Enhancing Few-Shot Object Detection through Pseudo-Label Mining

Pablo Garcia-Fernandez[a,*], Daniel Cores[a], Manuel Mucientes[a]

[a] *Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain*

Abstract

Few-shot object detection involves adapting an existing detector to a set of unseen categories with few annotated examples. This data limitation makes these methods to underperform those trained on large labeled datasets. In many scenarios, there is a high amount of unlabeled data that is never exploited. Thus, we propose to **xPAND** the initial novel set by mining pseudo-labels. From a raw set of detections, xPAND obtains reliable pseudo-labels suitable for training any detector. To this end, we propose two new modules: Class and Box confirmation. Class Confirmation aims to remove misclassified pseudo-labels by comparing candidates with expected class prototypes. Box Confirmation estimates IoU to discard inadequately framed objects. Experimental results demonstrate that xPAND enhances the performance of multiple detectors up to +5.9 nAP and +16.4 nAP50 points for MS-COCO and PASCAL VOC, respectively, establishing a new state of the art. Code: `https://github.com/PAGF188/xPAND`.

Keywords: Few-shot, Object detection, Few-shot learning, Pseudo-label mining, Pseudo-labeling

1. Introduction

Object detection involves the process of identifying the class and position of all objects of interest that might appear in an image. In the last years, great success has been achieved in this field by training models with large databases of human-annotated labels [1, 2, 3]. However, the need to annotate large amounts of data limits the applicability of such object detectors in many real scenarios, as their performance drops significantly when data is limited. In response to this challenge, the emerging field of few-shot learning has gained prominence.

Few-shot learning techniques aim to extract general knowledge from large collections of base data and adapt quickly to limited novel data. Image classification with few-shot techniques has been widely studied as the first attempt to apply few-shot methods in computer vision [4, 5, 6, 7, 8]. Recently, the problem of few-shot object detection (FSOD) has attracted significant attention in order to replicate the success achieved in the field of image classification.

Nonetheless, the severe scarcity of labeled data for novel classes hinders the performance compared to approaches trained on large datasets.

The inclusion of unlabeled data for novel categories, which tends to be abundant in many scenarios, might mitigate this data scarcity. This approach has the potential to improve the detection precision at no additional annotation cost. The integration of unlabeled data has been extensively studied in the field of Semi-Supervised Object Detection (SSOD), and has recently begun to be explored within the few-shot paradigm [9]. While the objective in both cases is to increase the number of labeled samples through pseudo-label mining, the availability of abundant annotations for base categories is a key difference. SSOD methods aim to train an object detector from scratch on partially annotated datasets without any large fully annotated base training set. In contrast, FSOD approaches seek to adapt general knowledge extracted from this base set to new categories. In this paper, we propose to **xPAND** the initial novel set by mining pseudo-labels. Thus we propose a pseudo-label mining pipeline for FSOD that can be combined both with few-shot and standard object detectors. Fig. 1 shows that xPAND consistently increases the average precision on the novel categories (nAP) —those with very few annotations— for several object detectors on different shot sizes.

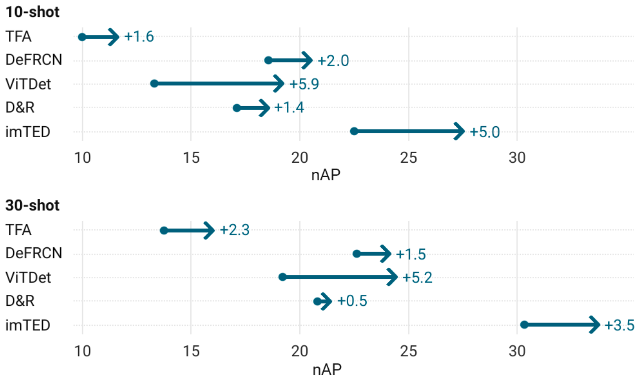

Figure 1: xPAND can be combined with any type of detector, either few-shot or standard, significantly improving the precision of the novel categories (nAP) in few-shot scenarios by exploiting the unlabeled data. Results show the nAP of five detectors, and the increase in nAP after applying xPAND with those detectors on the MS-COCO dataset for 10- and 30-shots.

The standard pseudo-labeling procedure consists of: (i) using

[*] Corresponding author

Email addresses: `pablogarcia.fernandez@usc.es` (Pablo Garcia-Fernandez), `daniel.cores@usc.es` (Daniel Cores), `manuel.mucientes@usc.es` (Manuel Mucientes)

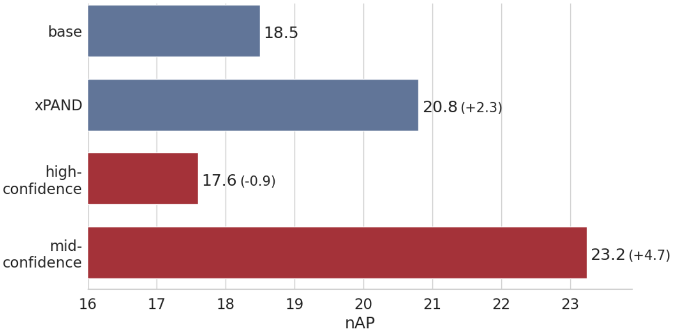

Figure 2: nAP for DeFRCN [10] with different pseudo-labeling techniques on MS-COCO 10-shot dataset. In red oracle results. (i) **Base**, DeFRCN [10] without pseudo-labeling; (ii) **xPAND**, DeFRCN+xPAND; (iii) **high-confidence oracle** and (iv) **mid-confidence oracle**, DeFRCN improved by filtering detections with confidence thresholds of 0.9 and 0.5, respectively, and removing those with an IoU less than 0.7 with ground truth. While confidence-based pseudo-labeling limits dataset diversity, xPAND enhances diversity and effectively filters noisy pseudo-labels, narrowing the gap to the best oracle results.

a detector, trained on a small labeled dataset, to generate an initial set of pseudo-labels from the unlabeled data; (ii) filtering these pseudo-labels; and (iii) retraining the detector with both the original labeled data and the filtered pseudo-labels.

The mined pseudo-labels are, however, biased by the detector, i.e., those objects similar to the initial labeled data are typically detected more accurately and with higher confidence, while those dissimilar have a higher probability of being overlooked. This could limit the variability of the training set, reducing the improvement of the final detector.

Most methods based on pseudo-label mining heavily rely on the detection confidence to include high-quality pseudo-labels in the training set, neglecting the influence of the detector bias. Fig. 2 shows the effect of this bias when retraining a few-shot object detector. Selecting with an oracle high-confidence detections as pseudo-labels causes a performance drop of -0.9 points in nAP, while selecting the same number of mid-confidence detections with the oracle, improves nAP +4.7 points. This observation effectively justifies our hypothesis that detection confidence is not a reliable estimator of pseudo-label quality.

Ideally, we would like to set the confidence threshold as low as possible to increase diversity. However, this introduces more noisy pseudo-labels to the initial set, requiring a strong filtering pipeline to remove them. Following this idea, xPAND starts with a set of unreliable pseudo-labels and enhances them by automatically filtering out those with incorrect class labels and/or inaccurately framed bounding boxes. The high diversity of the pseudo-labels selected by xPAND, together with its ability to filter noisy pseudo-labels, boosts the performance of the detector —+2.3 points in nAP for the example shown in Fig. 2.

To summarize, our contributions are as follows:

- We propose xPAND, a pseudo-label mining pipeline for FSOD that allows the extraction of high-quality and diverse pseudo-labels from a set of raw candidates obtained with any detector. The diversity of the mined pseudo-labels, and the robust filtering capabilities of our pipeline, enhances the performance of the detector, effectively addressing the inherent limitations of few-shot scenarios.
- A Class Confirmation module, built as a few-shot classifier on the meta-learning approach through contrastive learning. It eliminates misclassified pseudo-labels by comparing them with the prototype of their expected class.
- A Box Confirmation module, constructed as an Intersection over Union (IoU) estimator, which filters out incorrectly framed pseudo-labels.
- An extensive experimentation on MS-COCO and PASCAL VOC datasets using five different baseline detectors. Results show that xPAND sets a new state of the art for both MS-COCO and PASCAL VOC.

2. Related Work

Two popular approaches to learn an object detector with few annotated instances are FSOD and SSOD. In FSOD, abundant annotations for a set of base categories are available, while annotated data for novel categories is scarce. The objective of FSOD is to detect objects of novel categories by leveraging knowledge extracted from base categories. A related problem is Generalized FSOD, in which the performance of the final detector in base categories is also relevant. SSOD aims to exploit large amounts of unlabeled data without defining a base fully annotated training set. Our proposal is framed into FSOD, although it takes inspiration from SSOD.

