Toward Generalized Few-Shot Open-Set Object Detection

Binyi Su, Hua Zhang, Jingzhi Li, and Zhong Zhou

Abstract— Open-set object detection (OSOD) aims to detect the known categories and reject unknown objects in a dynamic world, which has achieved significant attention. However, previous approaches only consider this problem in data-abundant conditions, while neglecting the few-shot scenes. In this paper, we seek a solution for the generalized few-shot open-set object detection (G-FOOD), which aims to avoid detecting unknown classes as known classes with a high confidence score while maintaining the performance of few-shot detection. The main challenge for this task is that few training samples induce the model to overfit on the known classes, resulting in a poor open-set performance. We propose a new G-FOOD algorithm to tackle this issue, named Few-sh<u>O</u>t <u>O</u>pen-set <u>D</u>etector (FOOD), which contains a novel class weight sparsification classifier (CWSC) and a novel unknown decoupling learner (UDL). To prevent overfitting, CWSC randomly sparses parts of the normalized weights for the logit prediction of all classes, and then decreases the co-adaptability between the class and its neighbors. Alongside, UDL decouples training the unknown class and enables the model to form a compact unknown decision boundary. Thus, the unknown objects can be identified with a confidence probability without any threshold, prototype, or generation. We compare our method with several state-of-the-art OSOD methods in few-shot scenes and observe that our method improves the F-score of unknown classes by 4.80%-9.08% across all shots in VOC-COCO dataset settings. Source code is available on-line at https://github.com/binyisu/food.

Index Terms— Generalized few-shot open-set object detection, class weight sparsification classifier, unknown decoupling learner.

I. INTRODUCTION

OBJECT detection is a fundamental task in the field of computer vision, which involves not only recognizing objects but also determining their precise locations by drawing

bounding boxes around them. With the assistance of deep learning, object detection has achieved remarkable progress. However, existing object detection models [1], [2], [3] are under a strong assumption that there exist enough samples for all the categories, which is time-consuming and expensive to annotate instances for supervised training.

To alleviate this issue, few-shot object detection (FSOD) methods [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] are developed to reduce the data dependence of the CNN models. FSOD aims to train a detector based on a few samples. Various approaches have presented significant improvements in the FSOD problem. However, these methods hold a closed-set assumption, where the training and testing sets share the same classes. In open-set situations, there are countless unknown classes, not included in the training set. These unknown objects can easily disrupt the rhythm of the closed-set models, causing them to identify the unknown classes as known ones with a high confidence score [15].

To make the model better handle the open-set scenarios, open-set object detection (OSOD) [15], [17], [18] has been constantly investigated, where the detector trained on the closed-set datasets is asked to detect all known classes and reject unknown classes in open-set conditions. These OSOD methods leverage good representations of the known classes with sufficient training samples to construct unknown-aware detectors. However, open-set detectors suffer from a serious over-fitting problem [16] with few known training samples, which greatly degrades the performance of open-set detection.

In this paper, we seek a solution for the unexplored generalized few-shot open-set object detection (G-FOOD) problem, which commits to training a detector for unknown rejection using few samples. The concept of G-FOOD is expressed in Fig. 1. G-FOOD has enormous value in safety-critical applications such as autonomous driving and medical analysis. For example, in autonomous driving scenarios, encountering data-hungry known classes such as rare animals and unknown objects such as unexpected obstacles can potentially pose high risks to the safe operation of autonomous vehicles. Typically, the car needs to detect all known classes (including data-abundant classes and data-hungry classes) and reject the countless unknown objects as an "unknown" class in dynamic scenes. There are three irresistible reasons to solve the G-FOOD problem. First, an open-set detector that can identify the few-shot classes is more useful than the one that does not. Second, data-abundant open-set detection is a challenge in all settings. However, few-shot open-set detection
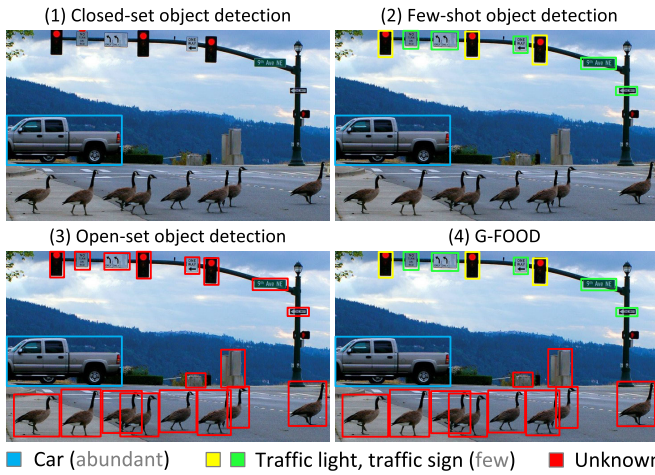
Fig. 1. The visualization of different tasks: closed-set object detection (CSOD), few-shot object detection (FSOD), open-set object detection (OSOD), and generalized few-shot open-set object detection (G-FOOD). In CSOD and FSOD tasks, unknown objects are ignored or incorrectly classified into the set of known classes. The OSOD task can reject unknown class, but it usually requires data-abundant known classes for training [16]. Our G-FOOD task can identify the data-abundant and data-hungry known objects while rejecting unknown objects based on limited training data, which provides a better open-scene understanding paradigm.

is harder than the data-abundant open-set detection [19]. With limited training data, there is a higher chance that the available few-shot examples may not fully represent the intra-class variations and complexities present in the known classes, causing serious overfitting problems. This can lead to poor generalization and difficulty in distinguishing known classes from unknown during inference. Third, like open-set detection, the main challenge of few-shot detection is to make accurate decisions for the unknown data during inference. Once the few-shot detectors have the ability for unknown rejection, they are likely to be more robust for real-world applications.

Actually, the dataset in real-world scenes often exhibits long-tail distributions [20] and there are always some unexpected categories that are not included in the dataset. G-FOOD aims to use this unbalanced dataset to train a detector that can identify all known classes and reject the countless unknown classes. G-FOOD could be viewed as an extension of the few-shot open-set recognition (FSOSR) [16], [19], [21], [22], which focuses on assigning a single label to an entire image without providing location information. However, our method is not to migrate the methods from FSOSR to G-FOOD. For example, the previous FSOSR studies adopt pseudo-unknown sample generation methods [19], [21] or prototype-based methods [16], [22] to identify the unknown classes, however, our method is independent of the pseudo-unknown sample generation or the prototypes that denote the average feature of one class. Furthermore, the scarcity of training data induces the model to easily overfit on the few-shot known classes, making it challenging for the model to generalize and accurately reject instances of the unknown classes in real-world scenarios [16]. Therefore, how to solve the over-fitting problem without degrading the performance of few-shot known classes during unknown rejection becomes our main intention.

We know that dropout [23] suppresses over-fitting by reducing the co-adaptations between neurons. Thus, we draw inspiration from the consensus that reduction of the co-adaptability between the class and its neighbors can effectively suppress the over-fitting problem. We propose to reject the unknown classes by decoupling the co-adaptations between the known and unknown classes from two aspects: 1) The optimization process for the unknown class does not consider the interactions with the known classes, decoupling training it; 2) The classifier randomly sparses parts of the normalized weights for the class logit prediction, and then decreases the co-adaptability between the class and its neighbors.

To this end, we propose a novel generalized few-shot open-set object detector, named FOOD, which is threshold-free, prototype-free, and generation-free to reject unknown objects. This means that our approach does not rely on fixed thresholds or specific example representations of known classes, and avoids potential errors or biases arising from the pseudo-unknown sample generation process. Innovatively, FOOD employs Faster R-CNN [1] as the base detector, where we replace the original classifier with a novel class weight sparsification classifier (CWSC) and additionally plug a novel unknown decoupling learner (UDL). These two modules cooperate with each other to solve the over-fitting problem of G-FOOD. After good optimization, our model can achieve the best G-FOOD performance than other state-of-the-art (SOTA) methods. Our method is also compatible with various backbones, including transformer-based architectures [24], which can leverage the advantages of different backbone models, such as their ability to capture long-range dependencies and semantic relationships. The main contributions are threefold:

- We define a new problem, called generalized few-shot open-set object detection (G-FOOD), which aims to quickly train a detector based on a few labeled samples while rejecting all detected unknown objects.
- We propose a novel G-FOOD algorithm (FOOD) with two well-designed modules: CWSC and UDL, which can improve the model's generalization ability for unknown rejection in few-shot scenes.
- We developed the first G-FOOD benchmark. We modify several state-of-the-art OSOD methods into G-FOOD methods. Compared with these methods, our FOOD improves the F-score of unknown classes by 4.80%-9.08% across all shots in VOC-COCO dataset settings.

This paper is organized as follows: Section II shows an overview of the related works. Section III introduces the proposed method. Section IV presents the extensive experiments. Finally, Section V concludes the paper.

II. RELATED WORK

A. Few-Shot Object Detection

Since the conventional detectors based on supervised learning require abundant annotated samples for training, few-shot object detection (FSOD) has received significant progress recently [4], [5], [6], [7], [8], [11], [12], [25], [26], [27], [28]. FSOD can be roughly divided into three types. 1) Meta

learning-based methods. This type of works aims to learn the task-level knowledge that can be adapted to the new task with few support samples, such as FSRW [25], Meta-RCNN [26], FSOD [27], and Meta-DETR [28]. 2) Transfer-learning based approaches. This line of works adopts a simple two-stage fine-tuning strategy to train the detector, *i.e.*, base-training and few-shot fine-tuning phases, which expects to transfer the general knowledge learned from the base-training phase to the few-shot fine-tuning phase, such as TFA [4], FSCE [5], and DeFRCN [11]. 3) Pseudo-sample generation approaches. This form of works views FSOD as a data unbalanced problem, they employ the data-augmentation technologies to generate the samples of few-shot classes and train the detector end-to-end, such as [12].

Although these FSOD methods focus on good detection performance under the closed-set settings, the unknown rejection capability is not guaranteed. We extend a simple FSOD method TFA [4] with various OSOD methods to identify the unknown objects and find that our method can better reject the unknown classes in few-shot scenes than other SOTA methods.

