MINI: MINING IMPLICIT NOVEL INSTANCES FOR FEW-SHOT OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-Shot Object Detection (FSOD) aims to detect novel concepts given abundant base data and limited novel data. Recent advances propose an offline mining mechanism to discover implicit novel instances, which exist in the base dataset, as auxiliary training samples to retrain a more powerful model. Nonetheless, the offline mined novel instances remain unchanged during retraining, thus hindering further improvements. A straightforward alternate adopts an online mining mechanism that employs an online teacher to mine implicit novel instances on the fly. However, the online teacher relies on a good initialization which is non-trivial in the scenarios of FSOD. To overcome the obstacles, we present ***M**ining **I**mplicit **N**ovel **I**nstances (MINI)*, a framework that unifies the offline mining mechanism and online mining mechanism with an adaptive mingling design. In **offline mining**, MINI leverages an offline discriminator to collaboratively mine implicit novel instances with a trained FSOD model. In **online mining**, MINI takes a teacher-student framework to simultaneously update the FSOD network and the mined implicit novel instances on the fly. In **adaptive mingling**, the offline and online mind implicit novel instances are adaptively combined, where the offline mined novel instances warm up the early training and the online mined novel instances gradually substitute the offline mined instances to further improve the performance. Extensive experiments on PASCAL VOC and MS-COCO datasets show MINI achieves new state-of-the-art performance on any shot and split of FSOD tasks. All code will be made available.

1 INTRODUCTION

Few-Shot Object Detection (FSOD), which aims to train an object detector with abundant data on base classes and few shot samples on novel classes, has invoked great interest. Current FSOD methods mostly follow a *pretrain-transfer* paradigm, which first pre-trains the object detector on the base classes and then finetunes the model with few shot samples of novel classes with freezing most its parameters. Although various methods have been proposed following this paradigm, including meta-learning (Yan et al., 2019; Xiao & Marlet, 2020; Fan et al., 2020), metric learning (Karlinsky et al., 2019; Li et al., 2021a), and fine-tuning (Wang et al., 2020; Cao et al., 2021; Sun et al., 2021), their performance are limited in two aspects. (1) *Data Scarcity*. The scarce novel samples fail to provide a sufficient diversity of novel classes, making FSOD models tend to overfit few shot samples. (2) *Implicit Novel Instances*. Due to the co-occurrence between base and novel classes on benchmark datasets, the object detector pre-trained on base classes is learned to treat the co-occurred novel instances as backgrounds.

Motivated by these observations, recent advances (Li et al., 2021b; Kaul et al., 2022) propose to discover and leverage these implicit novel instances, which exist in the base dataset, with an **offline mining** mechanism as in Fig. 1(a). It first trains an FSOD model to mine implicit novel instances, which will be verified by an external model and then be taken as auxiliary training samples to re-train the detector. In this way, the enriched training samples greatly promote the discriminative ability of the network on novel classes. Despite substantial progress, the offline mining mechanism is imperfect, where the mined novel instances are sourced from a noisy FSOD detector and remain unchanged during re-training. It lacks a mechanism to upgrade the mined novel instances as the network improves during re-training, hindering further performance improvement.