Most common **SSOD** strategies to improve the performance are consistency regularization and pseudo-labeling. Consistency regularization forces the method to generate the same prediction under different transformations, like data augmentation, while pseudo-labeling exploits the unlabeled data to automatically generate new pseudo-labels. Most SSOD are based on a teacher-student architecture for knowledge distillation. Unbiased Teacher [11] focuses on solving class imbalance for exploiting pseudo-labeling. This is improved in Unbiased Teacher v2 [12] by introducing a regression loss for the pseudo-labels. In [13] they also follow a teacher-student framework but adapted to one-stage detectors. Finally, [14] adapts the teacher-student architecture to the DETR-based framework [15].

FSOD is usually solved following two paradigms: meta-learning and fine-tuning. Meta-learning approaches [16, 17, 18, 19, 20, 21] focus on learning a distance metric in which classification is performed by comparing an annotated support set with a query image. Fine-tuning-based methods [10, 22, 23, 24, 25], approach the problem as a transfer learning scenario, where a model is learned from base categories and adapted to novel categories by fine-tuning. TFA [22] pioneered the two-phase fine-tuning approach by adapting the final layers of the base model on few examples of novel classes. DeFRCN [10] modifies the Faster R-CNN framework by introducing two key components: a Gradient Decoupled Layer, which adjusts gradient scaling during the backward process, and a Prototypical Calibration Block, aimed at tuning the confidence scores for classification. D&R [26] refines DeFRCN via knowledge distillation. Leveraging CLIP [27] ——a multi-modal large-scale pre-trained

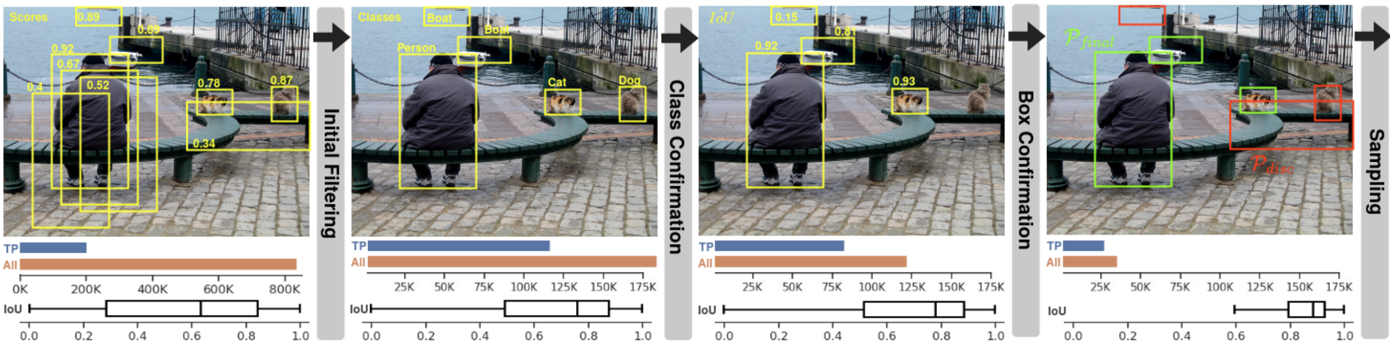

Figure 3: Main stages of xPAND. The resulting set of pseudo-labels $\mathcal{P}_{final}$ (green boxes) expands the initial training set, boosting the detector performance. Discarded pseudo-labels $\mathcal{P}_{disc}$ (red boxes) are ignored in the retraining of the detector. The total number of pseudo-labels and the good pseudo-labels (TP) —according to ground truth, correctly classified and with $IoU > 0.5$— are shown for each stage. The box plots show the IOU between pseudo-labels and ground truth. Data from imTED+xPAND on MS-COCO 10-shot.

model——, it introduces a new branch in the FSOD approach to extract text embedding representations of the categories. These text embeddings are aligned with the original visual features to guide the learning process towards a general and valuable semantic knowledge.

Self-supervised learning has improved object detection by pretraining with large unlabeled datasets. This is the case of imTED [23], which leverages a fully pretrained feature extraction path. MAE [28] is a masked autoencoder that learns to reconstruct the original image given its partial observation. DINO [29] uses a self-distillation process in which a teacher-student architecture learns to relate two slightly modified views of the same image. Both our Class and Box Confirmation modules are built on vision transformers (ViT) [30] pre-trained through self-supervision, trained on the base set, and fine-tuned on the novel set.

Another way to exploit unlabeled data in a few-shot framework is **pseudo-labeling**, which is inspired by semi-supervised learning. The idea is to expand the initial novel training set by adding pseudo-labels from unlabeled objects of novel categories to boost the performance of the detector. LVC [9] is the most representative work in this line. It defines a new customized detector to acquire candidate pseudo-labels. Then, the candidates with incorrect class labels are filtered out with a kNN classifier that uses features from a self-supervised ViT model. Finally, the bounding boxes are refined with a cascade of three class-agnostic regressors.

xPAND differs significantly from previous pseudo-labeling methods for FSOD and SSOD in two key ways. First, xPAND features plug-and-play adaptability, meaning it can be seamlessly integrated with any detector, whether standard or few-shot, whereas state-of-the-art methods often require specific detectors. Second, xPAND operates with a more diverse yet noisy pseudo-label set due to its reduced threshold for initial filtering. The Class and Box Confirmation modules effectively filter out noisy pseudo-labels, resulting in a final set of high-quality, diverse labels. This approach contrasts with traditional Semi-Supervised Object Detection (SSOD) methods, which typically rely on higher filtering thresholds and have less flexibility in detector choice.

3. FSOD with pseudo-label mining

3.1. Problem Definition

The standard configuration of the few-shot object detection problem [22, 20] consists of an image dataset, $\mathcal{D}$, with two sets of annotations, $\mathcal{Y}_{base}$ and $\mathcal{Y}^{K}_{novel}$, where each annotation $y_i = (c_i, b_i) \in \mathcal{Y}_{base} \cup \mathcal{Y}_{novel}$ is defined by its category c_i and its bounding box b_i. $\mathcal{Y}_{base}$ is composed of exhaustively annotated instances of base classes, C_{base}, while $\mathcal{Y}^{K}_{novel}$ consist of only K annotated instances per category, being C_{novel} the set of novel classes. C_{base} and C_{novel} are non-overlapping groups, *i.e.* $C_{base} \cap C_{novel} = \emptyset$, and K must be a small number —usually between 1 and 30. Objects of novel categories are not exhaustively annotated, so it is usual that in $\mathcal{D}$ there is a set of unlabeled objects belonging to $c_i \in C_{novel}$. xPAND exploits this set of unlabeled objects with a pseudo-label mining pipeline to expand the novel dataset with high-quality pseudo-labels, so that the final detector can be trained on the novel categories with a larger amount of instances, boosting its performance.

3.2. xPAND

Fig. 3 shows the xPAND pipeline. First, a detector —xPAND is not tied to any particular detector— trained on $\mathcal{Y}_{base} \cup \mathcal{Y}_{novel}$ is executed on $\mathcal{D}$ to generate the initial pseudo-label set. Many of these initial pseudo-labels have wrong categories and/or are poorly framed, especially those with mid-confidence values which are the ones that most improve the diversity of the pseudo-label set. Therefore, including those pseudo-labels directly in the training set of the final detector would hinder the learning process.

In the first stage, *Initial Filtering* applies Non-maximum Suppression (NMS) and a confidence threshold τ to remove very low-quality pseudo-labels. Previous pseudo-labeling methods rely on the detection score as the primary filtering criteria to discard most of the initial pseudo-labels. Although this ensures that only high-quality detections are considered after a simple initial filtering step, this metric might be heavily influenced by the detector bias. xPAND minimizes that by imposing a mid-confidence threshold τ, allowing a wide range of detections to be selected as pseudo-label candidates, thus significantly increasing the diversity of the pseudo-label set and improving the accuracy of the detector once the noisy pseudo-labels are filtered out in the next stages.

The second stage is *Class Confirmation* (Section 3.2.1), which consists of a meta-classifier with a DINO-pre-trained [29] ViT [30] as a feature extractor. It aims to exclude misclassified pseudo-labels by comparing the detection with the prototype of the predicted class. If both match, the pseudo-label is preserved. To ensure optimal discrimination performance, the training process follows a two-stage contrastive learning strategy. First, we leverage general knowledge from base categories, and then the model is fine-tuned on novel categories.

The next stage consists of a *Box Confirmation* module (Section 3.2.2), that aims to remove localization errors in the pseudo-label set. This component evaluates the localization accuracy by estimating the overlap between the pseudo-label candidate and the actual object. It also follows a two-stage fine-tuning learning approach leveraging a MAE-pre-trained [28] feature extractor. Pseudo-labels with a low estimated overlap are discarded. Fig. 3 shows how, after Box Confirmation, the distribution of pseudo-labels shifts towards higher overlaps with the ground truth, caused by the elimination of many poorly framed boxes.

The pseudo-label set resulting from the previous filtering stages is prone to a high class imbalance, mainly due to the initial detector bias. To address this, we set a maximum imbalance factor. Let $|\mathcal{P}_{c_i}|$ be the number of pseudo-labels for category c_i, then the maximum number of pseudo-labels that are randomly selected for each category is $\lambda\,|\mathcal{P}_{c_{min}}|$, being c_{min} the category with the lowest number of pseudo-labels.