B. Open-Set Object Detection

Open-set object detection (OSOD) methods intend to detect all known classes and reject the unknown classes, simultaneously. According to the acquisition way of the unknown samples, OSOD can be mainly divided into three categories. 1) Virtual unknown sample generation. This type of methods synthesizes the virtual unknown samples to train the unknown branch in the feature space [29] or image space [30]. 2) Select unknown samples from the background. This kind of works selects the background boxes with high uncertainty scores as the unknown class to train the open-set detector, such as ORE [31], UC-OWOD [32], ROWOD [33], and OSODD [34]. 3) Select unknown samples from the known classes. This form of works chooses the known samples with high uncertainty scores as the unknown class to train the open-set detector, such as PROSER [35] and OpenDet [15]. Moreover, there exist several threshold-based methods. This type of works uses the energy or entropy as the uncertainty score, which is compared with a threshold to reject the unknown class, such as OS [17], DS [18], MCSSD [36], and GMM-Det [37]. There are also several methods aiming at detecting all open-world objects without classifying them, such as OLN [38] and LEDT [39].

The previous methods of OSOD usually need abundant samples of the known classes to train the model. However, this cannot be satisfied in the few-shot conditions, which causes a serious over-fitting problem of the model to the few-shot known classes, resulting in a poor open-set performance. Inspired by the dropout [23], reduction of the neuron co-adaptations in optimization can efficiently suppress the over-fitting issue, we propose a class weight sparsification classifier and an unknown decoupling learner to dilute dependencies between all known and unknown classes.

C. Few-Shot Open-Set Recognition

Few-shot open-set recognition (FSOSR) has fascinated scant attention recently. However, to the best of our knowledge, generalized few-shot open-set object detection is still

TABLE I
DISTINCTIONS BETWEEN DIFFERENT TASKS

Task	Few-shot training	Known detection	Unknown detection/ rejection	Unknown discovery	Incremental learning	Cross-distribution evaluation
FSOD [4]	Yes	Yes	No	No	No	Not required
OD [40]	No	No	Yes	Yes	No	Not required
OWOD [31]	No	Yes	Yes	No	Yes	Not required
OOD [29]	No	Yes	Yes	No	No	Required
OSOD [15]	No	Yes	Yes	No	No	Not required
OSODD [34]	No	Yes	Yes	Yes	No	Not required
G-FOOD	Yes	Yes	Yes	No	No	Not required

not exploited. Here, we present several FSOSR works. PEELER [19] utilizes the pseudo-unseen class samples generated from seen classes to train the model. SnaTCHer [16] measures the distance between the query and the transformed prototype, then a distance threshold is set to identify the unseen classes. R3CBAM [21] leverages the outlier calibration network to recognize the objects in FSOSR scenes. SEMAN-G [22] learns an unseen prototype that automatically estimates a task-adaptive threshold for unseen rejection. Different from FSOSR, G-FOOD is indeed a more challenging task, because it involves not only identifying known and unknown object classes but also accurately localizing them in the image. Simultaneously, G-FOOD also has a background class, which often confuses the detector.

D. Generalized Few-Shot Open-Set Object Detection

As illustrated in Table I, we analyze the distinctions between our G-FOOD task and other related tasks including few-shot object detection (FSOD) [4], object discovery (OD) [40], open-world object detection (OWOD) [31], out-of-distribution detection (OOD) [29], open-set object detection (OSOD) [15], open-set object detection and discovery (OSODD) [34]. The first indicator (few-shot training) refers to the training of deep learning models using a limited amount of labeled examples. The second (known detection) refers to the ability of a model to identify instances that belong to known classes. The third (unknown detection/rejection) denotes that the model could identify instances that do not belong to any of the known classes or reject instances deemed as unknown. The fourth (unknown discovery) denotes discovering the categories of unknown objects with multiple classes in an unsupervised manner [40]. The fifth (incremental learning) aims to purposefully learn unknown objects in an incremental fashion [31]. The sixth (cross-distribution evaluation) involves testing the model's performance on distributions that were not part of its training data [29]. Our G-FOOD is characterized by few-shot training data, known and unknown detection, without unknown discovery and incremental learning, and not require cross-distribution evaluation, which is first explored in our paper.

III. PROPOSED METHOD

A. Problem Setup

We define the problem setup with reference to TFA [4] and OpenDet [15]. We are given an object detection dataset $D = \{(x, y), x \in \mathbf{X}, y \in \mathbf{Y}\}$, where x denotes an input image and $y = \{(c_i, b_i)\}_{i=1}^{I}$ represents the objects with its class c and its box annotation b. The dataset D is divided into the training
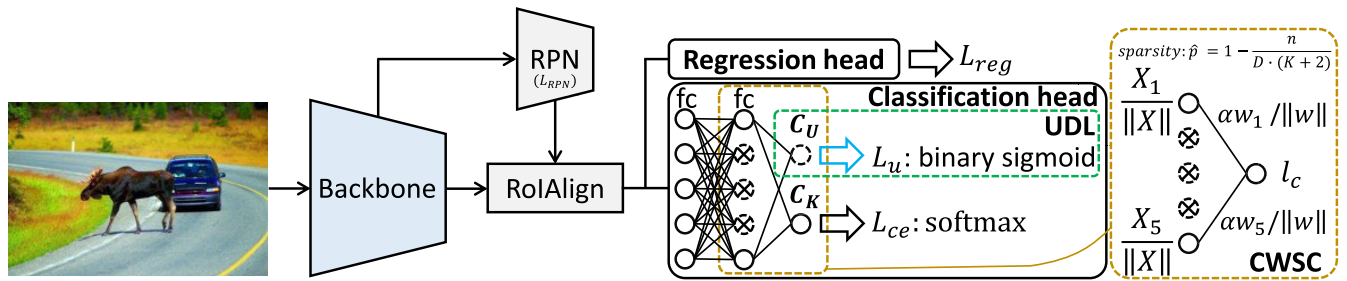
Fig. 2.   The framework of our FOOD for generalized few-shot open-set object detection. Compared to the standard Faster R-CNN, FOOD plugs a novel class weight sparsification classifier (CWSC) and a novel unknown decoupling learner (UDL). We sparsity the normalized weights for the class logit prediction and simultaneously optimize a binary sigmoid classifier and a multiply softmax classifier in the classification head. Our method is characterized by no pseudo-unknown sample generation, prototype-free, and threshold-free to reject unknowns in few-shot scenes.

set D_{tr} and the testing set D_{te}. $D_{tr} = D_B \cup D_N$ contains K known classes $C_K = C_B \cup C_N = \{1, \ldots, K = B + N\}$, where $C_B = \{1, \ldots, B\}$ expresses B data-abundant base classes, and $C_N = \{B + 1, \ldots, K\}$ denotes N data-hungry novel classes, each with M-shot support samples. Here, novel classes express few-shot known classes rather than unknown classes defined in open-set communities [15], [18]. D_B and D_N denote the training instances of the base and novel classes, respectively. We test the detector in D_{te} that includes $C_K = C_B \cup C_N$ known classes and C_U unknown classes. Duo to the countless unknown categories, we merge all of them into one class $C_U = \{K + 1\}$. Our goal is to employ the unbalanced data $D_{tr} = D_B \cup D_N$ to train a detector, which can be used to identify the base classes C_B, the novel classes C_N, the unknown class C_U, and the background class C_{bg}. Here, regions that do not contain any annotated objects of interest effectively serve as training samples for the background class. This is pivotal for enabling the model to explicitly identify regions that do not align with any specific object category (base, novel, or unknown).

B. Baseline Setup

As shown in Fig. 2, Faster R-CNN [1] is adopted as the base detector that is composed of a backbone, a region proposal network (RPN), a region of interest alignment (RoIAlign) layer, plus two heads (a regression head [1] and a classification head). Compared with the standard Faster R-CNN, three tricks are utilized to improve the detector. (1) **Classification and regression decoupling**: The original two heads contain two shared fully connected (fc) layers and two separate fc layers for classification and regression. In order to prevent the classification task from disturbing the regression task, the shared fc layers are replaced by two parallel fc layers. Simultaneously, the class-specific box regression is changed to class-agnostic. For example, the standard output of the box predictor (box regression head) is $4 \times (K + 2)$, now we set it as 4, where $K + 2$ denotes K known classes, 1 unknown class, and 1 background class. It means that for each region proposal, we predict one box for all classes, instead of one box per class. The above operations are utilized to decouple the classification and the regression, and then provide convenience to tackle the generalized few-shot open-set object detection task.

(2) **Two-stage fine-tuning strategy**: Following TFA [4], the training process consists of a base training stage and a few-shot fine-tuning stage. In the base training stage, we employ

abundant samples of the base classes C_B to train the entire base detector from scratch, such as Faster R-CNN. Then, in the few-shot fine-tuning stage, we create a small balanced training set with M shots per class, containing both base and novel classes. The balanced dataset is first used to train the last linear layers of the base detector while freezing the other parameters of the model (linear probing [41]), and then fine-tune all the parameters of the model (fine-tuning) in a soft-freezing way [11], which employs a scaled gradient to slowly update the parameters of the backbone network to get the generalized few-shot open-set object detector. Note that there are two layers for linear probing including the last box classification layer and the last box regression layer, namely the last linear layers of the base detector.

(3) **Classifier placeholder**: Classifier placeholders [35] refer to variables or symbols that represent the preserved classes in the classification head. In the base training stage, we reserve the dummy classifier placeholders for the novel classes C_N and an unknown class C_U to augment the class number of the open-set classifier. These placeholders reserved for the novel classes will be optimized in the few-shot fine-tuning stage. Overall, there are two advantages for the predefined classifier placeholder: one is that the classifier placeholder omits the additional model surgery step [4], [11], which is used to augment the number of model categories from the base training stage (base classes C_B) to the few-shot fine-tuning stage (base classes C_B + novel classes C_N) for the closed-set classifier. This means that our method simplifies the training process of the FSOD methods based on transfer learning. Another is that the dummy sub-classifier for the unknown class is necessary to optimize our proposed unknown decoupling learner (UDL), which can reject the unknown objects without relying on the pseudo-unknown samples for training.

