Label, Verify, Correct: A Simple Few Shot Object Detection Method

Prannay Kaul[1]   Weidi Xie[1,2]   Andrew Zisserman[1]

[1]Visual Geometry Group, University of Oxford   [2]Shanghai Jiao Tong University
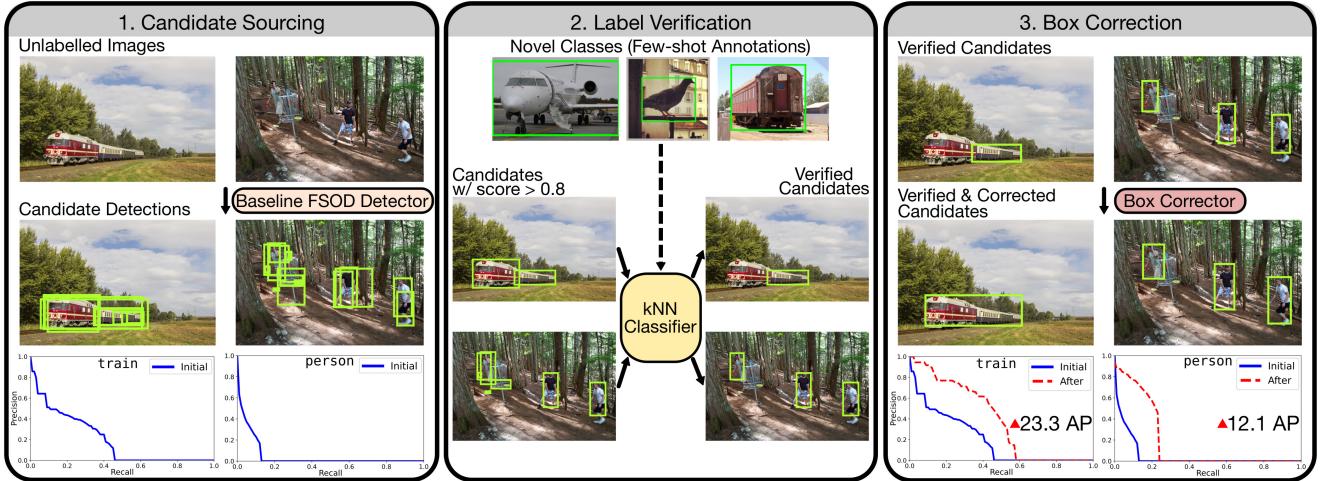
http://www.robots.ox.ac.uk/~vgg/research/lvc/

Figure 1. **Label, Verify, Correct:** Object detectors naïvely trained with K-shot annotations perform poorly on novel classes (bottom left). We propose to expand the novel class annotation set and re-train *end-to-end*. 1. Given a baseline few-shot object detector, noisy candidate detections are sourced from unlabelled images (left). 2. Labels for each candidate detection are verified by a kNN classifier, constructed from a self-supervised model using the same few-shot annotations, removing large numbers of false positives (centre). 3. A specialised box corrector, drastically improves the remaining bounding boxes, yielding high-quality pseudo-annotations (right). Re-training end-to-end with our pseudo-labelling method yields a large performance boost on novel class detection, improving precision and recall (bottom right).

Abstract

The objective of this paper is few-shot object detection (FSOD) – the task of expanding an object detector for a new category given only a few instances for training. We introduce a simple pseudo-labelling method to source high-quality pseudo-annotations from the training set, for each new category, vastly increasing the number of training instances and reducing class imbalance; our method finds previously unlabelled instances.

Naïvely training with model predictions yields sub-optimal performance; we present two novel methods to improve the precision of the pseudo-labelling process: first, we introduce a verification technique to remove candidate detections with incorrect class labels; second, we train a specialised model to correct poor quality bounding boxes. After these two novel steps, we obtain a large set of high-quality pseudo-annotations that allow our final detector to be trained end-to-end. Additionally, we demonstrate our method maintains base class performance, and the utility of simple augmentations in FSOD. While benchmarking on PASCAL VOC and MS-COCO, our method achieves state-of-the-art or second-best performance compared to existing approaches across all number of shots.

1. Introduction

Object detection refers to the task of determining if an image contains objects of a particular category, and if so, then localising them. In recent years, the community has seen tremendous successes in object detection by training computational models for a set of pre-defined object classes [8, 17, 32, 36, 40, 46, 55], with large numbers of human annotated labels, *e.g.* MS-COCO [30], and PASCAL VOC [11]. However, such training paradigms have limited the model to only perform well on a closed, small set of categories for which large training data is available.

In contrast, humans can continuously expand their vocabularies, learning to detect a much larger set of categories, even with access to only a few examples [43]. This is also a desirable ability for modern computer vision systems and is studied in the task of few-shot object detection (FSOD) [13, 22, 45, 51, 56]. The goal of our work is FSOD: given an existing object detector that has been

trained on abundant data for some categories, termed *base* categories, we wish to learn to detect *novel* categories using only a few annotations, *e.g.* 1–30 per category, whilst maintaining performance on the original base categories.

In this paper, we carefully examine the training procedure of the Faster R-CNN two-stage detector, and identify two critical factors that limit its performance on FSOD. *First*, training on base categories leads to a form of "supervision collapse" [9]; this model is naturally trained against detecting instances from novel classes, as they were unlabelled and treated as background; *second*, the FSOD problem involves learning from extremely unbalanced data – only K instances ($K \leq 30$) are available per novel class for training, so the number of training samples for *base* categories is far larger than that for *novel* ones. A model that overfits to a small number (K) of novel instances naturally lacks generalisation ability.

We adopt a simple *pseudo-labelling* technique (see Figure 1) and address both the factors that limit performance: we show that the Region Proposal Network (RPN) can be modified to successfully propose regions for the novel categories and use a detector trained with the few-shot *novel* data to label these regions over the images of the large training dataset, producing a set of candidate detections for each novel category. The novelty of our approach is in the two steps used to improve the precision of this candidate set: *first*, we build a classifier for novel categories to verify the candidate detections using features from a network trained with self-supervision (see Figure 1, centre); *second*, we train a specialised box regressor that improves the quality of the bounding box of the verified candidates (see Figure 1, right). The two steps together yield a large set of high precision pseudo-annotations for the novel categories, removing the class imbalance in the training data. This enables the detector to be trained end-to-end using the pseudo-annotations for the novel categories together with the original groundtruth annotations for base categories, and simultaneously avoids the impact of "supervision collapse" for *all* detector features.

To summarise, our contributions are as follows: (i) we carefully examine the problem of few-shot object detection with the modern two-stage object detector, *e.g.* Faster R-CNN, and identify the issue of "supervision collapse"; (ii) we introduce a novel verification and correction procedure to pseudo-labelling, which significantly improves the precision of pseudo-annotations, both class labels and bounding box coordinates; (iii) we analyse several critical components of data augmentation and conduct thorough ablation studies to validate their necessity; (iv) with the combination of pseudo-labelling and aggressive data augmentations, we set state-of-the-art (SotA) or comparable performance using a standard Faster R-CNN, for both the challenging MS-COCO benchmark and the PASCAL VOC benchmark.

We discuss limitations of our work and potential ethical concerns in Appendix E and F, respectively. Code and pre-trained models are available from the project webpage.

2. Related Work

Object Detection is one of the classical problems in computer vision, which makes it impossible to present a full overview here. We therefore only outline some key milestones here. In general, recent object detection methods can be cast into two sets: one-stage and two-stage detectors. *One-stage* detectors attempt to directly classify *and* regress bounding boxes by, either densely classifying a set of predefined anchor boxes [32, 33, 37, 38, 39, 46] or densely searching for geometric entities of objects *e.g.* corners, centres or regions [25, 47, 58]. Conversely, most *two-stage* detectors propose class-agnostic bounding boxes using a Region Proposal Network (RPN), with predefined sizes and aspect ratios, filtering out many negative (background) locations. These bounding box proposals are pooled to region-of-interest (RoI) features and are classified by a multilayer perceptron (MLP) in the second stage of the detector [14, 15, 28, 40].

Few-Shot Object Detection aims to expand the vocabulary of an object detector by only annotating a handful of samples. Several works [5, 12, 13, 17, 20, 22, 23, 26, 27, 29, 36, 45, 48, 49, 50, 53, 56, 57, 59] have been proposed in the recent literature. The meta-learning method, FSRW [22] conditions dense query image features from a YOLOv2 network [38] with a separate network operating on a support set. Recently, a simple two-phase fine-tuning approach (TFA) is proposed [48], in which a Faster R-CNN model is initially trained on the base data. In the second training phase, only the final classification layer is finetuned on a few samples of novel classes, with the rest of the model fixed. This work initiates a shift away from meta-learning based methods for few-shot object detection. FSCE [45], alleviates class confusion between novel classes by training a separate supervised contrastive learning [24] head on RoI features. A recent work, DeFRCN [36], decouples the training of RPN features and RoI classification. Another recent work, SRR-FSD [59] combines vision and natural language, projecting image features into semantic class embeddings learnt from a large text corpus.

