has been demonstrated in previous works [63], [64] that cosine similarity performs better than Euclidean and Manhattan distances. Furthermore, cosine similarity has been widely used in previous FSOD works [12], [15], [20], [49] to represent the similarity of convolution features, including the computation of similarity scores in PCB [20] and the learning of commonalities in MFDC [49]. Additionally, we also experimented with other widely used similarity metrics such as Euclidean distance and Manhattan distance. For distance metrics, we can easily compute the similarity score using the RBF kernel given by: $K(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\gamma d\left(\boldsymbol{x},\boldsymbol{y}\right)\right)$, where $d(\boldsymbol{x},\boldsymbol{y})$ is the distance function. From Table VII, we observe that the use of different similarity metrics in the feature calibration module does not significantly affect the results, as they are primarily used to assign the nearest centroid for each feature vector, and the distance metrics in the feature space yield similar assignments. Therefore, choosing a more computationally efficient metric like cosine similarity is justified, as the directional information measured by cosine similarity effectively fulfills the task. Furthermore, cosine similarity performs best when used to calibrate the logits compared to other metrics, as shown in Table VII. This further justifies the use of cosine similarity in both our feature calibration and logit calibration modules.

5) Efficiency Analysis: In this section, we compare our work to the widely used baseline, DeFRCN, and the Faster R-CNN benchmark to demonstrate the added complexity of

TABLE VI
PERFORMANCE COMPARISON AFTER SUBSTITUTING THE LOGIT CALIBRATION (LC) MODULE WITH THE PCB [20] ON PASCAL VOC DATASET.

	Novel Split 1					Novel Split 2					Novel Split 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
Ours	**66.9**	**73.5**	**73.7**	**75.4**	**76.4**	**43.5**	**48.4**	**54.7**	**57.0**	**59.6**	**62.0**	**63.9**	**62.1**	**70.2**	**70.6**
LC → PCB	64.1	67.4	68.2	72.6	73.1	42.2	45.2	50.4	54.8	56.1	56.8	60.1	59.8	65.8	66.2

TABLE VII
ABLATION STUDY ON VARIOUS SIMILARITY METRICS USED IN FEATURE CALIBRATION (FC) AND LOGIT CALIBRATION (LC) MODULES.

Module	Metrics	Novel Split 1					Novel Split 2					Novel Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FC	Cosine	**66.9**	**73.5**	**73.7**	**75.4**	**76.4**	**43.5**	**48.4**	**54.7**	**57.0**	59.6	**62.0**	**63.9**	62.1	**70.2**	**70.6**
	Euclidean	65.3	71.5	73.2	75.2	74.2	43.2	47.0	54.1	56.9	**59.9**	59.4	62.6	**62.3**	66.8	68.0
	Manhattan	64.4	70.5	73.2	75.3	75.9	39.8	45.0	54.0	56.8	58.1	53.9	57.8	60.7	67.3	68.7
LC	Cosine	**66.9**	**73.5**	**73.7**	**75.4**	**76.4**	**43.5**	**48.4**	**54.7**	**57.0**	**59.6**	**62.0**	**63.9**	**62.1**	**70.2**	**70.6**
	Euclidean	53.1	68.6	70.0	74.2	71.2	33.4	46.7	51.9	52.0	54.5	37.0	62.9	59.7	65.7	66.3
	Manhattan	57.9	57.4	68.1	68.9	65.5	38.8	35.4	34.4	48.2	51.7	52.7	43.7	57.1	64.2	64.5

TABLE VIII
EFFICIENCY COMPARISON BETWEEN THE BASELINE FASTER R-CNN [33], DEFRCN [20], AND OUR PROPOSED MODEL.

	Input Size	# Param (M)	GFLOPs	Memory (MiB)	Infer. time (s)
Faster R-CNN [33]	$(486, 500, 3)$	52.12	374.6	3408	0.046
DeFRCN [20]		96.67	478.3	3764	0.080
Ours		57.36	381.8	3466	0.045

our approach to the vanilla object detector. All efficiency experiments were conducted on PASCAL VOC Novel Split 1, and the results should be consistent across different splits. As shown in Table VIII, our proposed model is highly efficient compared to DeFRCN due to several factors discussed earlier. DeFRCN's significant complexity arises from the PCB module, which employs an additional feature extractor—specifically, a ResNet-101. This results in a substantial increase in floating point operations (FLOPs) and the total number of parameters. Consequently, the inference time nearly doubles, as PCB requires running another feature extractor with a different ResNet-101 backbone to compute the similarity score.