The output of the pipeline is a set of final pseudo-labels $\mathcal{P}_{final}$, and a set of discarded object annotations $\mathcal{P}_{disc}$. $\mathcal{P}_{disc}$ includes all the annotations discarded by xPAND throughout its different stages. During the final end-to-end training, the detector is provided with $\mathcal{Y}_{base} \cup \mathcal{Y}^{K}_{novel} \cup \mathcal{P}_{final}$, but also with $\mathcal{P}_{disc}$. $\mathcal{P}_{disc}$ allows the detector to ignore image regions that may potentially contain objects. These ignored regions are neither background nor high-quality pseudo-labels, *i.e.*, RPN-generated proposals that overlap with them are not taken into account for loss computation.

To capitalize that the detector obtained through xPAND surpasses the base-detector performance, xPAND executes its pipeline iteratively. The pseudo-label set of the iteration j is generated with the detector obtained at the end of iteration $j-1$. The stopping criterion is:

$$\underset{c_i \in C_{novel}}{\text{median}} \left(\frac{\mathcal{P}^{j+1}_{final}(c_i)}{\mathcal{P}^{j}_{final}(c_i)} \right) < \chi, \tag{1}$$

so that xPAND stops in a given iteration j when the median increment in the number of pseudo-labels for each category is lower than a threshold χ.

3.2.1. Class Confirmation

The Class Confirmation module determines whether to discard or retain pseudo-labels based on their similarity to their corresponding class prototypes. It is designed as a few-shot classifier built on the meta-learning approach, which learns a distance metric that can accurately determine the similarity between objects.

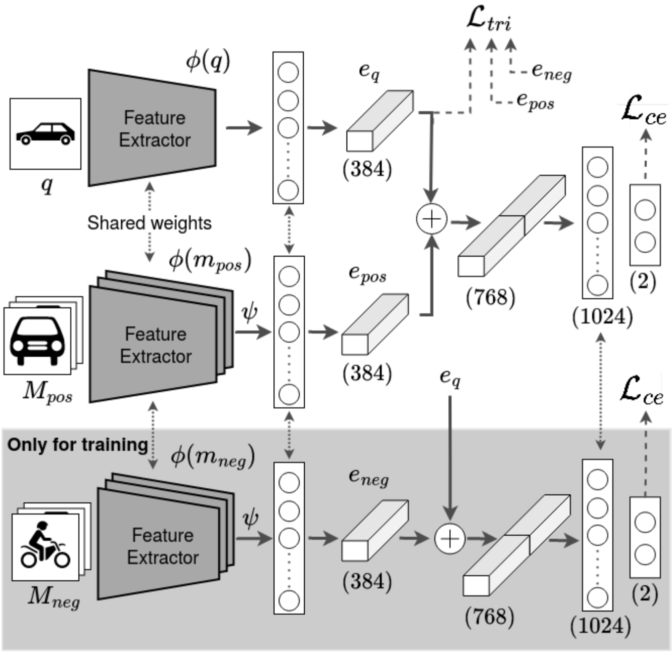

Figure 4: Class Confirmation architecture. The upper branch receives the object query to be confirmed (q). The middle and lower branches receive sets of positive (M_{pos}) and negative (M_{neg}) ground-truth objects. Triplet loss $\mathcal{L}_{tri}$ forces highly discriminative representations between the objects embeddings, e_q, e_{pos}, e_{neg}. $\mathcal{L}_{ce}$ is cross entropy. Numbers represent the size of the tensors

Fig. 4 shows the proposed architecture. Following a contrastive learning setting, it is composed of three branches. The upper branch receives as input an object query (q) whose label has to be confirmed. The middle and lower branches are given the sets of ground-truth objects (M_{pos}) and (M_{neg}), respectively, expected to belong to and not to belong to the query class (c).

The backbone ϕ processes q, $m_{pos} \in M_{pos}$ and $m_{neg} \in M_{neg}$ to extract their corresponding feature maps. To build it, we leverage recent advancements in self-supervised learning. Self-supervised models, trained on extensive unlabeled data, can learn general features that are useful for a variety of tasks. DINO [29] has proven effective for pre-training visual transformers and extracting classification-ready features. Consequently, we adopt the output CLS token of a DINO-pre-trained ViT [30] as the object embedding.

As M_{pos} and M_{neg} contain multiple support objects of the positive and negative categories, the prototype feature map is created by averaging the features for each object in the support set:

$$\psi(M) = \frac{1}{|M|} \sum_{m \in M} \phi(m), \tag{2}$$

where $M \in \{M_{pos}, M_{neg}\}$. The feature maps $\phi(q)$, $\psi(M_{pos})$ and $\psi(M_{neg})$ are fed into a shared fully connected layer to obtain e_q, e_{pos} and e_{neg}. Finally, e_q is concatenated with e_{pos} and e_{neg} generating positive and negative feature vectors that are passed to two fully connected layers. The final layer determines whether both elements belong to the same category.

The training of the Class Confirmation module follows a two-stage fine-tuning strategy. In the initial phase, the model is trained with samples of base classes, where the query and support samples come from $\mathcal{Y}_{base}$. In the second phase, a fine-tuning on $\mathcal{Y}_{novel}$ is performed to boost the performance on the novel classes. In both phases, training uses triplets (q, M_{pos} and M_{neg}).

To optimize the model, we define a multi-task loss function composed of a cross-entropy loss and a triplet loss. The cross-entropy loss is computed with the logits of each of the two classes. For a batch, it is formulated as follows:

$$\mathcal{L}_{ce}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{2} -Y_{i,t}\, log \frac{exp(X_{i,t})}{\sum_{j=1}^{2} exp(X_{i,j})}, \tag{3}$$

where n is the number of examples in the batch, X are the logits, and Y follows a one-hot encoding —indicates whether the query and the support belong to the same category.

Although cross-entropy is well-suited for classification tasks, we aim to construct a discriminative feature space enabling the differentiation of object pairs based on their similarity. To improve learning, we integrate a triplet loss [31] into our framework, encouraging the model to generate highly discriminative feature representations:

$$\mathcal{L}_{tri}(Q, P, N) = \frac{1}{n} \sum_{i=1}^{n} max\{d(e_q^i, e_{pos}^i) - d(e_q^i, e_{neg}^i) + \delta, 0\} \tag{4}$$

where n is the number of examples in the batch, ($e_q^i \in Q, e_{pos}^i \in P, e_{neg}^i \in N$) is the triplet formed by the query, positive and negative feature vectors, δ is the margin, and d is the Euclidean distance.

The final loss function is $\mathcal{L}_{class} = \mathcal{L}_{ce}(X, Y) + \mathcal{L}_{tri}(Q, P, N)$, which is crucial not only for constructing an effective classifier but also for creating a feature space that enhances intra-class similarities while accentuating inter-class dissimilarities. During inference, the negative branch is deleted, and the Class Confirmation module receives a pseudo-label to be confirmed and a set of support objects of the expected category. Those pseudo-labels that are not confirmed are filtered out.

3.2.2. Box Confirmation

The Box Confirmation module is an IoU estimator. It aims to predict the expected IoU between a pseudo-label bounding box and the ground truth. Fig. 5 shows the proposed architecture. It receives as input an image containing a pseudo-label. Similar to the Class Confirmation module, we also harness the benefits of unsupervised pretraining on pretext tasks to establish the backbone. MAE [28] pretraining method involves masking random patches of an input image, and training the model to reconstruct the missing pixels. This task aligns with our objective of estimating the IoU for objects. Specifically, pixel reconstruction helps the model in learning not only to identify object locations within an image, but also to discern the presence —or absence— of parts of objects. Therefore, the Box Confirmation backbone is implemented as a MAE-pretrained ViT [28], generating high-quality features for object localization.

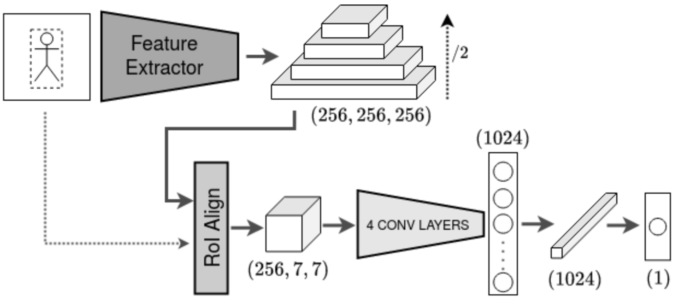

Figure 5: Box Confirmation architecture. The input is an image with a pseudo-label. After multi-scale feature extraction, ROI Align object features are passed through a regression header with a sigmoid function to estimate the expected IoU. Numbers represent the size of the tensors.

As in previous work on object detection with plain non-hierarchical vision transformers [2], we apply a set of deconvolution layers to generate a multiple-level feature map that enables multi-scale object localization. Then, the specific features of the candidate pseudo-label are obtained through the ROI Align method [32]. These features are fed to a regression header, comprising four convolutional layers and two fully connected layers. The final fully connected layer has a single output neuron with a sigmoid activation function. This configuration yields a continuous value within the range of 0 to 1, serving as an estimator for the IoU. Pseudo-labels can be discarded based on this value.