Semi-supervised Object Detection belongs to another related research area. Such a problem setup can be traced back to the pre-deep learning era [41], where the goal is to train detectors with a combination of labelled, weakly-labelled and unlabelled data. In the recent literature, the idea of exploiting consistency and self-training has been widely adopted, for example, [21] proposed to enforce the predictions of an input image and its flipped version to be consistent; [34, 44] pre-trained a detector using a small amount of labelled data and generated pseudo-labels on

unlabelled data for further fine-tuning. Generally speaking, these methods aim to train a detector on two separate subsets; one contains images with all objects of all classes being *exhaustively* annotated, and the other subset is fully unlabelled. Therefore, the model trained on the labelled set does not suffer the same issue as in FSOD, where a large number of *novel* object instances are wrongly treated as background during base training. Hence the scarce data issue of FSOD *does* exist in semi-supervised object detection, but "supervision collapse" *does not*.

Self-Training is a method for gaining noisy pseudo-labels which has gained renewed interest since it was initially proposed [42]. In recent years, self-training has been used to improve image classification by using a teacher-student training regime [6, 19, 54]. This idea is extended to general object detection in [60], however, their considered scenario is semi-supervised object detection.

3. Background and Supervision Collapse

In this section, we first outline the few-shot object detection task in Section 3.1. Next, in Section 3.2, we carefully examine the various components of the popular two-stage detector (Faster R-CNN), identify the critical issues that limit its use in the few-shot scenario and draw conclusions to ameliorate such critical issues.

3.1. Problem Definition

In this paper, we consider the same problem setup as in TFA [22]. Specifically, assuming we are given an image dataset, $\mathcal{D}$, and two annotation sets. First, $\mathcal{Y}_{\text{BASE}}$, with exhaustive annotations on a set of base categories, $\mathcal{C}_{\text{BASE}}$. Second, $\mathcal{Y}_{\text{NOVEL}}^{K}$, with only K annotations on a set of novel categories, $\mathcal{C}_{\text{NOVEL}}$. Note, the annotations on base categories are exhaustive, but for novel categories most instances are *unlabelled* as only K annotations are provided for the image dataset, $\mathcal{D}$, under the few-shot setting.

3.2. Training Strategy

In this section, we start by describing a baseline two-stage detector for the problem of few-shot object detection, following that of TFA [48]. In general, a Faster R-CNN detector, $\Phi_{\text{DET}}(\cdot)$, can be formulated as:

$$\Phi_{\text{DET}}(\cdot) = \Phi_{\text{CLS}} \circ \Phi_{\text{ROI}} \circ \Phi_{\text{RPN}} \circ \Phi_{\text{ENC}}(\cdot)$$

where, each input image is sequentially processed by a set of operations: an image encoder, (Φ_{ENC}); a Region Proposal Network, (Φ_{RPN}); a region of interest feature module, (Φ_{ROI}); and a classification layer on the RoI features, (Φ_{CLS}), mapping to a set of bounding boxes and classes. Note that, each module here contains the same number of convolutional or MLP layers as the standard Faster R-CNN [40].

Training few-shot object detectors involves a two-phase training procedure, as detailed below:

Novel Training?	# Proposals	nAR$_{\text{IoU}=0.5}$	min(R50$_{\text{IoU}=0.5}$)
✗	100	49.7	16.2
✓	100	71.0	40.0
Ideal RPN [40]	100	84.0	64.5
✗	1000	82.1	55.9
✓	1000	88.5	77.0
Ideal RPN [40]	1000	95.0	79.2

Table 1. RPN recall evaluation on 20 novel categories from the MS-COCO few-shot object detection benchmark, using an IoU=0.5 criterion. nAR50 – novel class average recall, min(nR50) – minimum novel class recall. Please refer to the text for detailed discussion.

Base Training: refers to the standard training for a Faster R-CNN model [40, 52] using *base class annotations* only, $\mathcal{Y}_{\text{BASE}}$. In this work, we do not modify this training regime.

Novel Training: requires extending the base detector, such that it can additionally detect instances from novel categories, *i.e.* $\mathcal{C}_{\text{BASE}} \cup \mathcal{C}_{\text{NOVEL}}$. In recent works, this is usually done by only training (relatively few) layers on novel and base class data, $\mathcal{Y}_{\text{BASE}}^{K} \cup \mathcal{Y}_{\text{NOVEL}}$; the detector is not trained end-to-end on novel class data. For example, in TFA [48], the fewest possible number of parameters are trained on novel class data, namely, Φ_{CLS} only.

Such a two-phase training strategy naturally leads to two questions: (i) Does an RPN trained on base categories actually generalise, *i.e.* are regions proposed for instances of the novel categories? (ii) How well do features trained only on base categories actually generalise, in other words, will the RoI features be discriminative for classifying *novel* categories? We aim to answer these two questions on the MS-COCO 30-shot object detection benchmark (these benchmarks are detailed in Section 5.1). Specifically, we follow the same data split as in TFA [48], with 60 categories being treated as base categories, and 20 as novel categories.

3.2.1 On Generalisability of RPNs

In standard two-stage object detectors, an RPN is considered as a necessary condition for high-performance detections, as classification and box coordinate regression will only act on the proposed regions. Here, we aim to evaluate the quality of an RPN for FSOD, based on recall with respect to the novel categories.

Specifically, we consider the following three settings: *first*, to understand whether an RPN trained on 60 base categories can directly propose novel object instances, we evaluate the recall of the RPN from the base detector; *second*, we finetune the RPN (composed of 2 convolutional layers) on both base and given novel categories, *i.e.* $\mathcal{Y}_{\text{BASE}} \cup \mathcal{Y}_{\text{NOVEL}}^{30}$; *third*, the Ideal RPN which is inherited from an off-the-shelf Faster R-CNN trained on exhaustive

data for all categories.

Discussion: Table 1 presents the large performance gap between an RPN from the base detector, and the Ideal RPN with respect to recalling instances from novel categories. However, finetuning the RPN specific parameters, Φ_{RPN}, on a handful of instances ($K = 30$), yields a substantial increase in average recall (49.7 vs. 71.0, 82.1 vs. 88.5, for 100 and 1000 proposals respectively), largely bridging the performance gap to the Ideal RPN. The minimum class recall also increases substantially (16.2 vs. 40.0, 55.9 vs. 77.0, for 100 and 1000 proposals respectively).

3.2.2 On Transferability of Base Features

In this section, we aim to measure the transferability of the visual features trained on base categories. Specifically, we keep the encoder (Φ_{ENC}) fixed, and finetune, individual or combinations of, subsequent modules ($\Phi_{CLS}, \Phi_{ROI}, \Phi_{RPN}$) during **Novel Training**.

Setting	Novel Training			Metrics		
	Φ_{CLS}	Φ_{ROI}	Φ_{RPN}	nAP	nAP50	nAP75
A1	✓	✗	✗	13.0	24.7	12.3
A2	✓	✓	✗	13.3	25.6	12.6
A3	✓	✓	✓	14.3	27.5	12.9
A4 (ALL)	✓	✗	✗	18.3	33.3	17.7
Ideal Faster R-CNN [40]				43.5	67.4	46.6

Table 2. Evaluation on the transferability of base features to 20 novel categories from the MS-COCO few-shot object detection benchmark. During **Novel Training**, we jointly finetune the different modules with the base and novel category data (only the given few-shot annotations, $\mathcal{Y}_{NOVEL}^{K}$). As an Oracle test, we also consider to finetune the classifier with *all* MS-COCO annotations, *e.g.* A4 (ALL). nAP – novel class average precision. Please refer to the text for discussion.

Discussion: As shown in Table 2, we compare TFA [48] (Setting A1) with finetuning more layers (A3), which tends to be beneficial (13.0 vs. 14.3 nAP), however, it remains substantially lower than the Ideal Faster R-CNN (14.3 vs. 43.5 nAP). To remove the factor caused by insufficient data annotation, we compare TFA (A1) with an Oracle test (A4-ALL) that finetunes the final classifier with exhaustive data on all categories. However, the result still largely underperforms the Ideal Faster R-CNN reference (18.3 vs. 43.5nAP), indicating that the feature encoder is heavily biased towards base classes, and hardly contains discriminative information for classifying instances from novel classes. This may be expected, as these categories were treated as background during **Base Training**.

These experiments demonstrate the "supervision collapse" issue present in FSOD detectors. We note that "supervision collapse" manifests in two ways: *first*, many false positives occur due to class confusion and poor bounding box regression (poor detection precision), *second*, there are

many false negatives or missing detections, despite an improved RPN (poor detection recall). We further analyse these manifestations in Appendix A.

3.2.3 Summary

After careful evaluations, we draw the following two conclusions: *first*, updating all parameters specific to the RPN (Φ_{RPN}) is essential to improve recall on novel categories; *second*, features trained on base category data are **not** discriminative enough to classify novel instances, leading to severe performance degradation, and so we update all parameters in the RoI feature module (Φ_{ROI}), in addition to the classification layer (Φ_{CLS}). These two choices constitute our **Novel Training** process, yielding a stronger baseline detector. This allows enough alleviation of "supervision collapse", such that our baseline detector can be used as a starting point in our pseudo-labelling method, as will be detailed in the next section.

4. Method

To address the "supervision collapse" issue, we adopt a simple *pseudo-labelling* method for mining instances of novel categories, effectively expanding their annotation set. However, pseudo-annotations that are naïvely sourced from the detector (after **Novel Training**), are unreliable, containing a large number of false positives. Here, we establish a method for improving the precision of these candidate pseudo-annotations by automatically filtering out candidates with incorrect class labels, and refining the bounding box coordinates for those remaining. Our method yields a large set of high precision pseudo-annotations for novel categories, allowing the final detector to be trained end-to-end on both base and novel category data. We detail the proposed method in the following sections.