In contrast, our method, while slightly slower than the vanilla Faster R-CNN, introduces only marginal complexity. This is evident from the per-sample inference time, which increases by just 0.001 seconds, and a modest 2% increase in FLOPs. This efficiency is attributed to our lightweight feature and logit calibration, which requires a minimal number of parameters to perform the calibration.

V. VISUALIZATIONS

A. Visualization of centroids

In this section, we assess the ability of the centroids to adapt to novel classes while avoiding overfitting. To achieve this, we generate similarity maps between the centroids and the feature maps of the test dataset. It is important that these centroids can generalize and accurately identify previously unseen novel objects, even when trained with a limited number of examples. In Fig. 5, we present the visualization of such similarity heatmaps using the test set from PASCAL VOC. The centroids undergo fine-tuning on a 1-shot novel set from novel split 1, where only one training instance is provided for each novel category. Remarkably, the figure demonstrates that despite being trained on a limited samples, the centroids effectively differentiate between various objects within the image. These findings provide strong evidence that our proposed method successfully learns the distinctive features of novel categories using only a few samples.

B. Visualization of latent feature space

This section helps us to understand the effect of feature calibration on the generated representations. We illustrate the representations generated by the backbone without feature calibration and those produced with feature calibration. To visualize the feature map, we do a channel-wise sum on all the channels in the feature map and display the normalized activation value of it. The results are shown in Fig. 6. Each sample is organized into three columns: the input image, the feature produced without feature calibration, and the feature produced with feature calibration. We discover that the representation produced by feature calibration has less background noise, enabling us to distinguish between foreground and background objects. Furthermore, we notice that the activation values in representations created without the feature calibration module often focus on a specific part of the object.

C. Visualization of the RoI features

The RoI features are illustrated in Fig. 7. In MS-COCO experiments, 20 novel classes are reserved for the novel fine-tuning phase, and only K (*e.g.*, $K = 10, 30$) objects are

Fig. 5. Visualization of the similarity heatmaps between the centroids, trained using a 1-shot setting on MS-COCO, and the feature maps of the MS-COCO validation set.

Fig. 6. Visualization of the feature map generated by the backbone (MS-COCO, 1-shot). Each sample is organized as follows: the input image, the feature map without using calibration, and the feature map with calibration. Each figure displayed is the channel-wise sum of the d_1-dimensional feature map.

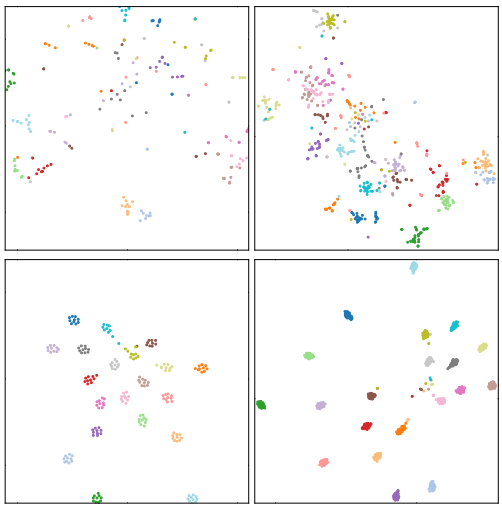


Fig. 7. The t-SNE [65] plot shows the retrieved RoI features from the support set. We visualize the RoI features of a 10-shot (left) and 30-shot (right) support set. Second row shows the RoI features in the first row after the transformation, as described in Sec. III-C2.

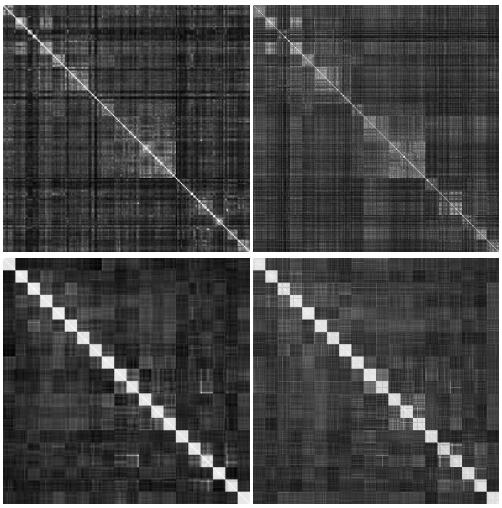


Fig. 8. Pairwise cosine similarity between the K-shot support set (Left: 10-shot; Right: 30-shot). The samples are grouped next to each other according to their categories. The second row illustrates the pairwise similarity after the transformation has been applied to the RoI features, as defined in Sec. III-C2.