C. Class Weight Sparsification Classifier (CWSC)

Few training samples induce the model to overfit on the known classes, thus the model cannot extract the generalization features that can be used for the unknown rejection. The neuron dropout theory [23] has been employed to suppress the over-fitting issue. The dropout randomly drops some weights during training. During the prediction process, if we still randomly drop some weights, the model will typically produce unstable results and hurt the detection performance. However, suppose all weights are reserved for prediction at

test time. In that case, it will result in a large expected (or mean) difference in the output between training (with dropout) and testing (without dropout), leading to performance degradation. So we need to make the output expected value of training and testing as consistent as possible. Our CWSC can be understood as class weight sparsification for unknown rejection in few-shot scenes. As shown in Fig. 2, our CWSC sparses the normalized weights $w/||w||$ rather than w and measures the cosine similarity with the normalized feature $X/||X||$ for the class logit prediction, where $|| \cdot ||$ denotes the L2 norm. This makes the expected difference with and without sparsification become bounded, and then the model is more stable and prominent for open-set detection in few-shot scenes.

Specifically, $w \in \mathbb{R}^{D \times (K+2)}$ represents the weights of the last linear mapping to $(K + 2)$ classes, where D denotes the weight dimension for each class. We use the sparsity probability $\widehat{p} = 1 - \frac{n}{D \cdot (K+2)}$, where n denotes the number of randomly selected weights for the class logit prediction. Then, the retention probability of the weight can be denoted as $p = 1 - \widehat{p}$. The weights can be denoted as $R * w/||w||$, where $R \in \{0, 1\}^{D \times (K+2)}$ is initialized with $R_{i,j} \sim$ Bernoulli(p) and $*$ represents the element-wise product. The total number of 1 in R is n, where 1 denotes retaining the weight and 0 denotes dropping the weight. The class logit output could be expressed as:

$$l_c = \alpha \frac{X}{||X||} \left[R * \frac{w}{||w||} \right], \quad (1)$$

where α denotes a positive temperature factor. In our CWSC, if we set the retention probability p, the expected value (or mean) of the logit output l_c with sparsification falls in $[-p\alpha, p\alpha]$, without sparsification, it falls in $[-\alpha, \alpha]$, thus the expected difference with and without sparsification is $\mu_{diff}^{cos} \in [-(1 + p)\alpha, (1 + p)\alpha]$. For the conventional output $\widehat{l}_c = Xw$, the expected difference of the logit output with and without sparsification is $\mu_{diff} \in (-\infty, +\infty)$. Compared with μ_{diff}, μ_{diff}^{cos} is bounded, which means the output distribution of our CWSC with and without sparsification is more consistent, producing better results for G-FOOD.

Next, we theoretically analyze why the CWSC can suppress the over-fitting issue. Assuming a linear regression task, $y \in \mathbb{R}^N$ is the ground truth label, the model tries to find a $w \in \mathbb{R}^D$ to minimize

$$||y - \alpha \frac{X}{||X||} \frac{w}{||w||}||^2. \quad (2)$$

We set $\bar{x} = X/||X||$ and $\bar{w} = w/||w||$. When the weight sparsification is adopted, the objective function becomes

$$\underset{w}{minimize}\ \mathbb{E}_{R \sim \text{Bernoulli}(p)}[||y - \alpha\bar{x}(R * \bar{w})||^2]. \quad (3)$$

This can reduce to

$$\underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 + \alpha^2 p(1-p)||\tau\bar{w}||^2, \quad (4)$$

where $\tau = (diag(\bar{x}^T\bar{x}))^{1/2}$. We set $\widetilde{w} = \alpha p\bar{w}$, then

$$\underset{w}{minimize}||y - \bar{x}\widetilde{w}||^2 + \frac{(1-p)}{p}||\tau\widetilde{w}||^2. \quad (5)$$

Eq. 5 can be viewed as a ridge regression with a particular form for τ. If a particular data dimension changes a lot, the regularizer tries to sparse its weight more [23], and then the over-fitting problem is alleviated through regularization. We can control the strength of the regularization by adjusting the sparsity probability $\widehat{p} = 1 - p$. For example, if we set $\widehat{p} = 0$, then $p = 1$, the regularization term is 0, which means that the regularizer does not work. As the sparsity probability $\widehat{p}$ increases, the regularization constant grows larger and the regularization effect becomes more pronounced. Our method benefits from the over-fitting suppression, and we verify that the class weight sparsification greatly improves the model's generalization ability for the open-set detection in few-shot scenes. A derivation of Eq. 2-5 is presented in Appendix.

D. Unknown Decoupling Learner (UDL)

Unknown decoupling learner (UDL) plays a decisive role in rejecting the unknown class, which provides a dummy unknown class for decoupling optimization, and then boosts the model to form a compact unknown decision boundary. The UDL does not depend on the pseudo-unknown samples generation method [19] to train the dummy unknown class, because the data distribution of the unknown class is more complex and changeable, the generated fake unknown samples often fail to simulate the real distribution of the unknown data. Inspired by the fact that the known class data and the unknown class data are often orthogonal [41], we propose to decouple optimizing the unknown class without relying on the predictions of the known classes. Thus, we select a sigmoid function that normalizes the predicted unknown logit to estimate the unknown probability:

$$p_u(l_{C_U}) = \frac{1}{1 + e^{-\delta \cdot l_{C_U}}}, \quad (6)$$

where l_{C_U} represents the predicted logit for the dummy unknown class C_U in the classification branch and δ is used to adjust the slope of the sigmoid function. Why do we choose the sigmoid function to compute the unknown probability p_u instead of softmax? If we select the softmax, the objective function for the unknown branch becomes:

$$L_u \downarrow = -\log p_u \uparrow = -\log \frac{e^{l_{C_U} \uparrow}}{\sum\limits_{c_i \in C, c_i \neq c_*} e^{l_{c_i}} + e^{l_{c_*} \uparrow}} \quad (7)$$

where $C = C_B \cup C_N \cup C_U \cup C_{bg}$, $\uparrow$ denotes that the optimized target should be as large as possible in optimization and $\downarrow$ represents the opposite. There are no real unknown samples or groundtruths in the training process. The positive pseudo-unknown samples used in this method are high conditional energy (see Eq. 9) samples selected from known classes or backgrounds. In addition to the groundtruth of an unknown class, each pseudo-unknown sample also has its own real known class or background label. During the unknown optimization, the model may predict high logits (l_{C_U} and l_{c_*}) for the unknown class C_U and its real ground-truth class c_*, simultaneously, which causes optimization conflicts, making it difficult for the unknown class to converge. While the unknown probability product by the sigmoid function does

not care about the logit outputs for other classes (see Eq. 6), we call it unknown decouple training, which can alleviate the above problem.

Therefore, the loss function of the dummy unknown class (unknown decoupling loss) is defined as follows:

$$
\mathcal{L}_u = \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} \log(1 + e^{-\delta_1 \cdot l_{C_U}(\mathcal{S}_{pos}^i)})
$$
$$
+ \frac{1}{N_{neg}} \sum_{j=1}^{N_{neg}} \log(1 + e^{-\delta_2 \cdot l_{C_U}(\mathcal{S}_{neg}^j)}), \tag{8}
$$

where N_{pos} and N_{neg} represent the number of positive pseudo-unknown samples $\mathcal{S}_{pos}$ and $\mathcal{S}_{neg}$ selected from foreground samples of known classes and background samples, respectively. Note that negative pseudo-unknown samples are not required to optimize the sigmoid-based classifier because it assigns a binary label (0 or 1) to each proposal, indicating the presence or absence of an object in each proposal [3]. How do we select the positive pseudo-unknown samples ($\mathcal{S}_{pos}$ and $\mathcal{S}_{neg}$) from all foreground proposals $\mathcal{S}_{fg}$ and background proposals $\mathcal{S}_{bg}$ to train the UDL branch? Ideally, a model should learn a more compact decision boundary that produces low uncertainty for the known data, with high uncertainty for unknown data elsewhere. Fortunately, the energy score [29], [42] has been used to measure the uncertainty of the object, where the larger the energy, the higher the uncertainty of the sample [29]. The motivation for selecting high-energy regions as pseudo-unknown samples is rooted in the assumption that objects from unknown classes are not seen during training. Since the model does not have specific knowledge about those unknown classes, unknown classes may have low-logit predictions compared with known and background classes frequently seen during training. The lower the logit, the higher the energy (see Eq. 9). Here, we propose to select the foreground and background samples with high conditional energy as positive pseudo-unknown samples to train the UDL branch. For a region proposal $\mathcal{S}_j \in \mathcal{S}_{fg} \cup \mathcal{S}_{bg}$, the conditional energy score is defined as:

$$
E(\mathcal{S}_j)_{c_i \neq C_U} = -\log \sum_{c_i \neq C_U, c_i \in C} e^{l_{c_i}(\mathcal{S}_j)}, \tag{9}
$$

where $l_{c_i}(\cdot)$ is the predicted logit for class c_i. Since there are no real unknown class training samples, the term $exp(l_{C_U}(\mathcal{S}_j))$ will become an interference term in the energy-based sampling process. Thus, we discard it and select the top-k samples ranked by the conditional energy score $E(\mathcal{S}_j)_{c_i \neq C_U}$ to optimize the unknown decoupling loss L_u. The positive pseudo-unknown samples ($\mathcal{S}_{pos}$ and $\mathcal{S}_{neg}$) used to train the UDL branch can be denoted as:

$$
\mathcal{S}_{pos} = \underset{\mathcal{S}_{j'} \in \mathcal{S}_{fg}}{topk} \left(-\log \sum_{c_i \neq C_U, c_i \in C} e^{l_{c_i}(\mathcal{S}_{j'})} \right)_{k=N_{pos}}, \tag{10}
$$

$$
\mathcal{S}_{neg} = \underset{\mathcal{S}_{j''} \in \mathcal{S}_{bg}}{topk} \left(-\log \sum_{c_i \neq C_U, c_i \in C} e^{l_{c_i}(\mathcal{S}_{j''})} \right)_{k=N_{neg}}. \tag{11}
$$

By selecting high conditional energy proposals as pseudo-unknown samples for training, the model can focus on learning to treat high conditional energy samples as unknown classes and low conditional energy samples as known classes or the background, thus assisting in forming the unknown decision boundary of the model. Overall, the sub-classifier of the dummy unknown class in UDL is equivalent to merging the binary sigmoid classifier into the multiply softmax classifier of real known classes. Then, the model can synchronously identify the specific class of known and unknown objects in inference. Instead of asynchronously distinguishing the unknown class from the known classes like the threshold-based methods [18], [36], if it is not an unknown class, the model would distinguish its specific known class.