4.1. Candidate Sourcing

The goal here is to generate a set of candidate detections that are *potentially* valid pseudo-annotations for novel categories. Specifically, the detector from **Novel Training** (as described in Section 3.2), is used to perform inference on the training images ($\mathcal{D}$) to generate a set of candidate detections, each containing a class label and predicted bounding box coordinates (see Figure 1, left). We limit the size of this set to be 1000s by taking *novel class* detections with high confidence scores, here we use $q > 0.8$, producing $\hat{\mathcal{Y}}_{NOVEL}$.

As demonstrated in earlier evaluations, the detector from **Novel Training** cannot detect instances from novel categories well, leaving a large number of incorrect predictions in the set of candidates, either misclassification, or imprecise bounding box coordinates. The key question now becomes, *how can we improve the precision of this list?*

4.2. Label Verification

In this section we take inspiration from the work on query expansion by Chum *et al.* [7], which uses spatial ver-

ification to accept or reject new instances during retrieval. The goal here is to verify the predicted class label for each candidate detection. Specifically, we consider to build a classifier for the novel categories with the very limited few-shot annotations ($\mathcal{Y}_{\text{NOVEL}}$).

Building classifiers with only a few annotations is clearly not a trivial task, as it often demands high-quality feature representations. Here, we benefit from the recent development of self-supervised models, *e.g.* MoCo [18], SwAV [3], DINO [4], and construct kNN classifiers with the high-quality features produced from those models. In practice, this work uses the output `CLS` token from a ViT model [10] trained with the self-supervised DINO [4] method, where the NN performance is shown to be particularly strong.

To perform Label Verification (see Figure 1, centre), we first compute features for each of the given novel class groundtruth annotations, using the self-supervised model. These features are used as the training data in our kNN classifier. Similarly, we compute features for each instance in the set of candidate detections using the same self-supervised model. In detail, to compute the feature of a given annotation/candidate detection, we first use the bounding box to crop the relevant image. This crop is then resized and passed as input to the self-supervised model.

We adopt a simple verification policy: a given candidate detection is accepted (or verified) if our kNN classifier, using cosine similarity, predicts the same class as the predicted class label from the detector. With such a verification step, we obtain a *verified* set of candidate detections with high precision with respect to classification labels.

4.3. Box Correction

In addition to verifying classification labels, we consider refining the bounding boxes for all remaining candidate detections in the *verified* set (see Figure 1, right). Taking inspiration from the Cascade R-CNN [2], we build a separate model, containing three *class-agnostic* regressors that gradually produce a higher-quality bounding box, with each only processing boxes of similar IOUs to the groundtruth.

Specifically, during **Novel Training**, we divide the RPN proposals into three splits using different IoU thresholds, and pass the RoI features through the corresponding regressors. For example, all features pooled from IoU> 0.3 boxes are passed to the first regressor, features pooled from IoU> 0.5 boxes are passed to the second regressor and features pooled from IoU> 0.7 boxes are passed to the third regressor. Once this is trained, the bounding boxes for the *verified* set can be corrected, by feeding their RoI features through the three regressors in succession.

We now possess a large set of *previously unlabelled* novel instances, with high precision class labels and high-quality bounding boxes. This verified and corrected set is then used as pseudo-annotations to re-train our detector

end-to-end on novel and base class instances.

5. Experiments

In this section, we first introduce the standard experimental benchmarks used in the literature [22, 45]. After this, in Section 5.2 we describe our implementation, training details, and conduct extensive ablation studies on the design choices of this work in Section 5.3. Lastly, inheriting the best experience from the ablation studies, we compare to the existing state-of-the-art approaches in Section 5.4.

5.1. Few-Shot Object Detection Benchmarks

We follow the same benchmarks as in [22], evaluating our model on the MS-COCO [30] and PASCAL VOC [11] datasets. To maintain a fair comparison, we use the same fixed lists of novel samples and data splits given in [22].

MS-COCO has 80 categories in total. In FSOD, the 20 categories present in PASCAL VOC are used as novel classes and the remaining categories are used as base classes. In this case, the benchmarks are designed for testing with $K = 10, 30$ shots, and we report standard MS-COCO metrics, namely Average Precision (IoU=0.5 : 0.95), Average Precision (IoU=0.5) and Average Precision (IoU=0.75) on novel classes, abbreviated to nAP, nAP50 and nAP75, respectively. Our ablation studies are all conducted on the MS-COCO benchmark.

PASCAL VOC contains 20 classes, in FSOD, the data is randomly split into 15 base classes and 5 novel classes. There are three such splits and for each novel class there are $K = 1, 2, 3, 5, 10$ shots available. For this dataset we report the standard PASCAL VOC metric Average Precision (IoU=0.5) for novel classes (nAP50).

5.2. Implementation Details

Experiments are conducted with a standard Faster R-CNN [40], with a FPN [31]. All experiments are run on 4 GPUs with batch-size 16. We use a SGD optimiser with momentum 0.9 and weight decay 10^{-4}, except for models with Transformer based backbones, in which we use the AdamW optimiser with standard hyperparameters and weight decay 0.05, following [35]. The number of fine-tuning iterations is scaled depending on the dataset and the number of available shots.

We apply `RandomCrop` and `ColorJitter` augmentations when finetuning our FSOD detector on novel class data, unless stated otherwise. For experiments on MS-COCO, we include the `Mosaic` augmentation introduced in YOLOv4 [1]. This augmentation helps improve detection performance on "small" objects by stitching 4 images into a 2×2 grid. `Mosaic` is not used on PASCAL VOC experiments as the dataset does not contain the same scale variation as MS-COCO.

The number of neighbours k used in Label Verification (Section 4.2) is determined by the number

of novel instances, $k = \min\left(\left\lfloor\frac{K}{3}\right\rfloor + 1, 10\right)$. Hence for $K = 1, 2, 3, 5, 10, 30$, $k = 1, 1, 2, 2, 4, 10$. We choose a self-supervised DINO ViT-S/8 [4, 10] to construct the kNN classifier for verification, using the output CLS token as the chosen feature. For Box Correction (Section 4.3), we train a series of three box regression heads, where positive boxes are defined as those with IoU > $(0.3, 0.5, 0.7)$, respectively. This choice enables the correction of relatively poor initial bounding boxes.

After Candidate Sourcing, Label Verification and Box Correction, we significantly increase the number of available samples for novel categories, however it is inevitable that many novel instances will remain absent in our pseudo-annotations, and would still be treated as background. To avoid this issue, we introduce *"ignore regions"*, during end-to-end re-training, which are considered as neither foreground nor background, in practice, we treat all *unverified* novel class detections as ignore regions. We analyse these *"ignore regions"* in Appendix B.

5.3. Ablation Studies

We conduct ablation studies to investigate our design choices. The following experiments are considered: *first*, we demonstrate the importance of data augmentation to yield a stronger baseline model before any of the pseudo-labelling steps; *second*, we analyse several critical components of our method and conduct thorough ablation studies to validate their necessity, namely, Candidate Sourcing, Label Verification, Box Correction; *third*, we show that our proposed approach maintains performance for base class detections. Note that, all ablation experiments are conducted on the MS-COCO benchmark with $K = 30$ and a ResNet-50 backbone.

Importance of Augmentations: Given we only have access to a limited number of samples for novel categories at the starting point, maximising data efficiency before any pseudo-labelling is critical. Table 3 presents our observations. When comparing to TFA [48] as a baseline model (equivalent to Setting B1), applying ColorJitter, RandomCrop and Mosaic augmentations (Settings B2-B4) yields negligible performance improvements. Since almost all layers in TFA have been frozen during **Novel Training**, augmentations can only affect the classification layer of Faster R-CNN, Φ_{CLS}. The combination of these three augmentations (Setting B5), only gives a marginal improvement of 0.8nAP.

As explained in Section 3.2, we improve the **Novel Training** stage by also updating all RoI parameters and the RPN, *i.e.* $\Phi_{\text{RPN}}, \Phi_{\text{ROI}}$. With this simple change, all augmentations (Setting C2) substantially improve results, yielding a 3.2nAP boost over the TFA baseline model (Setting B1). In addition, we observe a noticeable improvement in performance on small novel instances (6.2 vs. 4.3nAPs)

from Mosaic augmentations, *e.g.* B5 vs. B1-B3, and C2 vs. C1. To further combat the overfitting issue, we add Dropout on RoI activations (Setting C3), yielding a small additional improvement of 0.4nAP.

Ablation of Pseudo-Labelling Steps: Table 4 shows the importance of our overall pseudo-labelling method and the contribution of each step. Using the Candidate Sourcing step only is equivalent to treating the naïve detections as pseudo-annotations, as done in other self-training works [34, 44]; this gives a marginal performance improvement of 1.8nAP. Removing the pseudo-annotations with incorrect class labels by Label Verification, yields an additional 3.4nAP performance boost. Lastly, Box Correction brings a 3.7nAP performance boost, in particular, such improvement is largely attributed to that from the stricter metric, *i.e.* nAP75, clearly showing the reduction of bounding box regression errors. Note that, the number and distribution of class labels for pseudo-annotations are identical before and after Box Correction; only the box coordinates of each pseudo-annotation have been changed.