chosen as the training set. We extract the RoI features from these samples and illustrate their topological relationship in the feature space using t-SNE [65]. We can observe from the figure that there is no evident pattern in those features since

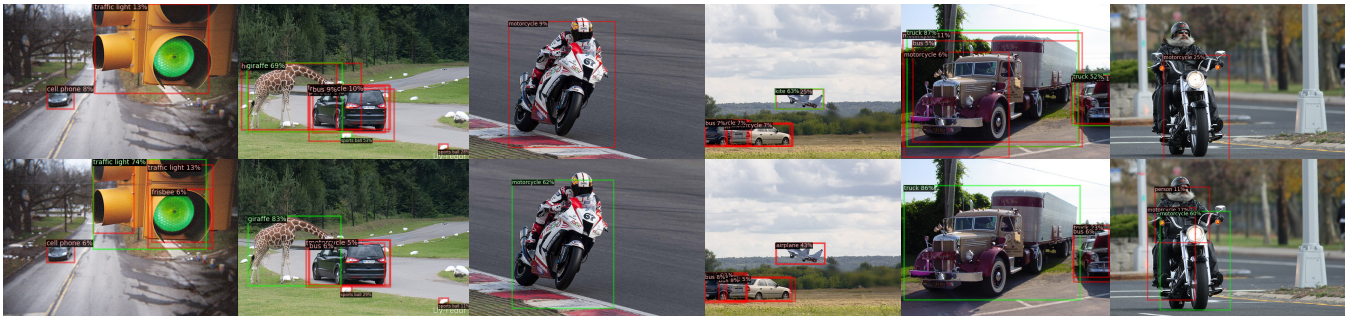


Fig. 9. Comparison of prediction results with and without logit calibration (MS-COCO 1-shot): logit calibration not applied (above) vs. logit calibration applied (below).

we cannot visually group the features into clusters. To further validate this, we plot the pairwise cosine similarity of each RoI feature and present the measurement in Fig. 8. We arrange the RoI features according to their categories; thus, there should be a high level of similarity between features in the same category and vice versa. However, we see a high similarity between features from different categories, which would be misleading if we utilize the feature directly. Therefore, we propose to treat the RoI features by applying a linear transformation to improve their discriminability. Figs. 7 and 8 show the t-SNE and pairwise cosine similarity scores after the transformation in the second row. The visualizations reveal that a simple linear modification effectively makes the RoI features more discriminative. We can easily see that the RoI features from the same categories formed a cluster and are positioned far apart from other clusters. The same behavior is shown in the pairwise cosine similarity score, where the similarity scores of features in the same class increase and vice versa.

D. Logit calibration

As mentioned in previous sections, logit calibration serves as an essential mechanism for addressing inaccurate prediction results by adjusting the underlying distribution of predictions. By applying calibration at the logit level, our method enhances the confidence scores of true positive samples while reducing the confidence scores of false positive predictions. Fig. 9 showcases a randomly selected set of calibration results. It is apparent that our method effectively boosts the prediction scores for correctly identified categories, while simultaneously lowering the scores for false positive predictions. For instance, in the fourth column, the initial misclassification of a bird as a cat is rectified after logit calibration. This demonstrates the ability of our method to refine predictions by exploiting the similarities between the support/query pair of RoI features. Moreover, logit calibration enables us to include additional predictions that were previously undetected due to low confidence scores (the bus in the first column).

E. Failure Cases

While our method effectively calibrates predictions by boosting the confidence scores of true positives and reducing false positives, it has limitations in scenarios where objects are occluded or poorly lit. As illustrated in Fig. 10, the model struggles with partially occluded objects, such as the chair in the second example of the first row and the frisbee in the person's hand in the first example of the second row. Additionally, the third examples in both rows demonstrate the model's difficulty in detecting persons in dark regions, where only silhouettes are faintly visible. These cases often require a context-aware approach that can infer object locations based on surrounding objects and environmental cues. For instance, a potential solution might involve leveraging a foundation model with advanced scene understanding capabilities, which could enhance object detection by better interpreting the context within an image. As our current method addresses different but equally significant challenges in this field, we will defer this exploration to future work.