E. Overall Optimization

With the unknown decoupling loss L_u, the final loss function is defined as follows:

$$
\mathcal{L} = \mathcal{L}_{RPN} + \mathcal{L}_{reg} + \mathcal{L}_{ce}(D_B \cup D_N) + \lambda \mathcal{L}_u(\mathcal{S}_{pos} \cup \mathcal{S}_{neg}), \tag{12}
$$

where $\mathcal{L}_{RPN}$ is the RPN loss that consists of a binary cross-entropy loss and a regression loss. $\mathcal{L}_{reg}$ denotes the smooth L1 loss. $\mathcal{L}_{ce}$ expresses the cross-entropy loss. λ is used to balance the proportion of the unknown decoupling loss $\mathcal{L}_u$. The above four sub-losses can be used to jointly optimize the G-FOOD model f_θ as:

$$
\underset{\theta}{argmin}(\mathbb{E}_{(x,y) \in D_B \cup D_N} \mathcal{L}_{RPN,reg,ce}(x, y; f_\theta)
$$
$$
+ \mathbb{E}_{(x,y) \in \mathcal{S}_{pos} \cup \mathcal{S}_{neg}} \mathcal{L}_u(x, y; f_\theta)), \tag{13}
$$

where θ denotes the learned weights for the G-FOOD model.

F. Inference

During inference, we normalize the logits of all classes (base classes, novel classes, the unknown class, and the background class) by the softmax function to get the final classification score:

$$
p_{c_m} = \underset{c_j \in C}{max}(\frac{e^{l_{c_j}}}{\sum_{c_i \in C = C_B \cup C_N \cup C_U \cup C_{bg}} e^{l_{c_i}}}). \tag{14}
$$

The specific class of the predicted box is c_m. We not continue to choose sigmoid to compute the probability of the unknown class p_u in inference. The reason is that if we choose sigmoid to compute p_u at test time, our method would become a threshold-based method to divide the unknown class, which means we need to set a threshold to identify the unknown class. However, softmax is a threshold-free method during testing, the class with the maximum softmax probability (MSP) is the final predicted class of the box. Thus, we select it and our method becomes threshold-free for unknown rejection based on a few training samples.

IV. EXPERIMENT

A. Experimental Setup

1) Datasets: We construct the G-FOOD benchmarks using PASCAL VOC 2007+2012 [43], MS COCO 2017 [44], and LVIS [45]. There are two types of benchmark settings. One is the single-dataset benchmark: VOC10-5-5 and LVIS315-454-461, which means only one dataset (PASCAL VOC or LVIS) is used to construct the G-FOOD benchmark. Another is the cross-dataset benchmark: VOC-COCO, which means two datasets (PASCAL VOC and MS COCO) are used to construct the G-FOOD benchmark. The VOC-COCO setting can ensure that the model barely sees the unknown classes during training.

a) VOC10-5-5: The 20 classes of PASCAL VOC are divided into 10 base known classes C_B, 5 novel known classes C_N, and 5 unknown classes C_U to evaluate the G-FOOD performance of our method. The novel known classes C_N have $M = 1, 2, 3, 5, 10$, and 30 objects per class sampled from the training data of PASCAL VOC. Here we select the test set of PASCAL VOC 2007 for the generalized few-shot open-set evaluation. In this paper, we define the above dataset divisions as the **VOC10-5-5** settings, where C_B={aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow}, C_N={diningtable, dog, horse, motorbike, person}, C_U={pottedplant, sheep, sofa, train, tvmonitor}={unknown}.

b) VOC-COCO: The MS COCO dataset contains 80 classes, 20 of which overlap with the PASCAL VOC dataset. We use 20 classes of PASCAL VOC and 20 non-VOC classes of MS COCO as the closed-set training data, where PASCAL VOC servers as the base known classes C_B and the 20 non-VOC classes of MS COCO are the few-shot splits of novel known classes C_N. The novel known classes C_N have $M = 1, 2, 3, 5, 10$, and 30 objects per class sampled from the training data of MS COCO. The remaining 40 classes of MS COCO are used as the unknown classes C_U, which are challenging. Meanwhile, we use the validation set of MS COCO for the generalized few-shot open-set evaluation. In this paper, we define the above dataset divisions as the **VOC-COCO** settings, where C_B={aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor}, C_N={truck, traffic light, fire hydrant, stop sign, parking meter, bench, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, microwave, oven, toaster, sink, refrigerator}, C_U={frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake, bed, toilet, laptop, mouse, remote, keyboard, cell phone, book, clock, vase, scissors, teddy bear, hair drier, toothbrush, wine glass, cup, fork, knife, spoon, bowl}={unknown}.

c) LVIS315-454-461: LVIS dataset has a natural long-tail distribution, the classes of which are divided into 315 frequent classes (appearing in more than 100 images), 461 common classes (10)−100 images), and 454 rare classes (less than 10 images). Here, we use frequent classes, rare classes, and common classes as base known classes, novel known classes, and unknown classes, respectively. It does not need the manual M-shot split, where the rare classes can directly serve as the

few-shot training data. Meanwhile, we use the validation set of LVIS for the generalized few-shot open-set evaluation.

2) Evaluation Metrics: The mean average precision (mAP) of known classes (mAP_K) is chosen to evaluate the known object detection performance. mAP_B and mAP_N are used to measure the performance for base and novel classes, respectively. To evaluate the unknown detection performance, the average precision (AP_U), precision (P_U), and recall (R_U) are reported. The unknown recall (R_U) is a popular metric that is currently concerned by unknown detection [46].

Precision and recall only focus on one aspect of performance, while F-score can be considered comprehensively for unknown evaluation,

$$F_U = (1 + \beta^2) \frac{P_U \times R_U}{\beta^2 \times P_U + R_U}, \qquad (15)$$

where the hyperparameter β (the default value is 10 in this paper) indicates the importance of recall relative to precision in the unknown performance evaluation. The calculations of precision and recall rely heavily on labeled data. However, the testing dataset contains many unknown classes $C_U^* \notin C_U$ that may be detected but not labeled, thus causing evaluation bias. We supply Wilderness Impact (**WI**) [15] and Absolute Open-Set Error (**AOSE**) [15] to mitigate the evaluation bias of these metrics. We use WI to measure the degree of unknown objects misclassified to known classes. We also use AOSE to count the number of misclassified unknown objects. Note that the smaller the WI and AOSE values, the better the open-set performance.

3) Implementation Details: Our base detector is Faster R-CNN and ResNet-50 with feature pyramid network (FPN) [47] is selected as the backbone. All models are trained using an SGD optimizer with a mini-batch size of 16, a momentum of 0.9, and a weight decay of 1×10^{-4}. The learning rate of 0.02 is used in the base training stage and 0.01 in the few-shot fine-tuning stage. For the CWSC, we set the temperature factor $\alpha = 20$ [4], the sparsity probability $\hat{p} = 0.6$, and the class weight dimension $D = 2048$ [4]. For the UDL, we use a slope factor $\delta_1 = \delta_2 = 0.09$ and $N_{pos} : N_{neg} = 3 : 12$. In the total loss, we set the trade-off factor $\lambda = 1.0$. Note that both the base training stage and the few-shot fine-tuning stage need to optimize the UDL branch. Then we argue that the generalization knowledge learned by the base training stage between the known and unknown classes can be reserved for the few-shot fine-tuning stage. CWSC is only used in the fine-tuning stage.

4) Baselines: We compare our proposed FOOD with the following methods: OpenDet [15], DS [18], ORE [31], and PROSER [35] combined with TFA [4] for G-FOOD. For dropout sampling (DS) [18], the detector is trained with a dropout layer inserted after the second fc layers and we enable it to perform 30 samplings during testing. For PROSER [35], we remove the proposal's probability of the ground truth label as the masked probability and match the masked probability with the unknown class $C_U = \{K + 1\}$, forcing the dummy unknown classifier to output the probability for unknown rejection. For ORE [31] and OpenDet [15], we use the official codes combined with a two-stage fine-tuning strategy

TABLE II

THE GENERALIZED FEW-SHOT OPEN-SET OBJECT DETECTION RESULTS ON **VOC10-5-5** DATASET SETTINGS. ↑ INDICATES THAT THE LARGER THE EVALUATION METRICS, THE BETTER THE PERFORMANCE. **BOLD** NUMBERS DENOTE SUPERIOR RESULTS. FOR A FAIR COMPARISON, WE REPORT THE AVERAGE RESULTS OF 10 RANDOM RUNS FOR ALL COMPARISON METHODS

VOC10-5-5	1-shot							2-shot						
	mAP_K ↑	mAP_B ↑	mAP_N ↑	AP_U ↑	P_U ↑	R_U ↑	F_U ↑	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	**45.31**	**63.71**	8.50	0.00	0.00	0.00	0.00	**46.48**	62.55	14.34	0.00	0.00	0.00	0.00
DS+TFA [18]	43.82	62.11	7.22	1.90	**7.61**	23.99	23.49	46.28	**63.20**	12.44	2.08	**8.20**	24.56	24.08
ORE+TFA [31]	43.25	61.31	8.62	1.23	5.97	18.25	17.89	44.86	61.08	12.17	1.58	5.78	22.19	21.58
PROSER+TFA [35]	41.64	58.22	8.49	3.26	5.60	30.95	29.62	42.70	57.27	13.56	3.42	5.35	31.53	30.07
OpenDet+TFA [15]	43.45	61.04	8.27	3.44	6.62	33.64	32.33	45.67	62.74	11.53	3.28	6.85	30.95	29.91
Our FOOD	43.97	61.48	**8.95**	**4.26**	3.58	**43.72**	**39.35**	45.85	61.37	**14.80**	**4.61**	3.51	**45.52**	**40.70**