The training of the Box Confirmation module follows a two-stage fine-tuning strategy. In the initial phase, the model is trained with abundant samples of base classes. To generate the training examples we use the base initial detector, performing inference on $\mathcal{Y}_{base}$ and selecting a balanced set of detections at different IoU thresholds. In the fine-tuning stage, due to the scarcity of annotated data for novel classes, the training examples are generated from $\mathcal{Y}_{novel}$ by randomly applying offsets to the few annotated objects in each of the four primary directions. From this set of randomly generated proposals we select, as in the previous stage, an IoU-balanced set. In both stages, the optimization is performed using a binary cross-entropy loss function to measure the dissimilarity between the predicted IoU and the actual value. The formulation of this loss function for a batch is as follows:

$$\mathcal{L}(U, V) = \frac{1}{n} \sum_{i=1}^{n} -\{V_i\, logU_i + (1 - V_i)\, log(1 - U_i)\}. \tag{5}$$

where n is the batch size, U contains the predicted IoU for each element in the batch, and V the actual IoU between the input bounding box and the ground truth.

In inference, the Box Confirmation module receives both the image and the pseudo-label bounding box. The pseudo-label is discarded if the predicted IoU is lower than a specified threshold (β).

4. Experiments

4.1. Experimental Setup

We evaluate xPAND on both MS-COCO [33] and PASCAL VOC 2007/12 [34] benchmark datasets. For a fair comparison with previous works, we follow the two main evaluation protocols and data splits for FSOD: FSRW-like [20], i.e., a single support set, and TFA-like [22] with 10 support sets for MS-COCO and 5 supports sets for PASCAL VOC.

MS-COCO [33] has a total of 80 categories. In a few-shot scenario [22, 10, 9], the 60 categories disjoint with PASCAL VOC are the base classes, while the remaining 20 classes are the novel classes. The number of instances per novel class —shot size— is $K \in \{10, 30\}$. Following previous work in few-shot object detection, we report the standard MS-COCO evaluation metrics for novel categories: nAP ($IoU = 0.5 : 0.95$), nAP50 ($IoU = 0.5$) and nAP75 ($IoU = 0.75$)

PASCAL VOC 2007/12 [34] includes 20 object categories. Based on previous papers [10, 35, 19], we use the combination of *trainval* VOC07 and *trainval* VOC12 for training. VOC07 *test* set is used for evaluation. We use 3 rotating splits, each containing 15 base classes and 5 novel classes. The shot size is $K \in \{1, 2, 3, 5, 10\}$. Following standard experimentation protocols, we report the mean Average Precision (mAP) —setting an IoU threshold of 0.5.

To demonstrate xPAND's effectiveness across different detection frameworks, we integrated it with five distinct detectors: TFA [22], DeFRCN [10], VitDet [2], D&R [26], and imTED [23]. TFA and DeFRCN are CNN-based detectors that adapt the original Faster R-CNN [1] to the few-shot problem. D&R [26] extends DeFRCN to incorporate knowledge distillation from CLIP text category embeddings. VitDet and imTED are ViT-based detectors, but only imTED has been specifically designed for FSOD.

4.2. Implementation details

The Class Confirmation module uses as backbone a DINO-self-supervised ViT-S/8 model pre-trained on ImageNet. We keep its weights frozen. Input images are resized so that the smallest dimension is no more than 1,024, while always preserving the original aspect ratio. We adopt Adam optimization algorithm with a batch size of 24, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. On both base training and fine-tuning, the learning rate is set to 1×10^{-4} for the initial 5 epochs, and then reduced to 1×10^{-5} for the final 5 epochs, with a triplet loss margin $\delta = 1$. In base training, the support size is 10, while in fine-tuning, the support size corresponds to the shot number K. In both phases, images are augmented with horizontal flipping and color jittering —brightness 0.4, contrast 0.4, saturation 0.4, hue 0.2.

The Box Confirmation module uses as backbone a MAE-self-supervised ViT-B model pre-trained on ImageNet. The network is trained end-to-end using a batch size of 24 and the AdamW [36] optimization algorithm with standard configuration and weight decay 0.1. In base training, the learning rate is set to 1×10^{-4} for 75 epochs. In the fine-tuning stage, the learning rate is reduced to 1×10^{-5} for 8 epochs. In both phases, images are randomly resized so that the short edge is between 1,024 and 2,048. Then, a random crop of $1,024 \times 1,024$ pixels is performed as data augmentation. The minimum estimated overlap with the actual object is set to $\beta = 0.8$.

Configuration	TFA	DeFRCN	VitDet	D&R	imTED	AVG ΔnAP
Baseline	9.6	18.4	13.3	17.1	22.0	-
(1) N	6.5	13.5	8.4	11.7	14.3	-5.2
(2) F+S	10.9	19.4	16.9	17.2	26.3	+2.0
(3) F+C+S	11.3	19.6	18.2	17.5	27.4	+2.7
(4) F+B+S	11.4	20.1	17.2	17.8	26.0	+2.5
(5) F+C+B+S	11.6	20.5	18.7	18.6	27.5	+3.3

Table 1: Ablation study for MS-COCO 10-shot —FSRW-like experimentation and a single iteration—: base detector trained with no pseudo-labels (baseline), with all the initial pseudo-labels (N), and with the pseudo-labels from different components of xPAND pipeline —Initial Filtering (F), Class Confirmation (C), Box Confirmation (B) and balanced sampling (S). AVG ΔnAP is the average variation across all detectors compared to the corresponding baseline.

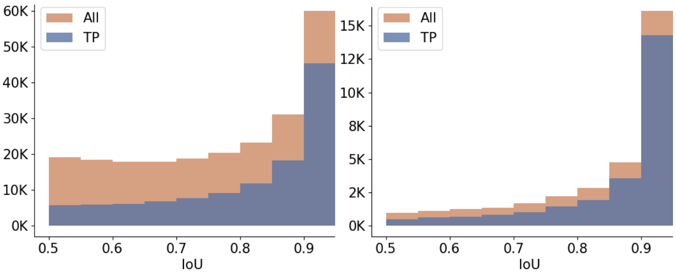

Figure 6: Distribution of pseudo-labels before (left) and after (right) xPAND. TP are pseudo-labels with correct category.

We remove from the initial pseudo-label set those detections with a confidence score lower than $\tau = 0.5$. The maximum imbalance factor λ for the final pseudo-label set is set to 10, and the threshold for the stopping criterion is $\chi = 25\%$. In the experiments, xPAND is always executed with the same hyper-parameters for all the detectors, datasets and shot sizes.

4.3. Ablation Study

Table 1 shows the ablation study for the different components of xPAND and the five selected base detectors. For (1), all detections are selected as pseudo-labels to retrain the detector. A significant performance drop can be observed in every detector, loosing 5.2 nAP points on average. This proves the need for robust filtering approaches that limit the presence of inaccurate pseudo-labels in the final training set. In (2), we apply the Initial Filtering and the final sampling to reduce the class imbalance, overcoming the baseline by 2.0 points on average. The inclusion of the Class Confirmation (3) and Box Confirmation (4) modules further improves the nAP with an average difference with the baseline of 2.7 and 2.5 nAP points respectively. Finally, the execution of one iteration of all the components defined in xPAND achieves a total improvement of 3.3 nAP points. This proves that the Class Confirmation and Box Confirmation modules are complementary, and that they are able to generate a high-quality pseudo-label set by filtering many of the noisy pseudo-labels.

CC		BC		nAP
DINO	MAE	MAE	DINO	
✓		✓		20.5
✓			✓	20.4
	✓	✓		19.7
	✓		✓	19.7

Table 2: Impact of DINO/MAE pre-trained ViT backbones on the Class Confirmation (CC) and Box Confirmation (BC) stages. DeFRCN [10] 10-shot FSRW-like experimentation[20] on MS-COCO.

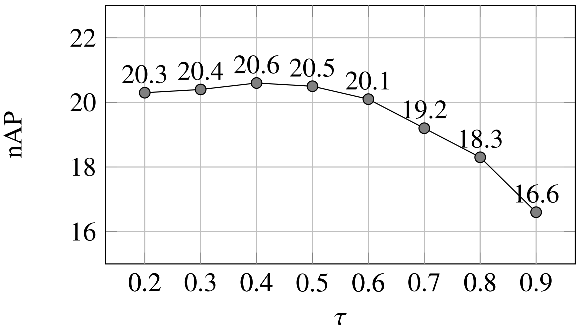

Figure 7: Influence of confidence threshold τ. DeFRCN [10] 10-shot FSRW-like [20] on MS-COCO.