VI. Conclusion

In this work, we presented a dual calibration framework that integrates feature and logit calibration modules for the representation and prediction levels, respectively. Firstly, we employed a feature calibration module to reduce variance in the feature map using a set of learnable centroids that eventually learn to represent the ensemble of partition centroids across the entire dataset. Through qualitative visualizations and quantitative evaluation of FSOD performance, we demonstrate that the centroids can accurately represent objects from novel categories, even with limited training samples. Additionally, we introduced logit calibration, utilizing the support set to calibrate prediction logits by comparing the similarity of improved Region of Interest (RoI) features. By observation, we believe it is necessary to include RoI transformation to increase the discriminative power of the RoI features, which we show improves the performance of logit calibration. Our method achieved state-of-the-art performance on the PASCAL VOC and MS-COCO benchmarks, and we provide comprehensive ablation experiments and visualizations to highlight the significance of each module. This dual calibration framework offers a robust approach to enhance object detection accuracy in few-shot context.

References

[1] X.-j. Han, Z. Qu, S.-Y. Wang, S.-F. Xia, and S.-Y. Wang, "End-to-end object detection by sparse R-CNN with hybrid matching in complex traffic scenes," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 512–525, 2024. 1

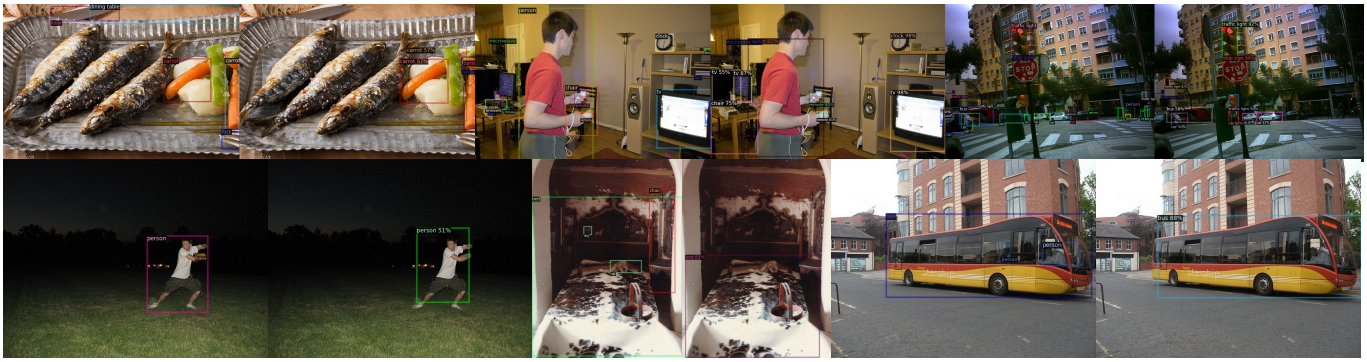


Fig. 10. Examples of our model's failure to detect certain objects in the MS-COCO 30-shot setting. Each pair consists of a left image showing the ground-truth annotation and a right image showing our model's prediction.

[2] S.-y. Wang, Z. Qu, and L.-y. Gao, "Multi-spatial pyramid feature and optimizing focal loss function for object detection," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1054–1065, 2024. 1

[3] Y.-H. Lin and Y.-S. Wang, "Modular learning: Agile development of robust traffic sign recognition," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 764–774, 2024. 1

[4] J. Chen, J. Zhu, R. Xu, Y. Chen, H. Zeng, and J. Huang, "ORNet: Orthogonal re-parameterized networks for fast pedestrian and vehicle detection," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2662–2674, 2024. 1

[5] D. Dodge and M. Yilmaz, "Convex vision-based negative obstacle detection framework for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 778–789, 2023. 1

[6] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *2019 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 8419–8428. 1, 3, 4, 6, 7

[7] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *2019 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9924–9933. 1, 3

[8] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta R-CNN: Towards general solver for instance-level low-shot learning," in *2019 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9576–9585. 1, 3, 4, 6, 7