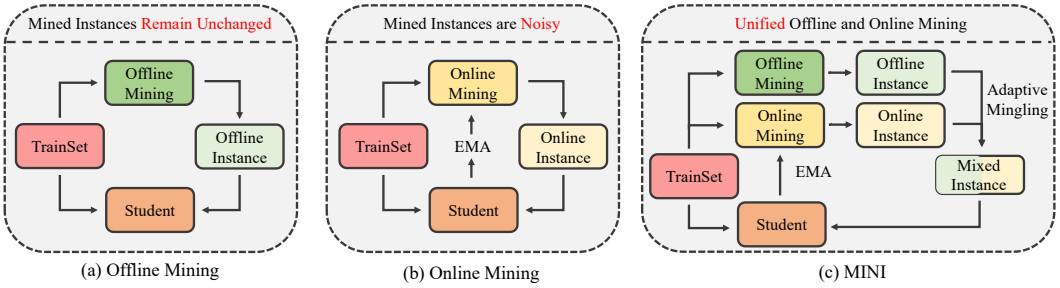

Figure 1: Comparison between different mining mechanisms. (a) Offline mining. The offline teacher first mines implicit novel instances from the train set, which will be verified by an external model and then be taken as auxiliary training samples to re-train the student model. These mined novel instances remain unchanged during re-training. (b) Online mining. It first trains a FSOD model to initialize the teacher and student model. The online teacher mines implicit novel instances at each training iteration, and the weight of the teacher model is updated by the student via EMA. Because the teacher is initialized from a poor-learned FSOD detector, the mined novel instances can be very noisy. (c) MINI. MINI unifies an offline mining mechanism and an online mining mechanism via an adaptive mingling design. The offline mined instances warm up the early training of the online teacher. The online mined novel instances gradually substitute the offline instances to further improve the performance.

A straightforward alternate is performing pseudo labeling on the fly, named **online mining** mechanism, as shown in Fig.1(b). Similar to Semi-Supervised Object Detection (SSOD) (Liu et al., 2021), it first trains a FSOD model to initialize the teacher and student model. The online teacher then mines novel instances at each training iteration as pseudo labels to supervise the student model, and the weight of the teacher model is updated by the student model via EMA (Kingma & Ba, 2014). However, we observe that it is challenging to apply the above online mining mechanism in the scenario of FSOD. Due to the limited performance of the initial FSOD model on novel classes, the online teacher that initialized from such a poor-learned model mines very noisy novel instances at the beginning, hence diverging the training.

Towards the aforementioned drawbacks, this paper proposes a framework called ***Mining Implicit Novel Instances (MINI)*** that significantly improves the FSOD performance via mining reliable implicit novel instances as pseudo labels. The contributions of MINI are two-fold. First, MINI unifies the **offline mining** mechanism and the **online mining** mechanism with an **adaptive mingling** design, as shown in Fig. 1(c). Second, MINI introduces several effective designs to improve the quality of mined implicit novel instances in both offline and online mining mechanisms. Specifically, 1) in the offline mining mechanism, MINI leverages an offline discriminator to calibrate the classification confidences on each novel class of the initial few-shot detector. An adaptive threshold scheme is then applied to find a class-wise threshold to filter unreliable mined instances. 2) In the online mining mechanism, MINI takes a teacher-student framework to simultaneously update the FSOD network and the mined implicit novel instances on the fly. An IoU branch is adopted to distinguish the mined novel instances that are precisely localized. 3) In the adaptive mingling design, the offline mined and online mined implicit novel instances are adaptively combined, where the offline mined novel instances warm up the early training and the online mined novel instances gradually substitute the offline mined instances to further improve the performance.

We extensively evaluate MINI on Pascal VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) benchmarks, and achieve new SOTA performance for all settings. Concretely, we improve the current SOTA performance (novel AP50) by 8.6, 7.9, 4.7, 5.3, 8.1 and 9.7, 7.1, 8.9, 6.8, 9.9 and 8.9, 8.2, 5.0, 4.7, 6.7 for K=1, 2, 3, 5, 10 on novel split 1, 2 and 3, respectively. Even on the challenging COCO split, we push the limit of the envelope performance (novel mAP) by 2.4 and 2.8 for $K = 10$ and 30, respectively, which demonstrates the effectiveness of the proposed MINI.

2 Related Work

2.1 Few-Shot Object Detection

Few-Shot Object Detection(FSOD) aims to detect novel concepts given abundant base data and limited novel data. One main line of FSOD methods is meta-learning based approaches (Karlin-