Effect on Base Class Performance: While reading the performance on base classes from Table 4, we observe that, for the Baseline model, the improved performance on novel classes comes at the cost of performance on base classes, *e.g.*, bAP drops from 36.1 to 29.5. Our proposed pseudo-labelling method improves the detection of novel classes, while recovering performance on base classes.

5.4. Comparison to SotA

Existing methods include the meta-learning approaches: CGDP+FSCN [29], CME [27], TIP [26], DCNet [20], and two-phase training works: TFA [48], FCSE [45], Retentive R-CNN [13], SRR-FSD [59], DeFRCN [36], FSOD-UP [50], QA-FewDet [17]. In particular, we note that very few works report results on the base class detection performance, and methods like DeFRCN [36], can actually only detect novel classes and does not maintain the ability to detect base classes as with ours.

We report two sets of results for each task, Baseline which makes use of augmentations and the improved training outlined in Section 3 and Pseudo-Labelling which follows the method as outlined in Section 4.

MS-COCO Results are shown in Table 5. Using a ResNet-50 backbone, our Baseline method, which makes extensive use of augmentations and improved **Novel Training**, outperforms many existing works for $K = 30$, reflecting the importance of our findings in Section 3.2. When applying our Pseudo-Labelling method, we set new SotA performance for $K=30$, with a performance boost of up to 2.9, 6.1 and 10.4 for nAP, nAP50 and nAP75 metrics, respectively. When $K=10$, with a ResNet-101 backbone, our Pseudo-Labelling method achieves state-of-the-art or second-best performance in terms of nAP, nAP50,

Setting	Augmentations Color Jitter	Random Crop	Mosaic	Finetune Φ_{RPN}, Φ_{ROI}?	Φ_{ROI} Dropout?	nAP	nAPs
B1						13.0	5.5
B2	✓					13.2	5.6
B3		✓				13.2	5.6
B4			✓			12.8	6.1
B5	✓	✓	✓			13.8	6.8
C1	✓	✓		✓		14.5	4.3
C2	✓	✓	✓	✓		16.2	6.2
C3	✓	✓	✓	✓	✓	16.6	7.0

Table 3. Few-shot object detection performance on *novel* classes on the MS-COCO 30-shot task. Note that, the baseline model B1 is equivalent to a strong baseline proposed in TFA [48]. nAP – novel class average precision (nAPs is for small instances only)

Method	nAP	nAP50	nAP75	bAP	bAP50	bAP75
Baseline	16.6	30.9	15.8	29.5	46.7	32.4
+ Candidate Sourcing	18.4	34.0	18.2	31.7	53.2	35.6
+ Label Verification	21.8	40.3	21.1	31.6	53.4	35.2
+ Box Correction	25.5	42.0	27.3	33.3	53.6	36.1
Ideal Faster R-CNN [40]	43.5	67.4	46.6	36.1	56.0	39.2

Table 4. Ablation study on our pseudo-labelling method.

Method	Backbone	nAP 10	30	nAP50 10	30	nAP75 10	30	bAP 10	30
CGDP+FSCN [29]	ResNet-50	11.3	15.1	20.3	29.4	-	-	-	-
Ours (Baseline)		*11.4*	*16.6*	*21.2*	*30.9*	*11.1*	*15.8*	*28.4*	*29.5*
Ours (Pseudo-Labelling)		**17.6**	**25.5**	**30.9**	**42.0**	**17.3**	**27.3**	**29.7**	**33.3**
TFA w/ cos [48]	ResNet-101	10.0	13.7	19.1	24.9	9.3	13.4	*32.4*	*34.2*
FCSE [45]		11.9	16.4	-	-	10.5	16.2	-	-
Retentive R-CNN [13]		10.5	13.8	-	-	-	-	**39.2**	**39.3**
SRR-FSD [59]		11.3	14.7	23.0	29.2	9.8	13.5	-	-
CME [27]		15.1	16.9	24.6	28.0	*16.4*	*17.8*	-	-
DCNet [20]		12.8	18.6	23.4	32.6	11.2	17.5	-	-
TIP [26]		16.3	18.3	**33.2**	*35.9*	14.1	16.9	-	-
QA-FewDet [17]		11.6	16.5	23.9	31.9	9.8	15.5	-	-
FSOD-UP [50]		11.0	15.6	-	-	10.7	15.7	-	-
DeFRCN [36]		**18.5**	**22.6**	-	-	-	-	N/A	N/A
Ours (Baseline)		12.1	17.8	22.0	31.6	11.8	17.7	31.9	31.8
Ours (Pseudo-Labelling)		*17.8*	*24.5*	*30.9*	**41.1**	**17.8**	**25.0**	31.9	33.0
Ours (Baseline)	Swin-T	12.6	19.0	23.3	35.3	12.1	18.3	27.2	31.5
Ours (Pseudo-Labelling)		18.6	26.1	32.1	45.0	18.5	26.8	29.2	31.1
Ours (Baseline)	Swin-S	14.4	20.3	25.8	37.2	14.4	20.2	33.0	36.2
Ours (Pseudo-Labelling)		19.0	26.8	34.1	45.8	19.0	27.5	28.7	34.8

Table 5. Few-shot detection performance on the MS-COCO benchmark. We report performance on the 20 novel classes of MS-COCO in the FSOD setting. Best and second-best results, per backbone, are coloured **blue** and *red*, respectively. We also report performance on base classes.

nAP75. In terms of nAP, only DeFRCN outperforms our work. However, note that, DeFRCN is not able to detect base classes after training on novel categories, which strives for a different purpose than ours, we aim to *expand* our detector, rather than to *transfer* the detector.

In addition, while using a more powerful Transformer backbone models (Swin-T, Swin-S [35]), our proposed pseudo-labelling provides additional performance boosts.

PASCAL VOC Results are shown for the three standard novel splits in Table 6. Incorporating our proposed Pseudo-Labelling method is among the top 2 best performing models in all cases (except Novel Split 2 for $K=2$), and sets SotA performance for the majority of cases.

Notably, for Novel Split 3, our Pseudo-Labelling method achieves SotA for all K, with a boost of up to 4.7 nAP50, when using a ResNet-101 backbone. In many cases our weaker detection backbone (ResNet-50) already achieves SotA performance. Note that, once again the transferred detector DeFRCN outperforms ours in some cases.

5.5. Qualitative Results

In Figure 2, we present the qualitative results after each step of our pseudo-labelling procedure. The top row of Figure 2 shows examples of Label Verification (Section 4.2). The first three examples demonstrate the case in which the predicted class label from our detector matches kNN classification, and so the candidate detection is verified. The last three examples show the opposite case in which candidate detections are correctly rejected. The bottom row of Figure 2 shows examples of Box Correction (Section 4.3). The first three examples show very poor bounding boxes from

verified candidates (dashed blue boxes), which are drastically improved during the Box Correction step. The last three examples show acceptable bounding boxes from verified candidates (dashed blue boxes), also being improved with Box Correction. This demonstrates the ability of our Box Correction model to deal with a wide range of bounding box quality with respect to input candidate detections.

In Figure 3, we show precision-recall curves for some novel classes on the MS-COCO benchmark for $K=30$ using the stricter IoU=0.75 criterion. Our Pseudo-Labelling method substantially improves novel class performance, with improved precision and novel class recall. We note that for many novel classes, baseline models suffer from poor recall due to limited novel class annotations. This poor recall is improved by our pseudo-labelling method, however the poor recall of the baseline detector puts a limit on the diversity of pseudo-annotations for end-to-end retraining.

6. Conclusion

In this paper, we tackle the problem of few-shot object detection by training on pseudo-annotations. We present two novel methods to improve the precision of the pseudo-labelling procedure: *first*, we use the given K few-shot annotations to construct classifiers to verify class labels sourced from a baseline detector; *second*, we train a specialised box correction model to drastically improve the precision of pseudo-annotation bounding box coordinates. Our method generates a large number of high-precision pseudo-annotations with precise bounding boxes, removing the class imbalance issue in FSOD. This enables re-training of our detector *end-to-end*, alleviating the identified issues

Method/Shot	Backbone	Novel Split 1					Novel Split 2					Novel Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
CGDP+FSCN [29] *CVPR 21*		40.7	45.1	46.5	57.4	62.4	27.3	31.4	40.8	42.7	46.3	31.2	36.4	43.7	50.1	55.6
Ours (Baseline)	ResNet-50	37.7	42.0	50.3	57.0	58.0	19.8	22.8	35.6	42.7	44.2	33.9	36.0	38.6	49.8	51.6
Ours (Pseudo-Labelling)		50.5	53.1	56.4	61.7	62.7	*36.4*	33.8	46.1	49.3	48.2	*42.4*	44.3	49.1	*55.2*	*57.6*
TFA w/ cos [48] *ICML 20*		39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
FSCE [45] *CVPR 21*		44.2	43.8	51.4	61.9	*63.4*	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
Retentive R-CNN [13] *CVPR 21*		42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
SRR-FSD [59] *CVPR 21*		47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
CME [27] *CVPR 21*		41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
DCNet [20] *CVPR 21*	ResNet-101	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
TIP [26] *CVPR 21*		27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
QA-FewDet [17] *ICCV 21*		42.4	51.9	55.7	62.6	*63.4*	25.9	*37.8*	46.6	48.9	**51.1**	35.2	42.9	47.8	54.8	53.5
FSOD-UP [50] *ICCV 21*		43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5
DeFRCN [36] *ICCV 21*		*53.6*	*57.5*	*61.5*	*64.1*	60.8	30.1	**38.1**	*47.0*	**53.3**	47.9	**48.4**	*50.9*	*52.3*	54.9	57.4
Ours (Baseline)		36.0	40.1	48.6	57.0	59.9	22.3	22.8	39.2	44.2	47.8	34.3	43.4	42.9	52.0	54.5
Ours (Pseudo-Labelling)		**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**