[9] Y. Xiao and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," in *Comput. Vis. – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer Int. Publishing, 2020, pp. 192–210. 1, 3, 6, 7

[10] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye, "Beyond max-margin: Class margin equilibrium for few-shot object detection," in *2021 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2021, pp. 7359–7368. 1, 3, 4, 6, 7

[11] T.-I. Chen, Y.-C. Liu, H.-T. Su, Y.-C. Chang, Y.-H. Lin, J.-F. Yeh, W.-C. Chen, and W. H. Hsu, "Dual-awareness attention for few-shot object detection," *IEEE Transactions on Man-Machine Systems*, vol. 25, pp. 291–301, 2023. 1, 3, 7

[12] G. Zhang, Z. Luo, K. Cui, S. Lu, and E. P. Xing, "Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation," *IEEE_J_PAMI*, pp. 1–12, 2022. 1, 3, 4, 6, 7, 9

[13] L. Zhang, S. Zhou, J. Guan, and J. Zhang, "Accurate few-shot object detection with support-query mutual guidance and hybrid loss," in *2021 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2021, pp. 14 419–14 427. 1, 3

[14] S. Zhang, L. Wang, N. Murray, and P. Koniusz, "Kernelized few-shot object detection with efficient integral aggregation," in *2022 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2022, pp. 19 185–19 194. 1, 3, 4, 6, 7

[15] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu, "Frustratingly simple few-shot object detection," in *Proc. 37th Int. Conf. Mach. Learning*, ser. Proc. Mach. Learning Res., H. D. III and A. Singh, Eds., vol. 119. PMLR, 2020-07-13/2020-07-18, pp. 9919–9928. 1, 2, 3, 4, 6, 7, 8, 9

[16] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *Comput. Vis. – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer Int. Publishing, 2020, pp. 456–472. 1, 3, 4, 6, 7, 8

[17] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *2021 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2021, pp. 8778–8787. 1, 2, 3, 4, 6, 7

[18] A. Li and Z. Li, "Transformation invariant few-shot object detection," in *2021 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2021, pp. 3093–3101. 1, 3, 4, 6, 7

[19] A. Wu, Y. Han, L. Zhu, and Y. Yang, "Universal-prototype enhancing for few-shot object detection," in *2021 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 9547–9556. 1, 3, 4, 6, 7, 8

[20] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang, "DeFRCN: Decoupled faster R-CNN for few-shot object detection," in *2021 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 8661–8670. 1, 2, 3, 4, 6, 7, 8, 9, 10

[21] P. Kaul, W. Xie, and A. Zisserman, "Label, verify, correct: A simple few shot object detection method," in *2022 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2022, pp. 14 217–14 227. 1, 2, 3, 4, 6, 7, 8

[22] G. Han, J. Ma, S. Huang, L. Chen, and S.-F. Chang, "Few-shot object detection with fully cross-transformer," in *2022 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2022, pp. 5311–5320. 1, 3, 4, 6, 7

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. 1

[24] H. Chen, Q. Wang, K. Xie, L. Lei, M. G. Lin, T. Lv, Y. Liu, and J. Luo, "SD-FSOD: Self-distillation paradigm via distribution calibration for few-shot object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023. 1, 3, 6, 7, 8

[25] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. 2, 3

[26] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. 2, 3

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2016, pp. 779–788. 2

[28] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *2017 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2017, pp. 6517–6525. 2

[29] ——, "YOLOv3: An incremental improvement," Apr. 2018. 2

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE_J_PAMI*, vol. 42, no. 2, pp. 318–327, 2020. 2

[31] A. Bochkovskiy, C.-Y. Wang, and H.-y. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Apr. 2020. 2

[32] R. Girshick, "Fast R-CNN," in *2015 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448. 2

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE_J_PAMI*, vol. 39, no. 6, pp. 1137–1149, 2017. 2, 4, 6, 10

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988. 2

[35] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *2019 IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9626–9635. 2

[36] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Comput. Vis. – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer Int. Publishing, 2018, pp. 765–781. 2

[37] M. Zand, A. Etemad, and M. Greenspan, "ObjectBox: From centers to boxes for anchor-free object detection," in *Comput. Vis. – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 390–406. 2

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Comput. Vis. – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer Int. Publishing, 2020, pp. 213–229. 2