Fig. 6 shows the distribution of pseudo-labels before and after applying xPAND. Those approaches that select only high-confidence pseudo-labels build a more reliable training set, although due to the bias of the initial detector, the diversity of pseudo-labels is also low. We strive to recover as many mid-confidence pseudo-labels as possible to reduce the aforementioned bias issue, setting a confidence threshold of $\tau = 0.5$. The counterpart is that mid-confidence detections introduce many noisy and unreliable pseudo-labels that require a strong filtering pipeline. As we apply xPAND, the number of pseudo-labels significantly decreases but, also, the proportion of correct pseudo-labels is notably higher, particularly for the lower confidence cases. This enables our method to consider a wider range of confidence scores without dramatically reducing the quality of the training set.

Table 2 analyses the influence of various pre-trained ViT backbones on the Class Confirmation and Box Confirmation stages. It is clear that DINO performs slightly better than MAE at the Class Confirmation stage (nAP 20.5 vs. 19.7), indicating DINO's advantage in obtaining classification-ready features. At the Box Confirmation stage, the choice between DINO and MAE has minimal effect, as both offer similar performance.

Fig. 7 shows the influence of the confidence threshold τ. Within the broad range of 0.2 to 0.6, the value of τ has minimal impact, highlighting xPAND's filtering capability. However, as the threshold increases and label diversity decreases, performance drops because many good pseudo-labels are filtered out, and those that remain reflect detector biases and fail to capture more diverse and informative instances. This observation reinforces our hypothesis that relying exclusively on high-confidence thresholds for pseudo-labeling is ineffective. Instead, lowering the confidence threshold and applying a robust filtering strategy, as implemented in xPAND, can lead to better results.

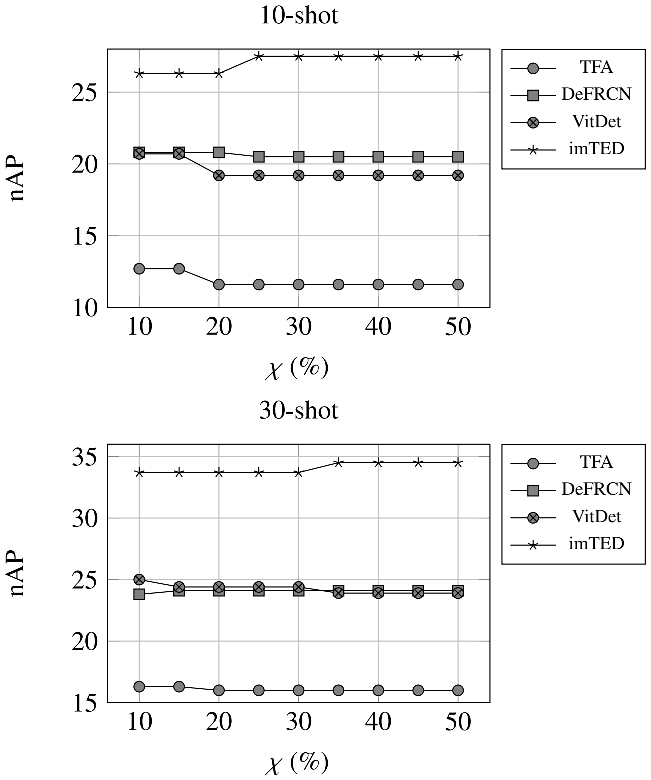

Figure 8: Influence of stop criterion threshold χ. TFA [22], DeFRCN [10], VitDet [2] and imTED [23] detectors on both 10-shot and 30-shot FSRW-like [20] MS-COCO experimentation.

Fig. 8 presents a comparative analysis of the stop criterion χ across four different detectors. The results demonstrate that χ is not highly sensitive, implying that achieving optimal performance does not necessitate extensive tuning. Furthermore, χ exhibits even lower sensitivity in the 30-shot setting compared to the 10-shot setting, suggesting that its influence diminishes as the shot size increases.

4.4. Comparison Results

Table 3 shows the results of the state-of-the-art FSOD and SSOD methods on the MS-COCO dataset. xPAND is able to significantly improve all the detectors. For TFA, DeFRCN, and D&R, the improvements range between +0.5 and +2.3 nAP points, but for ViT-based detectors the improvements reach up to +5.9 nAP points. The less inductive bias of modern ViT-based object detectors, such as VitDet, makes them suffer from severe overfitting when training with few examples. Thus, ViTDet underperforms other CNN-based detectors, like DeFRCN. However, when expanding its training set with pseudo-labels, ViTDet achieves very competitive results with a performance boost of more than +5 nAP points for both 10 and 30-shot. Although imTED mitigates the inductive bias issue by integrally pretraining the feature extraction path to perform better in

Method	nAP	
	10-shot	30-shot
TFA [22] (ICML 20)	10.0	13.7
DeFRCN [10] (ICCV 21)	18.5	22.6
FSOD-SR [25] (PR 21)	11.6	15.2
DCNet [37] (CVPR 21)	12.8	18.6
LVC* [9] (CVPR 22)	19.0	26.8
FCT [38] (CVPR 22)	15.3	21.4
CFA [39] (CVPR 22)	19.1	23.0
ViTDet† [2] (ECCV 22)	13.3	19.2
Meta-DETR [19] (TPAMI 22)	19.0	22.2
DCFS [40] (NIPS 22)	19.5	22.7
σ-ADP [35] (ICCV 23)	20.3	20.8
MLFDA [41] (CVPR 23)	18.8	23.4
Norm-VAE [42] (CVPR 23)	18.7	22.5
NIFF [43] (CVPR 23)	18.8	20.9
D&R† [26] (AAAI 23)	17.1	20.9
imTED [23] (ICCV 23)	**22.5**	**30.2**
Consistent-Teacher*† [44] (CVPR 23)	14.4	20.2
Semi-DETR*† [45] (CVPR 23)	**22.7**	**29.3**
TFA [22] (ICML 20) + **xPAND**	$11.6_{\uparrow 1.6}$	$16.0_{\uparrow 2.3}$
DeFRCN [10] (ICCV 21) + **xPAND**	$20.5_{\uparrow 2.0}$	$24.1_{\uparrow 1.5}$
VitDet [2] (ECCV 22) + **xPAND**	$19.2_{\uparrow 5.9}$	$24.4_{\uparrow 5.2}$
D&R [26] (AAAI 23) + **xPAND**	$18.5_{\uparrow 1.4}$	$21.4_{\uparrow 0.5}$
imTED [23] (ICCV 23) + **xPAND**	$\mathbf{27.5}_{\uparrow 5.0}$	$\mathbf{33.7}_{\uparrow 3.5}$

Table 3: MS-COCO results following the experimental setting from [20]. Red, blue and green, represent 1st, 2nd and 3rd respectively, and the numbers on the right of the arrows show the difference with the corresponding baseline. † indicates our own experiments. * indicates pseudo-labeling methods.

Method	10-shot			30-shot		
	nAP	nAP50	nAP75	nAP	nAP50	nAP75
TFA [22]	9.7 ±0.6	18.1 ±1.2	9.3 ±0.6	12.7 ±0.3	23.6 ±0.5	12.2 ±0.3
TFA + xPAND‡	11.0 ±0.7	19.8 ±1.3	11.0 ±0.7	14.7 ±0.4	26.1 ±0.6	14.9 ±0.5
TFA + xPAND	**12.0** ±0.7	**21.2** ±1.3	**12.1** ±0.7	**15.2** ±0.4	**26.7** ±0.6	**15.6** ±0.5
DeFRCN [10]	19.0 ±0.4	33.9 ±0.7	19.0 ±0.6	22.7 ±0.3	39.8 ±0.4	23.0 ±0.5
DeFRCN + xPAND‡	**21.5** ±0.3	**36.5** ±0.5	**22.5** ±0.4	**24.0** ±0.3	**40.1** ±0.4	**25.4** ±0.3
DeFRCN + xPAND	**21.5** ±0.3	**36.5** ±0.5	**22.5** ±0.4	**24.1** ±0.3	**40.1** ±0.4	**25.4** ±0.3
ViTDet [2]	12.5 ±0.5	21.1 ±0.8	12.9 ±0.5	17.4 ±0.5	28.8 ±0.8	18.1 ±0.7
ViTDet + xPAND‡	18.7 ±0.5	30.2 ±0.9	20.1 ±0.5	24.2 ±0.4	38.1 ±0.5	26.2 ±0.5
ViTDet + xPAND	**20.2** ±0.5	**32.0** ±0.8	**22.0** ±0.5	**25.2** ±0.4	**39.0** ±0.6	**27.6** ±0.4
D&R [26]	15.5 ±0.5	29.0 ±1.2	14.6 ±0.4	19.4 ±0.4	35.5 ±0.7	18.9 ±0.4
D&R + xPAND‡	**17.5** ±0.5	**31.7** ±1.1	**17.5** ±0.5	**20.7** ±0.3	**37.3** ±0.7	**20.7** ±0.3
D&R + xPAND	**17.5** ±0.5	**31.7** ±1.1	**17.5** ±0.5	**20.7** ±0.3	**37.3** ±0.7	**20.7** ±0.3
imTED [23]	19.1 ±0.9	29.7 ±1.4	20.6 ±1.0	26.2 ±1.0	38.8 ±2.3	28.7 ±1.1
imTED + xPAND‡	26.2 ±0.6	38.8 ±0.9	29.2 ±0.6	**31.3** ±0.9	**45.9** ±1.1	**35.1** ±1.0
imTED + xPAND	**26.7** ±0.7	**39.0** ±0.9	**29.8** ±0.7	30.7 ±0.9	44.7 ±1.2	34.5 ±1.0

Table 4: MS-COCO results following the experimental setting from [22]. In bold face the best results for each group. ‡ indicates a single iteration of xPAND. ±*x* is the confidence interval.

low-data regimes, it still highly benefits from xPAND, improving nAP +5 and +3.5 points for 10 and 30-shot respectively. imTED+xPAND sets a new state of the art for both shot sizes on the MS-COCO dataset, outperforming previous methods, including those based on pseudo-labeling like LVC.