Table 6. Few-shot detection performance across the three splits on the PASCAL VOC benchmark. Best and second-best results are coloured **blue** and *red*, respectively. Please refer to the text for discussion.
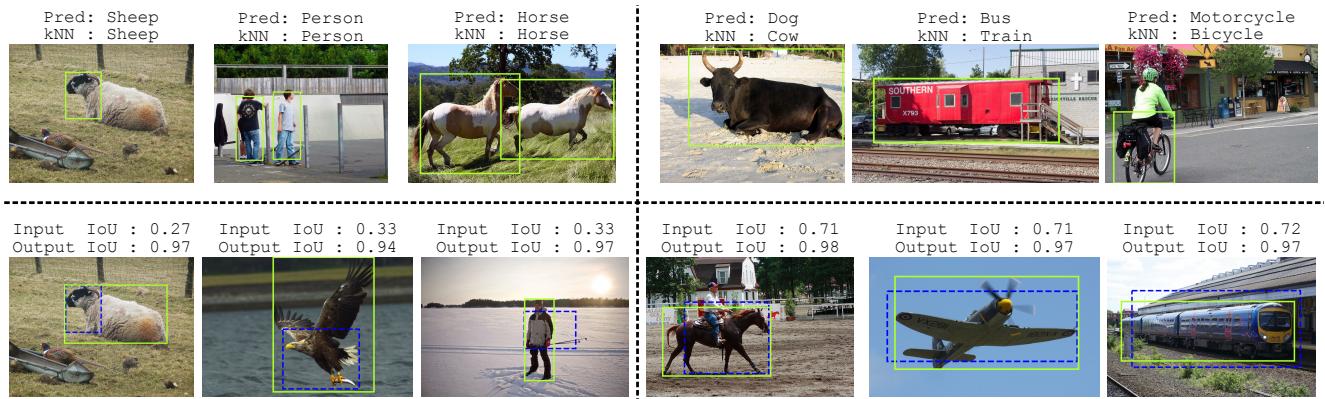

Figure 2. Top Left: Predicted instances which are verified during *Label Verification*; the predicted class labels from our baseline detector and our *k*NN classifier match. Top Right: Predicted instances which are rejected during *Label Verification*; the predicted class labels from our baseline detector (false positive) and the *k*NN *do not* match. Bottom Left: Verified bounding boxes with very poor quality (blue dashed) are drastically improved (lime solid) during *Box Correction*. Bottom Right: Verified bounding boxes which are acceptable (blue dashed) are further improved (lime solid).
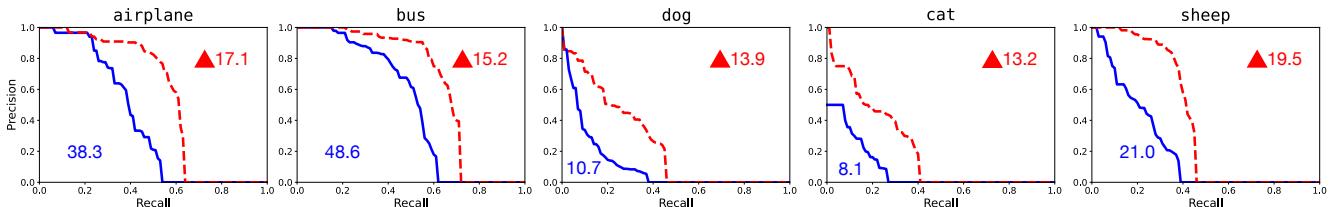

Figure 3. Precision-Recall curves (using the stricter IoU=0.75 criterion) for K=30 on MS-COCO showing Baseline performance (solid blue) and the substantial performance boost after making use of our Pseudo-Labelling method (dashed red). Our Pseudo-Labelling method yields improved precision and improved recall for novel classes.

around "supervision collapse" in few-shot object detectors. Furthermore, we have illustrated the importance of augmentations for FSOD, this was previously under-explored, despite augmentations being a key part of preventing overfitting. Our method achieves state-of-the-art or second-best performance performance on both PASCAL VOC and MS-COCO benchmarks, across all number of shots.

7. Acknowledgements

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 5

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162. IEEE Computer Society, 2018. 5

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 5

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 5, 6

[5] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. 2018. 2

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[7] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007. 4

[8] Xiyang Dai, Yinpeng Chen, Bin (Leo) Hsiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2021. 1

[9] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems*, 2020. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations*, 2021. 5, 6

[11] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015. 1, 5, 24

[12] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[13] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4527–4536, 2021. 1, 2, 6, 7, 8

[14] Ross Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision*, 2015. 2

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 2

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 24

[17] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the International Conference on Computer Vision*, pages 3263–3272, October 2021. 1, 2, 6, 7, 8

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 5

[19] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015. 3

[20] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10185–10194, 2021. 2, 6, 7, 8

[21] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, 2019. 2

[22] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the International Conference on Computer Vision*, pages 8420–8429, 2019. 1, 2, 3, 5

[23] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 2020. 2

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[26] Aoxue Li and Zhenguo Li. Transformation invariant few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3094–3102, June 2021. 2, 6, 7, 8

[27] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2021. 2, 6, 7, 8

[28] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the International Conference on Computer Vision*, 2019. 2

[29] Yiting Li, Haiyue Zhu, Yu Cheng, Wenxin Wang, Chek Sing Teo, Cheng Xiang, Prahlad Vadakkepat, and Tong Heng Lee.

Few-shot object detection via classification refinement and distractor retreatment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15395–15403, June 2021. 2, 6, 7, 8

[30] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 1, 5, 24

[31] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the International Conference on Computer Vision*, 2017. 1, 2

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016. 2

[34] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *Proceedings of the International Conference on Learning Representations*, 2021. 2, 6

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the International Conference on Computer Vision*, pages 10012–10022, October 2021. 5, 7

[36] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proceedings of the International Conference on Computer Vision*, pages 8681–8690, October 2021. 1, 2, 6, 7, 8

[37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[38] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[39] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2016. 1, 2, 3, 4, 5, 7

[41] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *Winter Conference on Applications of Computer Vision*, 2005. 2

[42] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, pages 363–371, 1965. 3

[43] Linda B. Smith and Lauren K. Slone. A developmental ap-proach to machine learning? *Frontiers in Psychology*, 2017. 1

[44] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2, 6

[45] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2021. 1, 2, 5, 6, 7, 8

[46] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the International Conference on Computer Vision*, 2019. 2

[48] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *Proceedings of the International Conference on Machine Learning*, 2020. 2, 3, 4, 6, 7, 8, 12, 14

[49] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the International Conference on Computer Vision*, pages 9925–9934, 2019. 2

[50] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *Proceedings of the International Conference on Computer Vision*, pages 9567–9576, October 2021. 2, 6, 7, 8

[51] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Proceedings of the European Conference on Computer Vision*, 2020. 1

[52] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

[53] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[54] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[55] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. 2021. 1

[56] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the International Conference on Computer Vision*, October 2019. 1, 2

[57] Ze Yang, Yali Wang, Xianyu Chen, Jianzhuang Liu, and Yu Qiao. Context-transformer: Tackling object confusion for few-shot detection. 2020. 2

[58] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Ob-

jects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[59] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8782–8791, June 2021. 2, 6, 7, 8

[60] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pretraining and self-training. *Advances in Neural Information Processing Systems*, 2020. 3

Appendices

A. Supervision Collapse

In Section 3 of the main paper, we experimentally demonstrate a form of "supervision collapse" in few-shot object detection; detector features trained *only on base class data* are naturally biased against novel instance detection. In this section we demonstrate two aspects of supervision collapse.

A.1. On Generalisability of RPNs

Figure 4 qualitatively illustrates the manifestation of "supervision collapse" in the Region Proposal Network (RPN) of a Faster R-CNN detector. We note that an RPN trained only with base class data struggles to propose boxes for the novel instances, due to these novel instances having been treated as background during **Base Training**.

We find that training RPN specific parameters (2 convolutional layers) on limited novel class data is adequate to drastically improve the proposal of object regions containing novel instances. This improvement is shown qualitatively in Figure 4 and quantitatively in Table 1 of the main paper.

A.2. On Transferability of Base Features

Figure 5 qualitatively demonstrates the other manifestation of "supervision collapse"; after training the second stage of Faster R-CNN as in [48] *i.e.* only training the final classification layer on novel class data, features in the second stage of the detector (which take RPN proposals as input), incorrectly classify Regions-of-Interest (RoIs) containing novel instances as "background", *i.e.* the detector features in the second stage do not contain discriminative information to classify most novel instances, even in the cases where the object regions proposed by the RPN are very good.

We find that training parameters specific to the second stage of the detector, (Φ_{CLS}, Φ_{ROI}), can moderately alleviate this form of "supervision collapse" such that the resultant detector can be used for Candidate Sourcing in our Pseudo-Labelling method. Figure 5 shows examples in which RoIs are correctly classified after *our* **Novel Training**.