[39] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive Classifier–Predictor for few-shot learning," *IEEE_J_NNLS*, vol. 32, no. 8, pp. 3458–3470, 2021. 3

[40] C. Simon, P. Koniusz, and M. Harandi, "Meta-learning for multi-label few-shot classification," in *2022 IEEE Winter Conf. Appl. of Comput. Vis. (WACV)*, 2022, pp. 346–355. 3

[41] P. Tian, W. Li, and Y. Gao, "Consistent meta-regularization for better meta-knowledge in few-shot learning," *IEEE_J_NNLS*, vol. 33, no. 12, pp. 7277–7288, 2022. 3

[42] L. Fan, C. Zeng, H. Liu, J. Liu, Y. Li, and D. Cao, "Sea-net:visual cognition-enabled sample and embedding adaptive network for SAR image object classification," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2023. 3

[43] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learning*, ser. Proc. Mach. Learning Res., D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017-08-06/2017-08-11, pp. 1126–1135. 3

[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *2018 IEEE Conf. Comput. Vis. and Pattern Recog.*, 2018, pp. 1199–1208. 3

[45] G. Cheng, C. Lang, and J. Han, "Holistic prototype activation for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4650–4666, 2023. 3

[46] C. Lang, G. Cheng, B. Tu, and J. Han, "Few-Shot Segmentation via Divide-and-Conquer Proxies," *International Journal of Computer Vision*, vol. 132, no. 1, pp. 261–283, Jan. 2024. 3

[47] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1050–1065, 2022. 3

[48] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 8057–8067. 3

[49] S. Wu, W. Pei, D. Mei, F. Chen, J. Tian, and G. Lu, "Multi-faceted distillation of base-novel commonality for few-shot object detection," in *Comput. Vis. – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 578–594. 3, 6, 7, 8, 9

[50] C. Lang, J. Wang, G. Cheng, B. Tu, and J. Han, "Progressive parsing and commonality distillation for few-shot remote sensing segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–10, 2023. 3

[51] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *2020 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2020, pp. 4012–4021. 6, 7

[52] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *2021 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2021, pp. 7348–7358. 6, 7, 8

[53] Q. Fan, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with model calibration," in *Comput. Vis. – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 720–739. 6, 7

[54] X. Jiang, Z. Li, M. Tian, J. Liu, S. Yi, and D. Miao, "Few-shot object detection via improved classification features," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2023, pp. 5386–5395. 6, 7, 8

[55] J. Xu, H. Le, and D. Samaras, "Generating features with increased crop-related diversity for few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 19 713–19 722. 6, 7, 8

[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. Journal of Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. 6

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Comput. Vis. – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer Int. Publishing, 2014, pp. 740–755. 6

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recog." in *2016 IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, 2016, pp. 770–778. 6

[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recog. Challenge," *Int. Journal of Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. 6

[60] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Inf. Process. Syst.*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. 7

[61] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. 7

[62] G. A. Miller, "WordNet: A lexical database for english," *Communications of The Acm*, vol. 38, no. 11, pp. 39–41, Nov. 1995. 7

[63] K. Kavitha, B. Sandhya, and B. T. Rao, "Evaluation of distance measures for feature based image registration using AlexNet," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018. 9

[64] S. Gkelios, A. Sophokleous, S. Plakias, Y. Boutalis, and S. A. Chatzichristofis, "Deep convolutional features for image retrieval," *Expert Systems with Applications*, vol. 177, p. 114940, 2021. 9

[65] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Mach. Learning Res.*, vol. 9, no. 86, pp. 2579–2605, 2008. 11

Ding Sheng Ong received the B.S. degree from the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, in 2020. He is currently a Ph.D. student at Aberystwyth University, UK. His research interests are in few-shot learning in computer vision, particularly object detection.

Yi Liu is currently a Full Professor at Changzhou University, China. He received the Ph.D. degree from Xidian University, China, in 2019. From 2018 to 2019, he was a Visiting Scholar at Lancaster University. His research interests include machine learning and computer vision, especially on saliency detection, capsule networks, 3D point cloud, and object detection.

Jungong Han is Chair Professor in Computer Vision at the Department of Computer Science, University of Sheffield, U.K. He also holds the honorary professorships with Aberystwyth University and the University of Warwick, U.K. His research interests include computer vision, artificial intelligence, and machine learning. He is the Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, and he is the Fellow of IAPR.