Concerning semi-supervised methods, Table 3 includes two originally SSOD detectors: Consistent-Teacher [44] and Semi-DETR [45]. Both utilize a teacher-student framework where the teacher generates pseudo-labels to guide the student's learning. Simultaneously, the student updates the teacher's weights via EMA. For a fair comparison, we adapted these methods to the few-shot setting. Specifically, the teacher and student were first trained on the fully labeled base classes. Afterwards, a pseudo-labeling online training was performed on novel classes, using both the teacher-generated pseudo-labels and the 10/30 annotated novel examples. xPAND outperforms the SSOD methods in both 10-shot and 30-shot scenarios. This highlights the distinct challenges of SSOD and FSOD tasks, and indicates that xPAND is more effective at managing the constraints of limited novel labeled data when abundant base data is available.

We also conducted the experimentation with different support sets following [22] (Table 4), including the baseline detector, a single iteration of xPAND, and the standard execution of xPAND with several iterations. Results show that a single iteration of xPAND suffices to improve base detectors in every metric for 10 and 30 shot sizes. The iterative execution of xPAND even improves the single iteration results for TFA and ViTDet, for DeFRCN and D&R it has no impact, and for imTED it is positive for 10-shot and negative for 30-shot. As in 3, ViT-based detectors —ViTDet and imTED— benefit the most from the combination with xPAND.

Table 5 shows a comparison with previous methods on PASCAL VOC dataset for five different shot sizes and three different splits [20]. xPAND is able to improve all baseline detectors across various shot sizes. The average nAP50 increments are 4.4, 4.7, 8.4, 8.5, 3.1 points for TFA, DeFRCN, ViTDet, imTED, and D&R respectively. Furthermore, the combination of D&R+xPAND yields superior performance compared to prior approaches, thereby setting a new state of the art across a majority of shots and splits.

In contrast to COCO, the transformer-based methods VitDet and imTED —before applying xPAND— do not outstand in VOC. Considering that the size of VOC is 10 times smaller than COCO, this highlights the data-hungry nature of transformers and their tendency to overfit on datasets with significantly fewer training examples for both base and novel categories. The combination of these methods with xPAND, partially alleviates this problem for FSOD.

On VOC we also conducted a TFA-like experimentation [22] with different support sets. The results for both the baseline detectors and their integration with xPAND are presented in Table 6. It can be seen how xPAND consistently enhances the performance of the baseline detectors. The average increases in nAP50 are 3.6, 2.4, 4.6, 6.4, and 3.4 points for TFA, DeFRCN, ViTDet, imTED, and D&R respectively. This underscores xPAND's effectiveness as a consistent and impactful pseudo-label mining pipeline.

Method	Novel Set 1					Novel Set 2					Novel Set 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW [20] (ICCV 19)	13.8	19.6	32.8	41.5	45.6	7.9	15.3	26.2	31.6	39.1	9.8	11.3	19.1	35.0	45.1
TFA† [22] (ICML 20)	21.3	22.8	25.2	28.7	34.3	7.1	16.6	22.0	22.1	19.2	16.1	14.2	19.8	27.9	28.1
FSOD-SR [25] (PR 21)	50.1	54.4	56.2	60.0	62.4	29.5	39.9	43.5	44.6	48.1	43.6	46.6	53.4	53.4	59.5
DeFRCN† [10] (ICCV 21)	51.2	53.1	47.2	64.3	57.8	30.5	39.0	48.8	51.7	47.7	43.6	45.7	53.0	56.4	54.8
LVC [9] (CVPR 22)	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
ViTDet† (ECCV 22) [2]	27.1	37.2	31.9	43.9	46.2	11.1	29.5	35.7	35.1	35.9	24.4	32.5	34.3	36.7	37.9
DCFS [40] (NIPS 22)	46.2	57.4	59.9	62.9	64.5	32.6	39.9	43.4	47.9	51.3	40.3	50.5	53.8	56.9	60.7
Meta-DETR [19] (TPAMI 22)	35.1	49.0	53.2	57.4	62.0	27.9	32.3	38.4	43.2	51.8	34.9	41.8	47.1	54.1	58.2
Norm-VAE [42] (CVPR 23)	**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**
imTED† (ICCV 23) [23]	11.2	12.3	13.6	34.8	44.8	4.4	9.8	22.1	18.4	35.9	9.9	16.9	18.3	35.3	35.8
D&R† (AAAI 23) [26]	60.4	**64.0**	**65.2**	64.7	**66.3**	**37.9**	**46.8**	48.1	**52.7**	**53.1**	**55.7**	**57.9**	**57.6**	**60.6**	**61.9**
TFA (ICML 20) [22] + xPAND	22.8 ↑1.5	28.4 ↑5.6	28.3 ↑3.1	39.1 ↑10.4	44.2 ↑9.9	6.9 ↓0.2	13.0 ↓3.6	14.0 ↓8.0	20.8 ↓1.3	24.3 ↑5.1	16.2 ↑0.1	14.4 ↑0.2	22.8 ↑3.0	35.8 ↑7.9	34.8 ↑6.7
DeFRCN (ICCV 21) [10] + xPAND	**60.6** ↑9.4	61.6 ↑8.5	60.3 ↑13.1	**64.9** ↑0.6	61.1 ↑3.3	**41.4** ↑10.9	42.3 ↑3.3	47.9 ↓0.9	51.6 ↓0.1	50.2 ↑2.5	38.8 ↓4.8	50.2 ↑4.5	56.2 ↑3.2	58.4 ↑2.0	58.8 ↑4.0
ViTDet (ECCV 22) [2] + xPAND	31.4 ↑4.3	40.2 ↑3.0	47.7 ↑15.8	58.6 ↑14.7	60.8 ↑14.6	14.8 ↑3.7	22.9 ↓6.6	26.1 ↓9.6	36.7 ↑1.6	42.0 ↑6.1	23.2 ↓1.2	27.0 ↓5.5	44.7 ↑10.4	51.7 ↑15.0	52.7 ↑14.8
imTED (ICCV 23) [23] + xPAND	11.6 ↑0.4	21.9 ↑9.6	30.0 ↑16.4	45.9 ↑11.1	47.0 ↑2.2	10.7 ↑6.3	18.2 ↑8.4	30.2 ↑8.1	28.4 ↑10.0	34.5 ↓1.4	18.0 ↑8.1	28.7 ↑11.8	35.5 ↑17.2	41.3 ↑6.0	46.3 ↑10.5
D&R (AAAI 23) [26] + xPAND	**61.7** ↑1.3	**69.5** ↑5.5	**70.0** ↑4.8	**71.2** ↑6.5	**70.7** ↑4.4	36.4 ↓1.5	**47.9** ↑1.1	**49.9** ↑1.8	**56.0** ↑3.3	**55.5** ↓2.4	**56.5** ↑0.8	**60.0** ↑2.1	**61.6** ↑4.0	**64.2** ↑3.6	**65.7** ↑3.8

Table 5: PASCAL VOC results following the experimental setting from [20]. Red, blue and green, represent 1st, 2nd and 3rd. Numbers beside the arrows show the difference with baseline. † indicates our own experiments.