Figure 4. **Supervision collapse for the RPN.** Top 100 RPN proposals before and after **Novel Training**. In each pair of rows, the upper row shows the top 100 object proposals (blue) of an RPN trained *only on base class data* and the lower row shows the top 100 object proposals (blue) after RPN specific parameters (Φ_{RPN}) have been trained on limited novel class data. Groundtruth novel instances are shown in dashed lime. We find an RPN trained only on base class data cannot propose many simple novel instances and that training RPN specific parameters on limited novel class data leads to an RPN which is substantially better at proposing object regions containing novel instances.
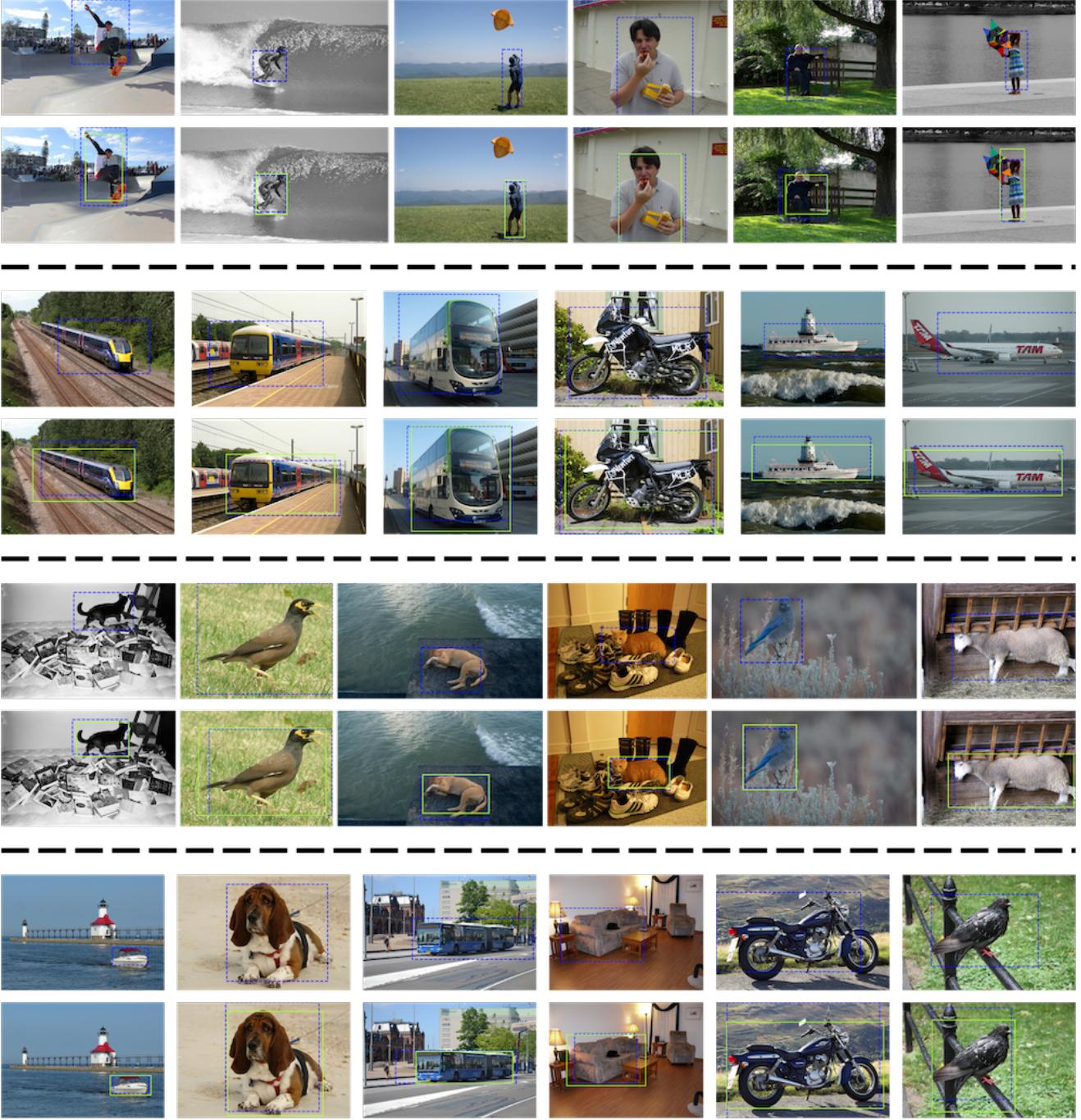
Figure 5. **Supervision collapse for RoI classification.** Comparing RoI classification between TFA [48], where only Φ_{CLS} is trained on limited novel class data, with *our* **Novel Training**, which additionally trains Φ_{ROI} on limited novel class data, for cases where the object proposal is very good. In each pair of rows, the upper row shows the best object region (dashed blue) proposed by the RPN in TFA and the lack of foreground detection (false negative) from the second stage, and the lower row shows the best object region (dashed blue) proposed by the RPN after *our* **Novel Training** and the correct foreground detection (solid lime). We find that TFA incorrectly classifies positive (with respect to novel classes) object regions (dashed blue) as background, whereas *our* **Novel Training** allows positive object regions (dashed blue) to be correctly classified as a novel instance (solid lime).

B. Ignore Regions

As mentioned in Section 5.2 of the main paper, we introduce the "*ignore regions*" during end-to-end training after making use of our Pseudo-Labelling method. Figure 6 shows the impact of ignore regions. During end-to-end training, we ensure no RoIs which have IoU> 0.5 with an ignore region are sampled for training the second-stage of the detector; RoIs which have IoU> 0.5 with an ignore region are neither pushed into foreground (positive) or background (negative) in the second-stage of the detector. Only with respect to the RPN during end-to-end training, "*ignore regions*" are treated as foreground *i.e.* the RPN after end-to-end training has learnt to propose regions-of-interest covering ignore regions. Using ignore regions prevents the undiscovered novel instances from being treated as background during end-to-end training (as occurs during **Base Training**) – there are undiscovered novel instances because we use a score threshold of 0.8 during Candidate Sourcing to yield high-precision pseudo-annotations. We use all detections from our baseline detector, with novel class labels, which are *unverified* after Label Verification as ignore regions – this includes detections with score less than 0.8.

An ablation study on the utility of ignore regions is presented in Table 7, using the MS-COCO 30-shot object detection benchmark. Using ignore regions improves performance by 4.4, 3.5 and 4.7 on novel categories, *i.e.* nAP, nAP50 and nAP75, respectively, when compared to not using ignore regions.

Method	Ignore Regions?	Metrics		
		nAP	nAP50	nAP75
Ours (Pseudo-Labelling)	✗	21.1	38.5	22.6
Ours (Pseudo-Labelling)	✓	25.5	42.0	27.3

Table 7. Ablation study on the utility of ignore regions.

Figure 6. Comparing images in the training set with and without ignore regions. In each pair of rows, the upper row shows images with pseudo-annotations found by our Pseudo-Labelling method (solid lime), with many novel instances not covered, the lower row shows the same images additionally showing the ignore regions (solid blue). Using ignore regions covers many novel instances which are not found by our Pseudo-Labelling method, preventing these novel instances being treated as background during end-to-end training.

16

C. More Qualitative Results

In this section we present more qualitative results from our Pseudo-Labelling method on the MS-COCO benchmark with $K=30$.

C.1. Label Verification

In Figure 7, we present additional qualitative results after the *Label Verification* step of our Pseudo-Labelling method. The first four rows in Figure 7 show examples of Label Verification (Section 4.2 of the main paper) in which the predicted class label from our baseline detector (after **Novel Training**) matches the class label from our kNN classifier. These candidate detections are *verified* and used as inputs during *Box Correction* after which they are included as pseudo-annotations for end-to-end training. The final two rows in Figure 8 show examples of Label Verification in which the predicted class label from our baseline detector *does not* match the class label from our kNN classifier. These candidate detections are *not verified* and rejected from our Pseudo-Labelling method.