VOC10-5-5	3-shot							5-shot						
	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	47.55	63.72	15.23	0.00	0.00	0.00	0.00	47.88	61.92	19.74	0.00	0.00	0.00	0.00
DS+TFA [18]	46.89	63.09	14.48	2.11	**8.32**	23.62	23.20	48.01	62.38	19.27	1.91	**8.85**	19.99	19.74
ORE+TFA [31]	45.88	62.23	14.52	1.71	5.96	22.23	21.64	46.29	62.71	18.49	1.82	5.98	23.01	22.38
PROSER+TFA [35]	43.30	57.30	15.16	3.23	5.32	32.30	30.76	45.12	57.64	20.08	3.56	5.34	32.68	31.10
OpenDet+TFA [15]	46.47	62.66	14.09	3.15	6.47	30.62	29.53	47.56	62.39	17.90	3.41	6.89	32.13	31.01
Our FOOD	**48.48**	**64.30**	**16.83**	**4.50**	3.54	**44.52**	**39.94**	**50.18**	**63.72**	**23.10**	**4.63**	3.51	**45.65**	**40.80**

VOC10-5-5	10-shot							30-shot						
	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	51.10	63.56	26.19	0.00	0.00	0.00	0.00	56.20	65.68	37.23	0.00	0.00	0.00	0.00
DS+TFA [18]	48.01	62.38	25.66	1.91	**8.85**	19.99	19.74	56.60	66.27	36.96	1.90	**9.95**	18.02	17.88
ORE+TFA [31]	48.17	62.18	25.40	1.84	6.02	23.48	22.82	56.41	65.83	35.84	1.88	6.12	23.77	23.11
PROSER+TFA [35]	48.35	59.96	25.13	3.59	5.16	32.61	30.98	53.93	62.96	35.86	3.46	4.76	33.93	31.99
OpenDet+TFA [15]	50.95	63.85	25.14	4.06	6.61	36.30	34.75	56.11	66.05	36.24	4.03	5.62	40.51	38.16
Our FOOD	**53.23**	**65.54**	**28.60**	**4.71**	3.59	**45.84**	**41.06**	**58.59**	**67.73**	**40.29**	**4.90**	3.62	**46.40**	**41.54**

(TFA [4]) for generalized few-shot open-set detection. We also present the FSOD results of TFA as a baseline to determine whether optimizing the dummy unknown class will reduce the performance of the known classes. Moreover, all methods employ the same ResNet-50 with FPN as the backbone for a fair comparison and we report the average results of 10 random runs for all comparison methods.

B. Results

1) VOC10-5-5: In Table II, we compare our FOOD with several OSOD methods combined with TFA on VOC10-5-5 dataset settings. Our FOOD outperforms other methods across all shots on the unknown metrics AP_U, R_U, and F_U. We achieve 0.65∼1.27 point improvement in AP_U over the best comparison method, around 5.89∼13.99 point improvement in R_U, and around 3.38∼10.63 point improvement in F_U. Simultaneously, our method outperforms other methods on the mAP_N of novel classes, which demonstrates its effectiveness for G-FOOD. The unknown F-score of our FOOD achieves a significant improvement over the second best, which verifies that our method alleviates the over-fitting issue and evidently improves the model's generalization ability for unknown rejection in few-shot scenes. However, in VOC10-5-5 settings, the detector may have seen unknown objects and treated them as background during training, which causes the evaluation bias for G-FOOD. To balance the above bias, we conduct experiments on VOC-COCO dataset settings, where the unknown classes are barely seen in the training data.

2) VOC-COCO: In Table III, we carry out experiments on VOC-COCO dataset settings. Only looking at the rejection performance of the unknown class (AP_U, P_U, R_U, and F_U), our method presents several significant advantages, especially

in unknown recall R_U and F-score F_U. For example, the highest F-score 22.62% of the unknown class is obtained by our FOOD, which illustrates that our method has a better unknown object rejection ability than other methods. Simultaneously, in extremely low shots (1, 2, 3, 5-shot) where the over-fitting problem is easier to occur than in high shots (10, 30-shot), our method achieves 7.96%, 9.08%, 9.02%, and 8.13% unknown F-score F_U improvements than the second best method respectively, which demonstrates the effectiveness of our method in suppressing the over-fitting issue caused by few training samples of known classes.

Compared with the baseline TFA on novel classes, the mAP_N of our method is similar to TFA. Meanwhile, the mAP_K and mAP_B of our method are higher than TFA, which verifies that our method not only improves the open-set detection performance of the unknown class, but also slightly improves the performance of known classes $C_K = C_B \cup C_N$ for the closed-set few-shot evaluation. Look at other methods, although DS+TFA, PROSER+TFA, and OpenDet+TFA achieve comparable closed-set metrics (mAP_K, mAP_B, and mAP_N), the open-set performance is poor. Overall, the proposed FOOD surpasses other methods and thus is of merit. The class weight sparsification and the unknown decoupling training effectively expand model's generalization capability for open-set detection in few-shot conditions, which brings performance improvement.

Our FOOD method often achieves greater unknown recall at a minor cost of unknown precision. With the precision and recall for unknown evaluation, it appears that these are calculated from the classes that are labeled in the respective datasets but treated as unknown classes. However, there may still be unknown objects that are not labeled in these datasets.

TABLE III

THE GENERALIZED FEW-SHOT OPEN-SET OBJECT DETECTION RESULTS ON **VOC-COCO** DATASET SETTINGS. ↑ INDICATES THAT THE LARGER THE EVALUATION METRICS, THE BETTER THE PERFORMANCE. **BOLD** NUMBERS DENOTE SUPERIOR RESULTS. FOR A FAIR COMPARISON, WE REPORT THE AVERAGE RESULTS OF 10 RANDOM RUNS FOR ALL COMPARISON METHODS

VOC-COCO	1-shot							2-shot						
	$mAP_K \uparrow$	$mAP_B \uparrow$	$mAP_N \uparrow$	$AP_U \uparrow$	$P_U \uparrow$	$R_U \uparrow$	$F_U \uparrow$	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	15.77	29.03	**2.50**	0.00	0.00	0.00	0.00	15.82	28.01	**3.63**	0.00	0.00	0.00	0.00
DS+TFA [18]	15.47	28.84	2.11	0.48	**11.53**	3.57	3.59	16.28	29.36	3.21	0.51	**11.46**	3.77	3.80
ORE+TFA [31]	14.14	28.12	2.18	0.32	6.12	4.59	4.60	15.64	28.17	3.22	0.42	6.28	4.87	4.88
PROSER+TFA [35]	13.58	24.84	2.32	0.87	7.14	7.53	7.53	14.27	24.91	3.62	0.95	6.79	8.66	8.64
OpenDet+TFA [15]	**16.01**	**29.72**	2.29	0.86	7.20	7.24	7.24	16.39	29.52	3.27	0.96	6.84	8.97	8.94
Our FOOD	15.88	29.50	2.26	**2.00**	5.75	**15.76**	**15.49**	**16.56**	**29.55**	3.58	**2.42**	5.68	**18.42**	**18.02**

VOC-COCO	3-shot							5-shot						
	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	16.55	28.27	**4.81**	0.00	0.00	0.00	0.00	17.13	27.71	6.56	0.00	0.00	0.00	0.00
DS+TFA [18]	16.93	29.38	4.49	0.54	**12.47**	3.95	3.98	17.10	27.91	6.30	0.57	**14.15**	3.86	3.89
ORE+TFA [31]	15.74	27.41	4.23	0.52	6.49	4.91	4.92	16.21	27.32	6.29	0.55	6.61	4.99	5.00
PROSER+TFA [35]	15.07	25.53	4.62	1.21	7.27	9.20	9.18	15.67	26.95	6.40	1.30	7.65	9.59	9.57
OpenDet+TFA [15]	16.82	29.11	4.55	1.16	7.39	9.56	9.53	17.16	27.75	6.56	1.48	7.84	11.49	11.44
Our FOOD	**17.56**	**30.57**	4.56	**2.72**	6.18	**18.93**	**18.55**	**18.08**	**29.47**	**6.69**	**2.92**	6.06	**20.02**	**19.57**

VOC-COCO	10-shot							30-shot						
	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U	mAP_K	mAP_B	mAP_N	AP_U	P_U	R_U	F_U
TFA [4]	18.67	28.32	9.02	0.00	0.00	0.00	0.00	23.10	31.03	15.16	0.00	0.00	0.00	0.00
DS+TFA [18]	19.06	28.66	9.46	0.60	**15.41**	3.75	3.78	23.40	31.53	**15.27**	0.63	**15.58**	3.95	3.98
ORE+TFA [31]	17.98	28.01	8.75	0.59	6.77	5.13	5.14	23.07	30.47	15.17	0.66	6.89	5.51	5.52
PROSER+TFA [35]	17.00	25.24	8.75	1.36	7.87	10.06	10.03	21.44	28.58	14.30	1.52	7.31	12.06	11.98
OpenDet+TFA [15]	18.53	28.36	8.70	1.82	7.38	13.89	13.77	22.93	31.61	14.02	2.64	7.52	18.07	17.82
Our FOOD	**20.17**	**30.67**	9.48	**3.27**	6.48	**21.48**	**21.00**	**23.90**	**33.32**	14.47	**3.80**	6.68	**23.17**	**22.62**

TABLE IV

THE OPEN-SET PERFORMANCE (WI AND AOSE) OF DIFFERENT METHODS ON VOC10-5-5 AND VOC-COCO DATASET SETTINGS. ↓ INDICATES THAT THE SMALLER THE EVALUATION METRICS, THE BETTER THE PERFORMANCE. **BOLD** NUMBERS DENOTE SUPERIOR RESULTS. FOR A FAIR COMPARISON, WE REPORT THE AVERAGE RESULTS OF 10 RANDOM RUNS FOR ALL COMPARISON METHODS