Method	Novel Set 1					Novel Set 2					Novel Set 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
TFA† (ICML 20) [22]	19.4 ±3.7	27.2 ±3.5	30.2 ±4.5	32.0 ±3.8	35.5 ±1.3	10.5 ±4.1	18.6 ±2.0	20.5 ±2.3	22.7 ±2.7	26.4 ±3.6	10.4 ±4.2	15.6 ±2.2	20.4 ±2.6	25.3 ±2.9	31.7 ±2.2
TFA† + xPAND	22.9 ±0.9	30.0 ±3.2	32.9 ±2.5	39.1 ±1.6	43.4 ±1.7	12.2 ±2.5	20.8 ±4.6	21.2 ±6.0	20.1 ±2.2	27.0 ±1.4	11.8 ±3.5	18.9 ±2.5	26.1 ±2.3	32.9 ±2.1	36.8 ±1.3
	↑ 3.5	**↑ 2.8**	**↑ 2.7**	**↑ 7.1**	**↑ 7.9**	**↑ 1.8**	**↑ 2.1**	**↑ 0.7**	**↓ 2.6**	**↑ 0.5**	**↑ 1.4**	**↑ 3.3**	**↑ 5.7**	**↑ 7.6**	**↑ 5.2**
DeFRCN† (ICCV 21) [10]	44.0 ±3.6	55.4 ±3.5	55.9 ±4.9	61.8 ±1.9	60.7 ±4.5	30.2 ±3.2	40.1 ±2.6	45.5 ±2.3	49.6 ±1.7	52.1 ±3.1	34.6 ±8.3	48.0 ±5.5	50.9 ±3.6	54.8 ±3.2	58.3 ±1.7
DeFRCN† + xPAND	49.5 ±6.0	57.8 ±2.7	59.3 ±3.8	66.1 ±1.4	62.3 ±4.0	34.1 ±4.3	41.7 ±3.6	45.7 ±2.8	48.8 ±3.0	51.5 ±1.8	39.5 ±9.8	52.0 ±2.8	55.2 ±2.4	57.1 ±2.2	58.6 ±0.8
	↑ 5.5	**↑ 2.4**	**↑ 3.4**	**↑ 4.3**	**↑ 1.7**	**↑ 3.9**	**↑ 1.6**	**↑ 0.2**	**↓ 0.8**	**↓ 0.6**	**↑ 4.9**	**↑ 4.0**	**↑ 4.3**	**↑ 2.3**	**↑ 0.3**
ViTDet† (ECCV 22) [2]	24.7 ±4.6	36.5 ±5.3	38.4 ±5.2	39.3 ±4.9	40.9 ±3.9	13.7 ±2.4	22.8 ±4.4	30.1 ±4.9	32.7 ±4.2	36.0 ±3.0	18.2 ±4.5	29.5 ±6.8	33.4 ±6.4	34.9 ±4.0	36.8 ±4.2
ViTDet† + xPAND	25.4 ±4.2	39.0 ±3.9	38.7 ±3.7	53.2 ±3.3	58.7 ±1.8	13.0 ±3.8	22.2 ±2.3	29.7 ±4.2	34.0 ±3.8	42.1 ±1.8	16.8 ±4.9	31.2 ±4.4	35.5 ±7.0	45.8 ±4.2	51.4 ±3.6
	↑ 0.6	**↑ 2.5**	**↑ 0.3**	**↑ 13.9**	**↑ 17.8**	**↓ 0.7**	**↓ 0.6**	**↓ 0.5**	**↑ 1.3**	**↑ 6.1**	**↓ 1.5**	**↑ 1.8**	**↑ 2.1**	**↑ 11.0**	**↑ 14.7**
imTED† (ICCV 23) [23]	8.3 ±2.7	21.3 ±4.6	21.6 ±5.6	36.8 ±1.7	47.3 ±3.6	5.4 ±1.6	11.4 ±1.8	16.6 ±2.5	22.7 ±3.1	34.0 ±4.0	6.0 ±3.3	15.8 ±2.5	20.8 ±6.7	31.0 ±5.9	42.4 ±3.3
imTED† + xPAND	10.8 ±2.4	29.3 ±4.2	36.9 ±4.9	36.9 ±4.4	44.6 ±2.0	10.0 ±2.4	17.0 ±1.3	23.3 ±4.9	28.6 ±3.4	36.3 ±2.0	9.0 ±6.2	28.7 ±1.5	34.5 ±3.4	37.6 ±5.1	47.1 ±1.0
	↑ 2.5	**↑ 8.0**	**↑ 15.3**	**↑ 0.1**	**↓ 2.6**	**↑ 4.7**	**↑ 5.6**	**↑ 6.6**	**↑ 5.9**	**↑ 2.3**	**↑ 3.0**	**↑ 13.0**	**↑ 13.6**	**↑ 6.6**	**↑ 4.6**
D&R† (AAAI 23) [26]	42.3 ±9.0	57.0 ±4.8	55.6 ±6.4	62.3 ±3.7	64.9 ±3.4	34.3 ±3.4	42.8 ±4.0	46.0 ±2.9	49.8 ±1.9	53.4 ±0.9	35.4 ±11.9	49.0 ±6.2	54.3 ±4.1	59.3 ±2.1	60.0 ±1.1
D&R† + xPAND	47.0 ±7.9	63.2 ±4.0	62.4 ±5.4	67.3 ±3.4	69.0 ±3.1	31.9 ±4.0	41.9 ±3.7	46.8 ±2.0	52.3 ±2.1	55.6 ±2.7	39.1 ±11.8	57.3 ±2.3	60.6 ±1.1	63.1 ±1.0	63.8 ±1.0
	↑ 4.7	**↑ 6.2**	**↑ 6.8**	**↑ 5.0**	**↑ 4.1**	**↓ 2.4**	**↓ 0.9**	**↑ 0.8**	**↑ 2.5**	**↑ 2.2**	**↑ 3.7**	**↑ 8.3**	**↑ 6.3**	**↑ 3.8**	**↑ 3.8**

Table 6: VOC results following the experimental setting from [22]. Numbers beside the arrows show the difference with the corresponding baseline (**blue** increase, **red** decrease). † indicates our own experiments. ±x is the confidence interval.

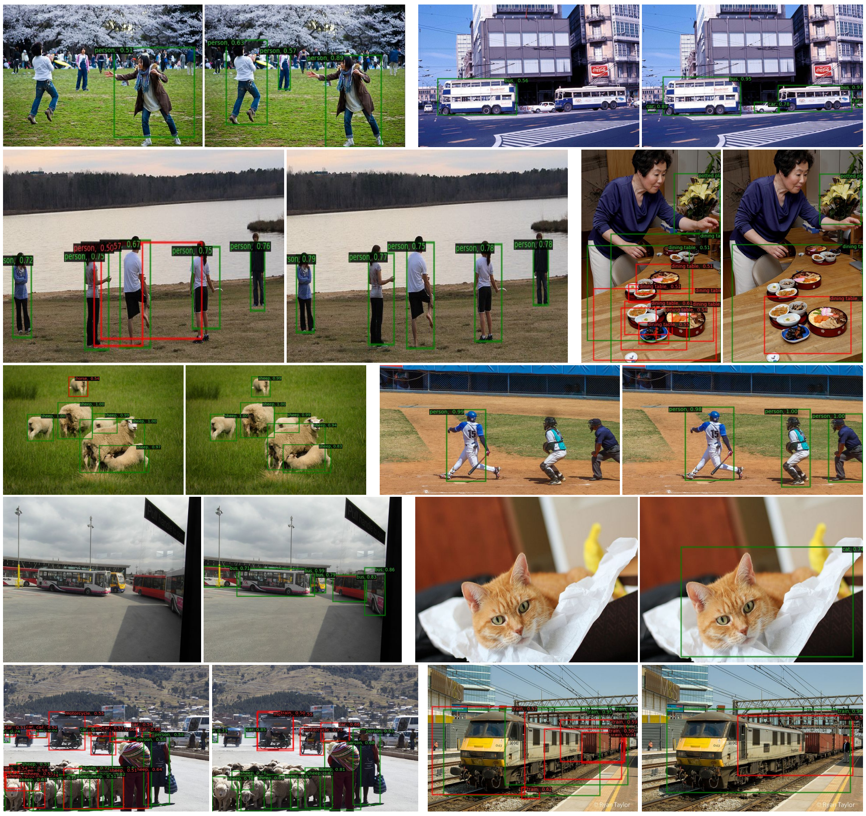

Figure 9: Visualization results on 30-shot MS-COCO dataset. We show the bounding boxes with score over 0.5. Each image pair illustrates the outcomes without xPAND (left) and with xPAND (right). Each row corresponds to a method, listed from top to bottom as TFA, DeFRCN, ViTDet, imTED, and D&R.

4.5. xPAND meets CLIP

In this section we explore the idea of using state-of-the-art Vision-Language Models for pseudo-label generation. We conduct a comparative analysis of xPAND and CLIP, evaluating their performance as filtering mechanisms across various shot sizes. This analysis aims to assess the individual filtering effectiveness of each method and to explore their complementary strengths when combined.

Table 7 presents the results. Using the two best-performing models as baselines for each dataset tested —MS COCO and PASCAL VOC— we compare xPAND and CLIP both separately and in combination. The results for CLIP are obtained by filtering the detections from the base detector to exclude instances where the predicted category does not align with CLIP's predictions. The combined results of CLIP+xPAND are achieved by incorporating CLIP's knowledge into xPAND as an additional step to recover discarded pseudo-labels. Specifically, the pseudo-labels eliminated by Class Confirmation or Box Confirmation, but whose class coincides with the class predicted by CLIP, are recovered.

CLIP outperforms xPAND in small-shot scenarios —COCO 10-shot and VOC 1-10-shot— due to its strong zero-shot capabilities, leveraging its pretrained alignment of visual and textual representations on vast image-text pairs. However, as the number of labeled samples increases, xPAND surpasses CLIP by better utilizing task-specific labeled data, extracting more detailed and specialized knowledge that improves performance in higher shot settings —COCO 30-shot and VOC 20,30-shot.