C.2. Box Correction

In Figure 8, we present additional qualitative results after the *Box Correction* step of our Pseudo-Labelling method. The first five rows in Figure 8 show examples of Box Correction (Section 4.3 of the main paper) in which the bounding box coordinates of *verified* candidate detections are poor (dashed blue) *i.e.* they have low intersection-over-union (IoU) with the groundtruth instance. During *Box Correction* these poor bounding boxes are drastically improved to near perfect bounding boxes (solid lime). The bottom row in Figure 7 show examples of Box Correction in which the bounding box coordinates of the verified candidate detections (dashed blue) have acceptable IoU with groundtruth instances. During *Box Correction* these acceptable bounding boxes are further improved to near perfect bounding boxes (solid lime).
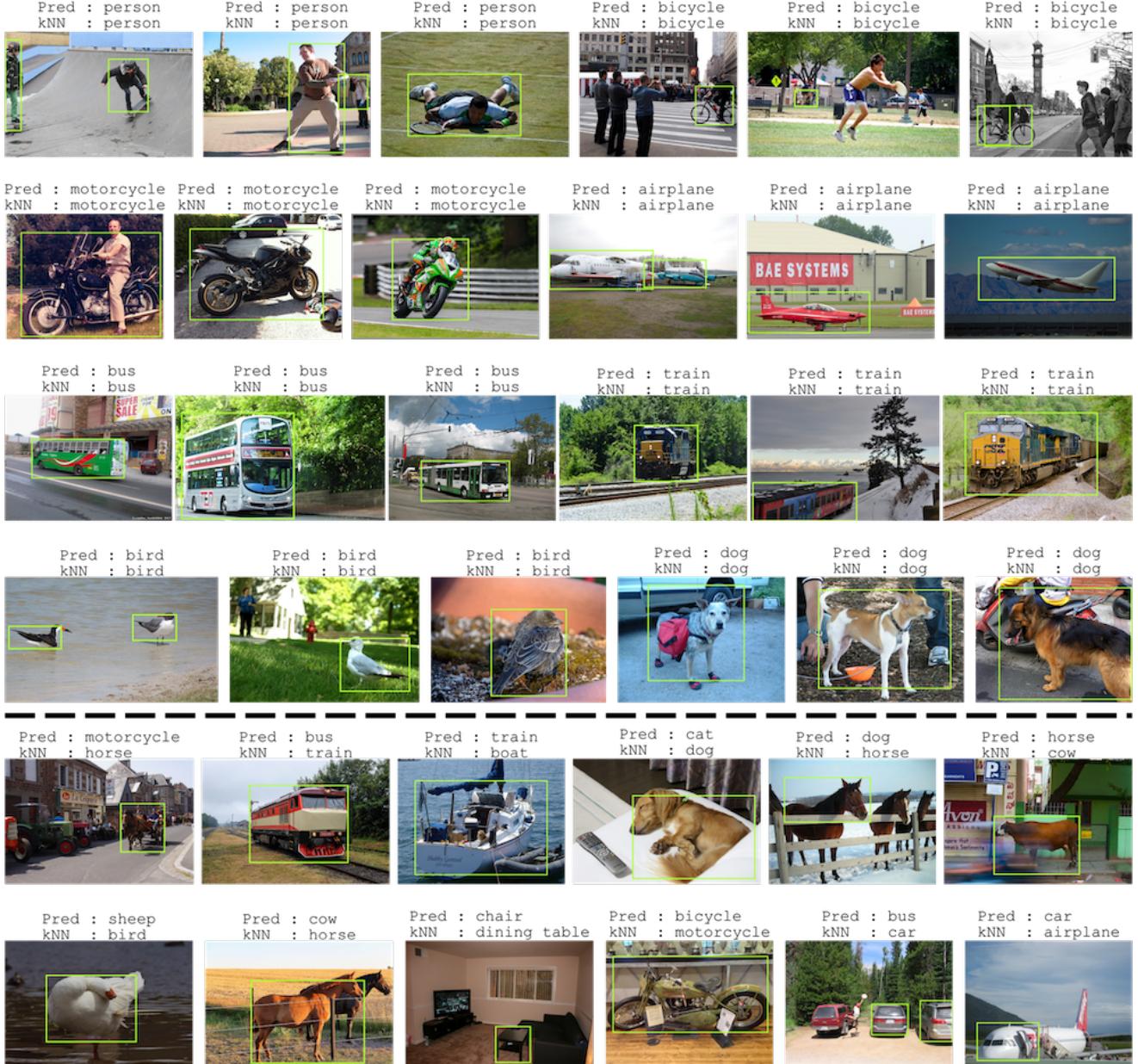
Figure 7. More examples of *Label Verification*. Top Four Rows: Candidate detections which *are verified* during *Label Verification*; the predicted class labels from our baseline detector and our kNN classifier match. Bottom Two Rows: Candidate detections which *are not verified* (or rejected) during *Label Verification*; the predicted class labels from our baseline detector (false positive) and our kNN classifier *do not* match.

Figure 8. More examples of *Box Correction*. First Five Rows: Verified bounding boxes with poor quality/IoU with groundtruth (dashed blue) are drastically improved (solid lime) during *Box Correction*. Bottom Row: Verified bounding boxes with acceptable quality/IoU with groundtruth (dashed blue) are further improved (solid lime) during *Box Correction*.

D. Quantitative Results

In this section we present quantitative results of each step of our Pseudo-Labelling method on the MS-COCO benchmark with $K=30$. We show precision-recall curves using the various IoU criteria used in MS-COCO.

Precision-recall curves when using the IoU=0.5, IoU=0.75 and IoU=0.5:0.95 MS-COCO criteria are shown in Figure 9,10,11, respectively.

For the vast majority of novel classes, adding our Pseudo-Labelling method improves performance over our baseline detector performance, regardless of which MS-COCO criteria is used. However, we note that the effect of each step is not uniform across different novel classes and metrics. For example, considering the `person` class, using Candidate Sourcing only is the best performing model when using IoU=0.5 due to higher recall, but when using IoU=0.75 and IoU=0.5:0.95 our full Pseudo-Labelling method performs best.

In terms of IoU=0.5:0.95, which is the main MS-COCO criterion, Figure 9 shows that different classes see the largest performance boost from different steps in our Pseudo-Labelling process. For example, the largest performance boost comes from Label Verification from some classes *e.g.* `bird`, `sheep`, `cat`; these classes suffer from many false positives with respect to class labels after Candidate Sourcing, which are removed during Label Verification yielding better performance. On the other hand, for other classes the largest performance boost comes from Box Correction *e.g.* `airplane`, `train`, `person`; these classes suffer from many false positives with respect to bounding box coordinates after Candidate Sourcing, which are corrected during Box Correction yielding better performance.

We note that in some rare cases an individual Pseudo-Labelling step may harm performance, *e.g.* adding Label Verification for `person` reduces performance by 2.4nAP compared to using Candidate Sourcing only, and adding Box Correction for `bicycle` reduces performance by 1.5nAP compared to using Candidate Sourcing and Label Verification only.