VOC10-5-5	$WI \downarrow$						$AOSE \downarrow$					
Method	1	3	5	10	30	Mean	1	3	5	10	30	Mean
TFA [4]	10.69	10.13	9.99	9.87	8.93	9.92	1308.40	1335.40	1256.10	1267.20	1237.00	1280.82
DS+TFA [18]	9.14	9.08	8.97	8.81	8.55	8.91	772.60	969.90	990.60	1025.70	1138.50	979.46
ORE+TFA [31]	9.54	9.88	10.16	9.65	9.64	9.77	930.30	1058.70	1019.70	1063.70	1159.10	1046.30
PROSER+TFA [35]	11.15	10.45	10.65	10.29	9.62	10.43	994.60	1021.70	1009.80	956.70	996.40	995.84
OpenDet+TFA [15]	10.47	9.27	9.01	8.50	7.54	8.96	867.30	954.50	1031.50	1021.40	1103.50	995.64
Our FOOD	**6.96**	**7.83**	**7.59**	**6.99**	**6.57**	**7.19**	**598.60**	**859.00**	**908.00**	**900.20**	**923.90**	**837.94**
VOC-COCO	WI						$AOSE$					
Method	1	3	5	10	30	Mean	1	3	5	10	30	Mean
TFA [4]	10.73	10.77	11.36	11.40	10.48	10.95	1441.80	1566.70	1673.30	1732.20	2294.10	1741.62
DS+TFA [18]	9.15	9.56	9.91	10.13	9.84	9.72	711.60	1007.60	1110.10	1336.40	1892.90	1211.72
ORE+TFA [31]	12.08	11.53	12.30	11.65	11.22	11.76	1087.00	1226.00	1344.00	1463.20	1867.00	1397.44
PROSER+TFA [35]	11.68	12.46	12.56	12.47	12.00	12.23	925.30	1165.10	1165.90	1160.00	1561.60	1195.58
OpenDet+TFA [15]	9.82	9.92	9.55	9.83	9.02	9.63	690.90	1053.00	1176.90	1400.60	1818.00	1227.88
Our FOOD	**6.78**	**6.95**	**7.37**	**7.95**	**8.13**	**7.44**	**485.00**	**700.40**	**859.00**	**1099.30**	**1480.90**	**924.92**

If the method detects these, it may lead to lower precision as it will be considered a false-positive (FP) unknown. This is an inherent weakness in using these metrics with unknown classes. We report several new evaluation metrics including WI and $AOSE$ to evaluate the open-set performance of our method, which can mitigate the evaluation bias caused by AP, precision, and recall. As illustrated in Table IV, we can see that our method achieves the best mean performance of WI (7.19 and 7.44) and AOSE (837.94 and 924.92) in VOC10-5-5 and VOC-COCO dataset settings, respectively, which demonstrates that our method has a strong ability for unknown rejection. Thus, the effectiveness of our method is further verified.

3) LVIS315-454-461: LVIS dataset has the characteristics of numerous classes (1230 classes) and long-tail distribution, which is very challenging. We treat the frequent classes as base known classes, the rare classes as novel known classes, and the common classes as unknown classes. As shown in Table V, weight sparsification helps the model suppress overfitting well, so the few-shot closed-set detection can maintain good experimental results (mAP_B and mAP_N). At the same time, compared with the second-best method, our model achieves good open-set performance improvements (0.96%, 3.88%, 4.03, and 343 in AP_U, F_U, WI, and $AOSE$, respectively). This proves that the decoupling optimization improves the model's generalization ability for unknown rejection in

TABLE V

THE GENERALIZED FEW-SHOT OPEN-SET PERFORMANCE ON
LVIS315-454-461 DATASET SETTINGS

LVIS315-454-461	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
TFA [4]	23.08	12.24	0	0	16.78	2261.20
DS+TFA [18]	23.19	12.79	0.28	3.87	15.45	2107.50
ORE+TFA [31]	22.41	12.10	0.26	3.91	15.21	2217.30
PROSER+TFA [35]	23.07	12.48	0.57	6.27	17.75	2158.80
OpenDet+TFA [15]	23.14	12.43	0.62	6.58	14.83	1923.50
Our FOOD	**24.51**	**13.66**	**1.58**	**10.46**	**10.80**	**1580.50**

TABLE VI

THE ABLATION STUDY OF DIFFERENT MODULES (AVERAGE OF
10 RANDOM RUNS). **BOLD** NUMBERS DENOTE
SUPERIOR RESULTS

CWSC	UDL	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
Baseline (TFA)		28.32	9.02	0	0	11.40	1732.20
✓		28.97	9.44	0	0	8.97	1658.90
	✓	**30.96**	**9.58**	2.35	16.44	8.58	1240.10
✓	✓	30.67	9.48	**3.27**	**21.00**	**7.95**	**1099.30**

TABLE VII

THE ABLATION STUDY OF DIFFERENT FINE-TUNING METHODS
(AVERAGE OF 10 RANDOM RUNS). **BOLD** NUMBERS
DENOTE SUPERIOR RESULTS

Fine-tuning methods	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
LP	30.55	9.22	3.14	20.47	8.67	1188.10
FT	20.59	**10.03**	0.78	5.19	8.30	1192.00
FT+GDL	30.42	9.78	3.20	20.83	8.12	1125.20
LP-FT	30.35	9.35	3.12	20.45	8.59	1203.40
LP-FT+GDL	**30.67**	9.48	**3.27**	**21.00**	**7.95**	**1099.30**

TABLE VIII

THE ABLATION STUDY ON DIFFERENT SAMPLING METHODS (AVERAGE
OF 10 RANDOM RUNS). **BOLD** NUMBERS DENOTE SUPERIOR RESULTS

Simpling methods	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
$Minmax-probability$	30.30	9.30	3.05	20.34	10.08	1332.50
$Min(l_{C_U})$	29.39	**9.66**	2.73	17.29	10.16	1448.50
$Max(entropy)$	29.17	9.41	2.79	18.18	9.24	1573.00
$Max(E(x,b))$	30.57	9.31	3.08	20.43	8.10	1288.80
$Max(E(x,b)_{c_i \neq u})$	**30.67**	9.48	**3.27**	**21.00**	**7.95**	**1099.30**

few-shot scenes, which is meaningful for real-world long-tail distribution applications.

C. Ablation Studies

We conduct comprehensive ablation studies on the 10-shot VOC-COCO dataset setting.

1) Effectiveness of Different Modules: We perform the ablation studies on different modules (CWSC and UDL) in Table VI, where the classical FSOD framework TFA [4] is used as the baseline. It can be seen that CWSC boosts the detection performance of the base classes (AP_B) and the novel classes (AP_N) simultaneously, which demonstrates the effectiveness of CWSC in alleviating the over-fitting problem. When exploring the influence of UDL, we find that the branch of UDL improves the detection ability for the closed-set known classes in the 10-shot setting. Simultaneously, UDL is a necessary condition for the unknown rejection of our FOOD. The best unknown detection result $AP_U = 3.27\%$ is obtained by the cooperation of two modules, although the detection performance of known classes is slightly degraded compared to the UDL alone ($\downarrow 0.10\%$), this is acceptable in terms of the overall results. Compared with the third line (the linear classifier with a dropout layer+UDL) and the fourth line (CWSC+UDL), the unknown metrics including F-score, WI, and $AOSE$ improve by 4.56%, 0.63, and 140.80, respectively, which demonstrates that the unknown rejection performance benefits from our CWSC. The class weight sparsification significantly enhances the model's generalization ability for unknown rejection in few-shot scenes.

2) Effectiveness of Different Fine-Tuning Methods: In Table VII, we carefully evaluate several fine-tuning methods to pick the most appropriate way. Linear-probing (LP) and fine-tuning (FT) [41] have been widely used in transfer learning to alleviate the over-fitting problem. Gradient decoupled layer (GDL) [11] is an auxiliary fine-tuning strategy, which conducts a stop gradient for RPN and a scaled gradient (scale=0.001 in this paper) for RCNN. As a hard-freezing method, LP can preserve the general knowledge of the base training stage, thus it achieves a competitive detection result in Table VII.

However, FT fine-tunes the entire model to fit the few-shot closed-set training data. The model gradually forgets the base classes and destroys the general knowledge during the few-shot fine-tuning stage. Thus, we can see that compared FT with LP, the results of the base classes degrade by 9.96%, and the unknown class descends by 2.36%. The advantage of FT is the better performance of the novel classes mAP_N than other approaches. LP-FT exploits the advantages of LP and FT, thus it achieves a compromised G-FOOD result.

Compared with FT, FT+GDL achieves better results in mAP_K and AP_U. This indicates that GDL reserves the general knowledge by the gradient decoupling training, which uses a scaled gradient to slowly update the parameters of the backbone network. We view the GDL as a soft-freezing method, which enables the backbone to slowly fit the closed-set data while retaining the ability to extract the generalization features. GDL improves the performance of the closed-set object detection and achieves a competitive unknown rejection performance. Based on the above analysis, we adopt a **hard-soft** combination approach (LP-FT+GDL), which first trains the last linear layers of the model while freezing other parameters. And then we fine-tune the model in a soft-freezing way. As illustrated in Table VII, LP-FT+GDL outperforms other approaches in terms of mAP_B, AP_U, F_U, WI, and $AOSE$ which verifies its effectiveness.

3) Effectiveness of Different Sampling Methods: The sampling rules for the positive pseudo-unknown samples to train the UDL branch are as follows:

- *Min max-probability* [15]: the samples are sorted in ascending order by the maximum predicted value across all classes, and top-k samples are chosen.
- $Min(l_{C_U})$: the samples are sorted in ascending order by the unknown logit, and top-k samples are chosen.
- $Max(entropy)$: the samples are sorted in descending order by the entropy score, and top-k samples are chosen.
- $Max(E(x,b)_{c_i \in C_U})$: the samples are sorted in descending order by the conditional energy score. We select top-k samples for optimization.

As presented in Table VIII, the conditional energy-based sampling method $Max(E(x,b)_{c_i \neq C_U})$ outperforms other
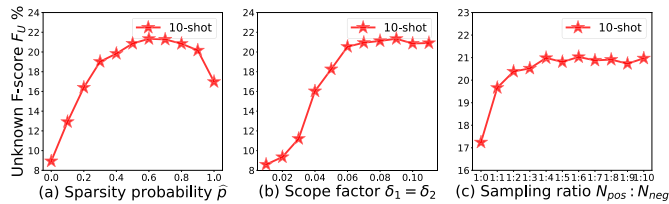
Fig. 3. Effect of different sparsity probabilities $\widehat{p}$, scope factors $\delta_1 = \delta_2$ and sampling ratios $N_{pos} : N_{neg}$ on the 10-shot VOC-COCO dataset setting.