	MS COCO	
Method	**10**	**30**
imTED	22.5	30.2
imTED + CLIP	28.6	32.4
imTED + xPAND	27.5	33.7
imTED + xPAND + CLIP	**31.4**	**36.5**

	PASCAL VOC						
Method	**1**	**2**	**3**	**5**	**10**	**20**	**30**
D&R	51.3	56.2	57.0	59.3	60.4	61.5	63.4
D&R + CLIP	56.6	62.7	62.8	65.1	64.1	62.9	64.7
D&R + xPAND	51.5	59.1	60.5	63.8	64.0	63.3	**65.2**
D&R + xPAND + CLIP	**57.2**	**63.1**	**63.3**	**65.2**	**64.7**	**63.5**	65.1

Table 7: Comparison of results between CLIP alone, xPAND alone, and their combination. Best results for each dataset are highlighted in bold.

However, the most significant finding is that the combination of CLIP and xPAND yields the best performance, highlighting the complementarity of both methods. CLIP's strong zero-shot generalization, and xPAND's ability to exploit labeled data to obtain new training samples, demonstrate that integrating these approaches can enhance the robustness of pseudo-labeling techniques

4.6. *Qualitative Results*

Fig. 9, shows some qualitative visualizations of the detected novel objects on MS-COCO dataset. Each pair of images emphasizes the distinctions between the detector trained with the starting annotated data and the one trained with the pseudo-labels obtained through xPAND. We showcase both successful (green boxes), and failure instances (red boxes). Results show the improvement of those methods based on xPAND for both classification and object localization. Furthermore, it is clear that xPAND contributes to decrease the number of undetected objects. All of these observations strongly support the reliability of xPAND.

5. Conclusion

We have presented xPAND, a pseudo-label mining pipeline designed to produce diverse and high-quality pseudo-labels for training detectors within a few-shot framework. Grounded on Class and Box Confirmation modules, xPAND effectively filters out numerous low-quality pseudo-labels initially present in the pseudo-label set. xPAND can be combined with any object detector, either few-shot or standard, generally improving nAP across all datasets and shot sizes, and establishing a new state of the art on both MS COCO and VOC datasets.

Acknowledgements

This research was partially funded by the Spanish Ministerio de Ciencia e Innovación (grant number PID2020-112623GB-I00, PID2023-149549NB-I00), and the Galician Consellería de Cultura, Educación e Universidade (grant numbers ED431C 2018/29 and ED431G2019/04). These grants are co-funded by the European Regional Development Fund (ERDF). Pablo Garcia-Fernandez is supported by the Spanish Ministerio de Universidades under the FPU national plan (grant number FPU21/05581).

References

[1] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Adv. Neural Inform. Process. Syst., 2015.

[2] Y. Li, H. Mao, R. Girshick, K. He, Exploring plain vision transformer backbones for object detection, in: Eur. Conf. Comput. Vis., 2022.

[3] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: Int. Conf. Comput. Vis., 2021.

[4] S. X. Hu, D. Li, J. Stühmer, M. Kim, T. M. Hospedales, Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[5] P. Bateni, R. Goyal, V. Masrani, F. Wood, L. Sigal, Improved few-shot visual classification, in: IEEE Conf. Comput. Vis. Pattern Recog., 2020.

[6] P. Rodríguez, I. Laradji, A. Drouin, A. Lacoste, Embedding propagation: Smoother manifold for few-shot classification, in: Eur. Conf. Comput. Vis., 2020.

[7] F. Liu, S. Yang, D. Chen, H. Huang, J. Zhou, Few-shot classification guided by generalization error bound, Pattern Recognition 145 (2024) 109904.

[8] Z. Zhao, Q. Liu, W. Cao, D. Lian, Z. He, Self-guided information for few-shot classification, Pattern Recognition 131 (2022) 108880.

[9] P. Kaul, W. Xie, A. Zisserman, Label, verify, correct: A simple few shot object detection method, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[10] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, C. Zhang, DeFRCN: Decoupled faster R-CNN for few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2021.

[11] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, P. Vajda, Unbiased teacher for semi-supervised object detection, in: Int. Conf. Learn. Represent., 2020.

[12] Y.-C. Liu, C.-Y. Ma, Z. Kira, Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[13] C. Liu, W. Zhang, X. Lin, W. Zhang, X. Tan, J. Han, X. Li, E. Ding, J. Wang, Ambiguity-resistant semi-supervised learning for dense object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[14] J. Zhang, X. Lin, W. Zhang, K. Wang, X. Tan, J. Han, E. Ding, J. Wang, G. Li, Semi-DETR: Semi-supervised object detection with detection transformers, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Eur. Conf. Comput. Vis., 2020.

[16] Q. Fan, W. Zhuo, C.-K. Tang, Y.-W. Tai, Few-shot object detection with attention-rpn and multi-relation detector, in: IEEE Conf. Comput. Vis. Pattern Recog., 2020.

[17] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, L. Lin, Meta R-CNN: Towards general solver for instance-level low-shot learning, in: IEEE Conf. Comput. Vis. Pattern Recog., 2019.

[18] S. Zhang, L. Wang, N. Murray, P. Koniusz, Kernelized few-shot object detection with efficient integral aggregation, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[19] G. Zhang, Z. Luo, K. Cui, S. Lu, E. P. Xing, Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation, IEEE Trans. Pattern Anal. Mach. Intell. (2022).

[20] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, T. Darrell, Few-shot object detection via feature reweighting, in: IEEE Conf. Comput. Vis. Pattern Recog., 2019.

[21] T.-I. Chen, Y.-C. Liu, H.-T. Su, Y.-C. Chang, Y.-H. Lin, J.-F. Yeh, W.-C. Chen, W. Hsu, Dual-awareness attention for few-shot object detection, IEEE Trans. Multimedia (2021).

[22] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, F. Yu, Frustratingly simple few-shot object detection, in: Int. Conf. Mach. Learn., 2020.

[23] F. Liu, X. Zhang, Z. Peng, Z. Guo, F. Wan, X. Ji, Q. Ye, Integrally migrating pre-trained transformer encoder-decoders for visual object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[24] J. Wu, S. Liu, D. Huang, Y. Wang, Multi-scale positive sample refinement for few-shot object detection, in: Eur. Conf. Comput. Vis., 2020.

[25] G. Kim, H.-G. Jung, S.-W. Lee, Spatial reasoning for few-shot object detection, Pattern Recognition 120 (2021) 108118.

[26] J. Li, Y. Zhang, W. Qiang, L. Si, C. Jiao, X. Hu, C. Zheng, F. Sun, Disentangle and remerge: interventional knowledge distillation for few-shot object detection from a conditional causal perspective, in: AAAI, 2023.

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: Int. Conf. Mach. Learn., 2021.

[28] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[29] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: IEEE Conf. Comput. Vis. Pattern Recog., 2021.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, Int. Conf. Learn. Represent. (2021).

[31] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: Int. Worksh. Similarity-Based Pattern Recog., 2015.

[32] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Int. Conf. Comput. Vis., 2017.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context,

in: Eur. Conf. Comput. Vis., 2014.

[34] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.

[35] J. Du, S. Zhang, Q. Chen, H. Le, Y. Sun, Y. Ni, J. Wang, B. He, J. Wang, s-adaptive decoupled prototype for few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[36] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: Int. Conf. Learn. Represent., 2018.

[37] H. Hu, S. Bai, A. Li, J. Cui, L. Wang, Dense relation distillation with context-aware aggregation for few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2021.

[38] G. Han, J. Ma, S. Huang, L. Chen, S.-F. Chang, Few-shot object detection with fully cross-transformer, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[39] K. Guirguis, A. Hendawy, G. Eskandar, M. Abdelsamad, M. Kayser, J. Beyerer, CFA: constraint-based finetuning approach for generalized few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2022.

[40] B.-B. Gao, X. Chen, Z. Huang, C. Nie, J. Liu, J. Lai, G. Jiang, X. Wang, C. Wang, Decoupling classifier for boosting few-shot object detection and instance segmentation, in: Adv. Neural Inform. Process. Syst., 2022.

[41] B. Demirel, O. B. Baran, R. G. Cinbis, Meta-tuning loss functions and data augmentation for few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[42] J. Xu, H. Le, D. Samaras, Generating features with increased crop-related diversity for few-shot object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[43] K. Guirguis, J. Meier, G. Eskandar, M. Kayser, B. Yang, J. Beyerer, NIFF: Alleviating forgetting in generalized few-shot object detection via neural instance feature forging, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[44] X. Wang, X. Yang, S. Zhang, Y. Li, L. Feng, S. Fang, C. Lyu, K. Chen, W. Zhang, Consistent-teacher: Towards reducing inconsistent pseudo-targets in semi-supervised object detection, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.

[45] J. Zhang, X. Lin, W. Zhang, K. Wang, X. Tan, J. Han, E. Ding, J. Wang, G. Li, Semi-detr: Semi-supervised object detection with detection transformers, in: IEEE Conf. Comput. Vis. Pattern Recog., 2023.