Moreover, we note that our Pseudo-Labelling method yields negligible performance improvements for the `bicycle` and `chair` classes yielding only a 0.1 and 0.3nAP boost, respectively, over our baseline detector.
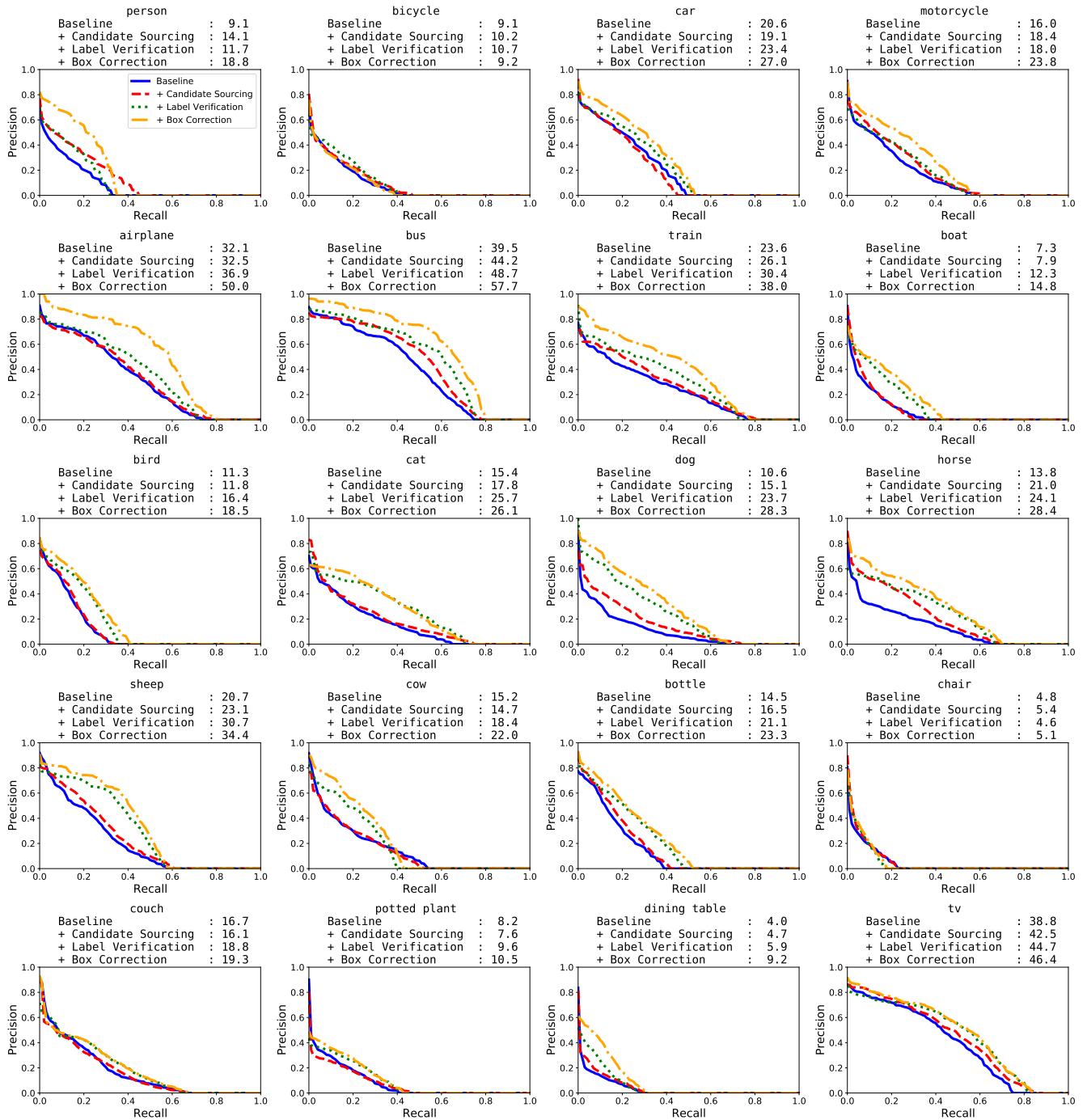
Figure 9. **Precision-Recall curves for novel classes with IoU=0.5:0.95.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP value for each step of our method is shown in the title of each subplot.
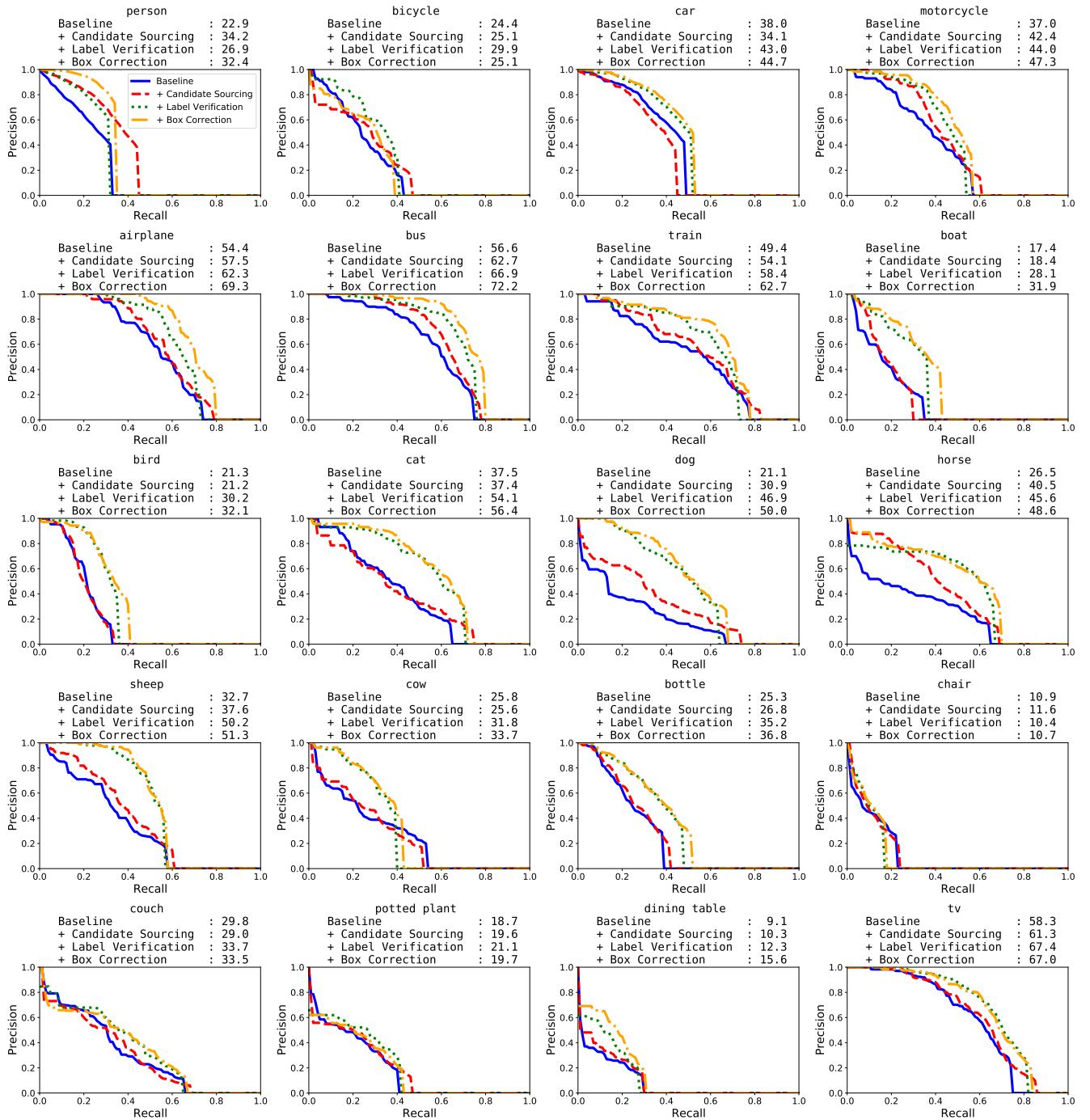
21

Figure 10. **Precision-Recall curves for novel classes with IoU=0.5.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP50 value for each step of our method is shown in the title of each subplot.
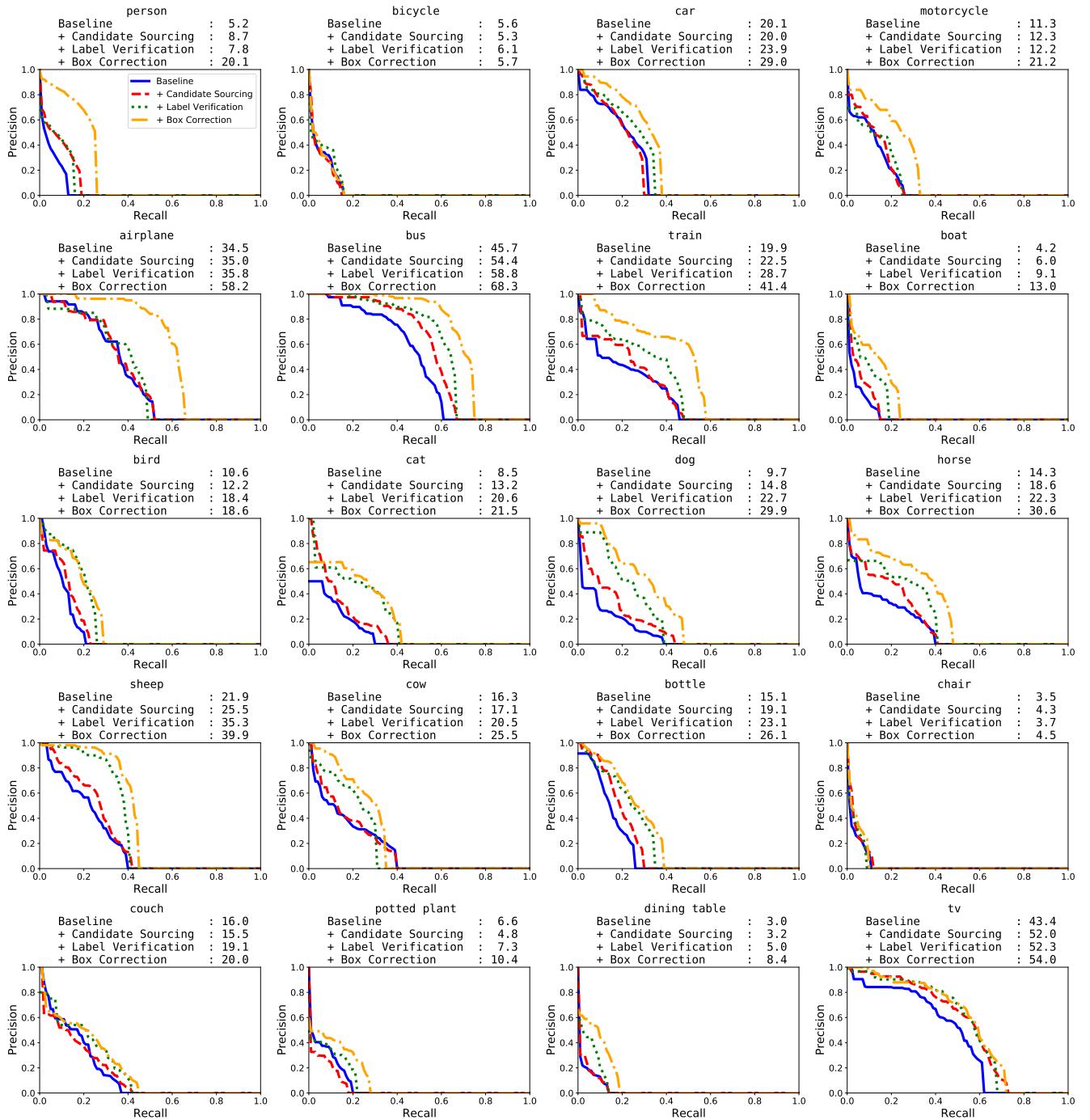
Figure 11. **Precision-Recall curves for novel classes with IoU=0.75.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP75 value for each step of our method is shown in the title of each subplot.

E. Limitations

There are two main limitations of our work: *first*, our work follows prior work and performs benchmark experiments on MS-COCO [30] and PASCAL VOC [11], which contain 80 and 20 classes, respectively. In future work, we will benchmark the idea on LVIS [16], which contains 1230 classes, and better characterise few-shot object detection methods. *Second*, in this work, we only perform one iteration of pseudo-labelling with kNN classifier. Given our pseudo-annotations are noisy, one can make use of noisy label learning to effectively incorporate our pseudo-annotations as training data for training more powerful classifiers in the Label Verification step and obtaining more pseudo-annotations.

F. Ethical Concerns

Our proposed method allows object detectors to detect novel categories with limited access to groundtruth annotations. Moreover, our work contains a Pseudo-Labelling method which enables object detectors to effectively re-train on (noisy) pseudo-annotations. Our work could be applied to numerous areas *e.g.* public surveillance and autonomous driving. While not our intention, our method could be used by agents with malicious purposes for detection with limited data. We think robust public debate and suitable regulations can reduce the risk of malicious applications of our work.