TABLE IX

THE ABLATION STUDY OF DIFFERENT (A) BACKBONES AND (B) FSOD METHODS (AVERAGE OF 10 RANDOM RUNS). **BOLD** NUMBERS DENOTE SUPERIOR RESULTS

(a) Backbones	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
ResNet-50	30.67	9.48	3.27	21.00	7.95	1099.30
ResNet-101	31.48	10.00	3.32	21.78	7.56	1035.00
Swin-T [24]	**40.49**	**10.80**	**3.41**	**22.32**	**7.27**	**867.00**
(b) FSOD methods						
TFA-based FOOD [4]	30.67	9.48	3.27	21.00	7.95	1099.30
DeFRCN-based FOOD [11]	30.91	13.82	7.03	27.14	6.47	918.00
FSRDD-based FOOD [14]	**31.05**	**15.47**	**7.12**	**28.94**	**6.25**	**909.90**

methods, which demonstrates its effectiveness for the pseudo-unknown sample selection. Furthermore, the absence of real unknown training samples causes the unknown class C_U to become a distractor, thus the results of our $Max(E(x, b))_{c_i \neq C_U}$ are better than $Max(E(x, b))$. Simultaneously, we can see that these energy-based sampling methods ($Max(E(x, b))$ and $Max(E(x, b))_{c_i \neq C_U}$)) perform better than other sampling methods for open-set detection (WI and $AOSE$), which proves that the energy score is an excellent uncertainty metric for the pseudo-unknown sample selection in the optimization of our UDL branch.

4) Sparsity Probability, Scope Factor, and Sampling Ratio: Fig. 3 presents the visualization of different sparsity probabilities, scope factors, and sampling ratios on the 10-shot VOC-COCO dataset setting. We fix the ratio of positive and negative samples 1 : 4 to explore the performance of different sparsity probabilities. As presented in Fig. 3(a), when the sparsity probability is set to 0.6, the unknown F-score F_U arrives at 21.34%, which is higher than other settings. Subsequently, we fix $\widehat{p} = 0.6$ to iterate over different scope factors and sampling ratios, respectively. As shown in Fig. 3 (b), when the scope factor is set to 0.09, the unknown F-score is best. Therefore, we set $\delta_1 = \delta_2 = 0.09$. As we can see from Fig. 3(c), starting from a sampling ratio of 1 : 4, the value of unknown F-score begins to be stabilized. The sampling ratio of 1 : 6 seems to perform the best but our usual default value of 1 : 4 is close to the optimal. Moreover, if the pseudo-unknown samples selected from the background do not participate in the optimization of the UDL branch (1 : 0), the unknown rejection performance drops significantly.

5) Effectiveness of Different Backbones: We use ResNet-101 [48] and swin transformer (Swin-T) [24] as the backbones in Table IX (a), and then compare them with ResNet-50. ResNet-101 tends to perform better than ResNet-50, which presents that our method benefits from the deeper backbone. While employing a more powerful transformer-based backbone (Swin-T), our proposed method achieves additional improvements.
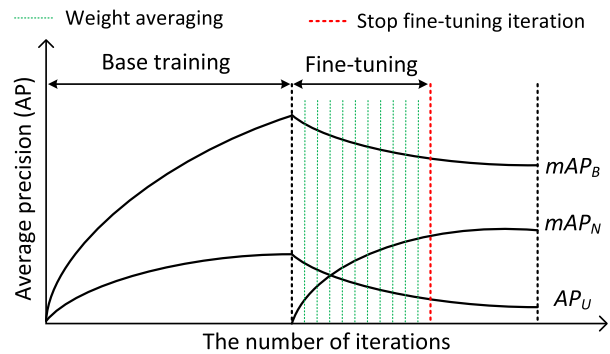

Fig. 4. The relationship between iteration and performance metrics (mAP_B, mAP_N, and AP_U) for our G-FOOD method. We are hard to select a proper stop fine-tuning iteration to balance the performance of base classes, novel classes, and the unknown class.

6) Does the Unknown-Class Placeholder Hurt the Accuracy of the Few-Shot Object Detection?: **No**. As illustrated in the first and third lines of Table VI, when the UDL branch introduces the unknown-class placeholder into the baseline TFA, the performance of novel classes has achieved 0.56% improvement, which proves that the dummy placeholder of the unknown class does not hurt the detection performance for few-shot novel classes. Furthermore, UDL and CWSC can feasibly be applied to most few-shot object detectors such as TFA [4], DeFRCN [11], and FSRDD [14]. As illustrated in Table IX (b), when incorporating UDL and CWSC into the recent FSRDD, few-shot performance and open-set performance achieve evidently improvements compared to the TFA-based few-shot open-set detection method, which demonstrates that our method is agnostic to the underlying few-shot object detection (FSOD) methods. It benefits from the high performance obtained by FSOD methods.

7) Weight Averaging: As shown in Fig. 4, the performance trends of the base classes AP_B and the unknown classes AP_U conflict with the novel classes AP_N in the fine-tuning stage of our FOOD method. Therefore, it is difficult for us to choose a suitable stop fine-tuning iteration that can make the base, novel, and unknown classes all perform best. Motivated by weight averaging (WA) [49] that is a simple ensemble method, but it achieves the state-of-the-art performance in domain generalization [50], [51], [52], We present WA to approximate the optimal model. The WA is defined as:

$$\theta_{WA}(\mathcal{L}(D_{tr} \cup \mathcal{S}_{pos} \cup \mathcal{S}_{neg}))$$
$$= \frac{1}{H+1}(\sum_{h=1}^{H}\theta_h(\mathcal{L}(D_{tr} \cup \mathcal{S}_{pos} \cup \mathcal{S}_{neg})) + \theta_{final\,model}),$$

(16)

where $\{\theta_h\}_{h=1}^{H}$ represents the weights of equidistant dense sampling in a single run and $\theta_{final\,model}$ denotes the final output weights in the above single run. WA uses the idea of ensemble learning to balance the representation bias between the base, novel, and unknown classes. As illustrated in Table X, when incorporating with WA (the model sampling step is 100 iterations), the evaluation metrics (mAP_K, mAP_B, AP_U, F_U, WI, and $AOSE$) of our FOOD achieve evidently improvements, which demonstrates its effectiveness. However,
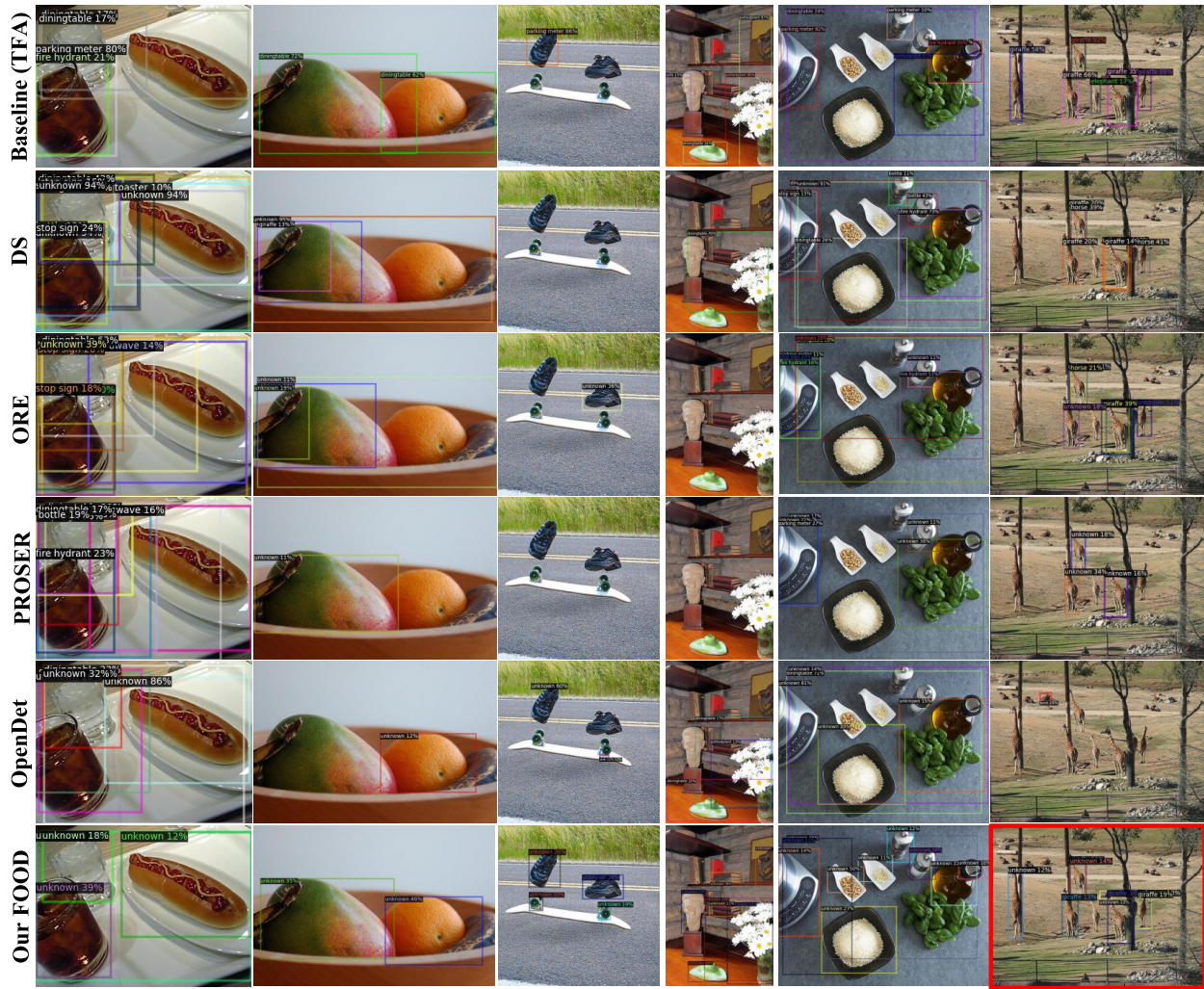
Fig. 5.　The visualization results (10-shot VOC-COCO setting). We visualize the bounding boxes with a score larger than 0.1. Our FOOD can detect more unknown objects than other methods. Red box is the failure case, several giraffes (novel class) are misidentified as the unknown class.

TABLE X

THE PERFORMANCE OF WEIGHT AVERAGING (WA) (AVERAGE OF 10 RANDOM RUNS). **BOLD** NUMBERS DENOTE SUPERIOR RESULTS

VOC-COCO	$mAP_B\uparrow$	$mAP_N\uparrow$	$AP_U\uparrow$	$F_U\uparrow$	$WI\downarrow$	$AOSE\downarrow$
Our FOOD	30.67	**9.48**	3.27	21.00	7.95	1099.30
Our FOOD+WA	**37.89**	8.76	**4.13**	**24.23**	**7.15**	**856.30**

the drawback is that WA slightly decreases the performance of the novel classes mAP_N. The main reason is that WA hurts the novel-class performance through the poor weight integration in low fine-tuning iterations. However, if you'd like to significantly improve the detection performance of the base classes and the unknown class at the expense of a little performance for novel classes, WA is a good choice.

D. Visualization

We provide qualitative visualizations of the detected unknown objects on the 10-shot VOC-COCO dataset setting in Fig. 5. We can observe that other methods easily recognize unknown objects as known classes and cannot detect unknown

objects frequently. Compared with other methods, our FOOD can achieve better unknown rejection performance, which demonstrates that our method benefits from the class weight sparsification and the unknown decoupling training. However, our method easily identifies the few-shot known objects as the unknown class. As shown in the red box of Fig. 5, several giraffes (novel class) are misidentified as the unknown class. Simultaneously, this phenomenon can also be seen from the quantitative analysis that our method shows slightly low unknown precision P_U, as illustrated in Table III. This situation is a limitation of the dummy class-based method. We argue that introducing Intersection over Union (IoU) constraints for unknown optimization may be a good solution, which will be further researched in the future.

V. CONCLUSION

In this paper, we propose a new task named generalized few-shot open-set object detection (G-FOOD) and build the first benchmark. To tackle the challenging G-FOOD task, we propose a simple method incorporating several tricks in Faster R-CNN with two novel modules: class weight sparsification

classifier (CWSC) and unknown decoupling learner (UDL). The CWSC is developed to decrease the co-adaptability between the class and its neighboring classes during the training process, thereby improving the model's generalization ability for open-set detection in few-shot scenes. Alongside, the UDL branch is employed to detect unknown objects in few-shot scenes without depending on the class prototype, threshold, and pseudo-unknown sample generation. Compared with other OSOD methods in few-shot scenes, our method achieves state-of-the-art results on different shot settings of VOC10-5-5, VOC-COCO, and LVIS315-454-461 datasets. In the future, we'd like to study the prompt learning [53] to solve the challenging G-FOOD task.

APPENDIX

We present a derivation of Eq. (2)–(5). In a linear regression task, assume $y \in \mathbb{R}^N$ is the ground truth label, the model tries to find a $w \in \mathbb{R}^D$ to minimize

$$||y - \alpha \frac{X}{||X||} \frac{w}{||w||}||^2. \tag{17}$$

We set $\bar{x} = X/||X||$ and $\bar{w} = w/||w||$. When the weight sparsification is adopted, the objective function becomes

$$\begin{aligned}
&\underset{w}{minimize} \, \mathbb{E}_{R \sim \text{Bernoulli}(p)}[||y - \alpha\bar{x}(R * \bar{w})||^2] \\
&= \underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 + \alpha^2(1-p)^2||\tau\bar{w}||^2 \\
&= \underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 + \alpha^2(p^2 - 2p)||\tau\bar{w}||^2 \\
&\quad + \alpha^2||\tau\bar{w}||^2, \tag{18}
\end{aligned}$$

where $\tau = (diag(\bar{x}^T\bar{x}))^{1/2}$. We view the above formula as a function of p. Then, this can reduce to

$$\underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 + \alpha^2(p^2 - 2p)||\tau\bar{w}||^2. \tag{19}$$

This can reduce to

$$\underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 - \alpha^2 p(2-p)||\tau\bar{w}||^2. \tag{20}$$

This can reduce to

$$\underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 - 4\alpha^2\frac{p}{2}(1 - \frac{p}{2})||\tau\bar{w}||^2. \tag{21}$$

This can reduce to

$$\underset{w}{minimize}||y - \alpha p\bar{x}\bar{w}||^2 + \alpha^2 p(1-p)||\tau\bar{w}||^2. \tag{22}$$

We set $\tilde{w} = \alpha p\bar{w}$, then

$$\underset{w}{minimize}||y - \bar{x}\tilde{w}||^2 + \underbrace{\frac{(1-p)}{p}||\tau\tilde{\omega}||^2}_{Regularization \, term}. \tag{23}$$

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[2] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 21–37.

[3] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[4] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 9861–9870.

[5] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7348–7358.

[6] H. Hu, S. Bai, A. Li, J. Cui, and L. Wang, "Dense relation distillation with context-aware aggregation for few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10185–10194.

[7] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7359–7368.

[8] A. Wu, Y. Han, L. Zhu, and Y. Yang, "Universal-prototype enhancing for few-shot object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9547–9556.

[9] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8782–8791.

[10] Z. Fan, Y. Ma, Z. Li, and J. Sun, "Generalized few-shot object detection without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4527–4536.

[11] L. Qiao et al., "DeFRCN: Decoupled faster R-CNN for few-shot object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8661–8670.

[12] P. Kaul, W. Xie, and A. Zisserman, "Label, verify, correct: A simple few shot object detection method," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14217–14227.

[13] L. Zhu, H. Fan, Y. Luo, M. Xu, and Y. Yang, "Few-shot common-object reasoning using common-centric localization network," *IEEE Trans. Image Process.*, vol. 30, pp. 4253–4262, 2021.

[14] B. Su, H. Zhang, Z. Wu, and Z. Zhou, "FSRDD: An efficient few-shot detector for rare city road damage detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24379–24388, Dec. 2022.

[15] J. Han, Y. Ren, J. Ding, X. Pan, K. Yan, and G.-S. Xia, "Expanding low-density latent regions for open-set object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9581–9590.

[16] M. Jeong, S. Choi, and C. Kim, "Few-shot open-set recognition by transformation consistency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12561–12570.

[17] A. R. Dhamija, M. Günther, J. Ventura, and T. E. Boult, "The overlooked elephant of object detection: Open set," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1010–1019.

[18] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3243–3249.

[19] B. Liu, H. Kang, H. Li, G. Hua, and N. Vasconcelos, "Few-shot open-set recognition using meta-learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 8795–8804.

[20] C. Feng, Y. Zhong, and W. Huang, "Exploring classification equilibrium in long-tailed object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3397–3406.

[21] D. Pal, V. Bundele, R. Sharma, B. Banerjee, and Y. Jeppu, "Few-shot open-set recognition of hyperspectral images with outlier calibration network," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3801–3810.

[22] S. Huang, J. Ma, G. Han, and S.-F. Chang, "Task-adaptive negative envision for few-shot open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7161–7170.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

[24] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[25] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8420–8429.

[26] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9576–9585.

[27] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4013–4022.

[28] G. Zhang et al., "Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12832–12843, Nov. 2023.

[29] X. Du, Z. Wang, M. Cai, and Y. Li, "VOS: Learning what you don't know by virtual outlier synthesis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2022, pp. 1–21.

[30] M. Cai and Y. Li, "Out-of-distribution detection via frequency-regularized generative models," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 5510–5519.

[31] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5830–5840.

[32] Z. Wu, Y. Lu, X. Chen, Z. Wu, L. Kang, and J. Yu, "UC-OWOD: Unknown-classified open world object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 193–210.

[33] X. Zhao et al., "Revisiting open world object detection," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Oct. 20, doi: 10.1109/TCSVT.2023.3326279.

[34] J. Zheng, W. Li, J. Hong, L. Petersson, and N. Barnes, "Towards open-set object detection and discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3960–3969.

[35] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4401–4410.

[36] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf, "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2348–2354.

[37] D. Miller, N. Sünderhauf, M. Milford, and F. Dayoub, "Uncertainty for identifying open-set errors in visual object detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 215–222, Jan. 2022.

[38] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo, "Learning open-world object proposals without learning to classify," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5453–5460, Apr. 2022.

[39] K. Saito, P. Hu, T. Darrell, and K. Saenko, "Learning to detect every thing in an open world," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 268–284.

[40] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley, and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14523–14533.

[41] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 202'2, pp. 1–54.

[42] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 21464–21475.

[43] E. Mark, V. Luc, and W. Christopher, "The PASCAL visual object classes (VOC) challenge," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2010, pp. 303–338.

[44] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2015, pp. 740–755.

[45] A. Gupta, P. Dollár, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5351–5359.

[46] A. Gupta, S. Narayan, K. J. Joseph, S. Khan, F. S. Khan, and M. Shah, "OW-DETR: Open-world detection transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9225–9234.

[47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[49] I. Pavel, P. Dmitrii, G. Timur, V. Dmitry, and W. Andrew, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, 2018, pp. 876–885.

[50] G. Ishaan and L.-P. David, "In search of lost domain generalization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–29.

[51] C. Junbum et al., "SWAD: Domain generalization by seeking flat minima," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 22405–22418.

[52] A. Rame, M. Kirchmeyer, T. Rahier, A. Rakotomamonjy, P. Gallinari, and M. Cord, "Diverse weight averaging for out-of-distribution generalization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 10821–10836.

[53] Y. Gao et al., "Visual prompt tuning for test-time domain adaptation," 2022, *arXiv:2210.04831*.

Binyi Su received the B.S. degree in intelligent science and technology and the M.S. degree in control engineering from the Hebei University of Technology, Tianjin, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in computer science and technology with Beihang University, Beijing, China.

His current research interests include computer vision and pattern recognition.

Hua Zhang received the Ph.D. degree in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2015.

He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include computer vision, multimedia, and machine learning.

Jingzhi Li received the Ph.D. degree in cyberspace security from the University of Chinese Academy of Sciences, Beijing, China.

She is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences. She was selected to join the Science and Technology Think Tank Young Talents Program of the China Association for Science and Technology. Her current research interests include image processing and face security.

Zhong Zhou received the B.S. degree in material physics from Nanjing University in 1999 and the Ph.D. degree in computer science and technology from Beihang University, Beijing, China, in 2005.

He is currently a Professor and a Ph.D. Advisor with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His main research interests include virtual reality, augmented reality, computer vision, and artificial intelligence.