sky et al., 2019; Kang et al., 2019; Yan et al., 2019; Xiao & Marlet, 2020; Fan et al., 2020; Li et al., 2021a; Zhang et al., 2021; Hu et al., 2021). FSRW (Kang et al., 2019) and Meta R-CNN (Yan et al., 2019) introduce feature re-weighting to one-stage and two-stage detectors, respectively. Meta-Det (Wang et al., 2019) disentangles the learning of category-specific and category-agnostic components. FSIW (Xiao & Marlet, 2020) improves FSRW (Kang et al., 2019) with more complex feature aggregation module and unify few-shot object detection and viewpoint estimation. The second line is fine-tuning based approaches (Wang et al., 2020; Sun et al., 2021; Li et al., 2021b; Fan et al., 2021; Cao et al., 2021; Qiao et al., 2021). TFA (Wang et al., 2020) firstly introduces a simple base-training and few-shot fine-tuning paradigm. FSCE (Sun et al., 2021) improves the TFA baseline by fine-tuning more layers and brings batch contrastive learning to FSOD. FADI (Cao et al., 2021) divides the fine-tuning stage into association and discrimination to attain a more discriminative classifier. DeFRCN (Qiao et al., 2021) devises GDL and PCB to alleviates the potential contradictions of Faster R-CNN (Ren et al., 2015) in FSOD. Another line is mining-based approaches, which aim to discover and utilize the implicit novel instances in the base dataset. (Li et al., 2021b) proposes a semi-supervised distractor utilization loss to assign positive gradients to these implicit novel instances. (Kaul et al., 2022) performs a pseudo-labeling technique to offline mine implicit novel samples and take them as extra training samples. MINI significantly boosts the performance and differs from (Kaul et al., 2022) with several key designs: 1) (Kaul et al., 2022) follows an offline mining mechanism, while MINI unifies the offline and online mining mechanisms with an adaptive mingling design. 2) (Kaul et al., 2022) leverages an offline discriminator to verify and eliminate the misclassified mined novel instances, while our offline discriminator co-mining can help to mine missed novel instances by the FSOD model. 3) (Kaul et al., 2022) adopts a fixed threshold for all novel classes, while MINI employs an adaptive threshold that varies across different novel classes.

2.2 Semi-Supervised Object Detection

Semi-Supervised Object Detection (SSOD) aims to train a detector with limited labeled data and abundant unlabeled data. There are two lines of methods, the consistency methods (Jeong et al., 2019; Tang et al., 2021a) and pseudo label methods (Sohn et al., 2020; Tang et al., 2021b; Liu et al., 2021; Zhou et al., 2021; Xu et al., 2021). CSD (Jeong et al., 2019) enforces a consistency loss between the original image and the horizontally flipped one. STAC (Sohn et al., 2020) proposes a simple pseudo-labeling framework, which trains the model with highly confident pseudo labels with strong augmentations. Unbiased Teacher (Liu et al., 2021) finds the bias existed in pseudo labels due to over-fitting and class imbalance, hence introducing EMA and Focal Loss (Lin et al., 2017b) to resolve them. There are many subsequent variants (Tang et al., 2021b; Zhou et al., 2021; Xu et al., 2021). The online mining mechanism in MINI shares similar ideas with pseudo labeling methods, but it is not feasible to directly apply SSOD methods. Due to the data scarcity and implicit novel instances problem, the poor-learned teacher model cannot well discover potential novel instances. Moreover, SSOD methods usually rely on a heuristics confidence threshold which fails to guarantee the quality of mined novel instances in FSOD scenario. Hence we propose *MINI* to better tackle it.

3 Our Approach

In the conventional few-shot object detection (FSOD), there exist two non-overlapping training sets, *i.e.*, a base dataset D^b and a novel dataset D^n. There are abundant images $\{x_i^b\}$ in the base dataset D^b. Each base image x_i^b has exhaustively annotated bounding boxes $B_i^b = \{b_{ij}^{c^b}\}$ of each base class $c^b \in C^b$, where $b_{ij}^{c^b}$ indicates the j-th bounding box of the base class c^b on the image x_i^b. In the novel dataset D^n, there are K-shot annotated sample pairs $\{(x_k^{c^n}, b_k^{c^n})\}$ for each novel class $c^n \in C^n$, with $k \leq K$. Specifically, $(x_k^{c^n}, b_k^{c^n})$ indicates the k-th sample that comprises of an image and a box in novel class c^n. The ultimate goal of FSOD is to optimize a robust detector to detect objects in a test set that comprises both classes in $C^b \cup C^n$.

In this work, we propose ***Mining Implicit Novel Instances (MINI)***, a framework that significantly improves the FSOD performance via mining reliable implicit novel instances as auxiliary training samples of novel classes. As shown in Fig. 2, we first train a FSOD model as an initial miner (Sec. 3.1). In the **offline mining** mechanism (Sec. 3.2), MINI leverages an offline discriminator to collaboratively mine implicit novel instances with the initial FSOD model. The unreliable mined

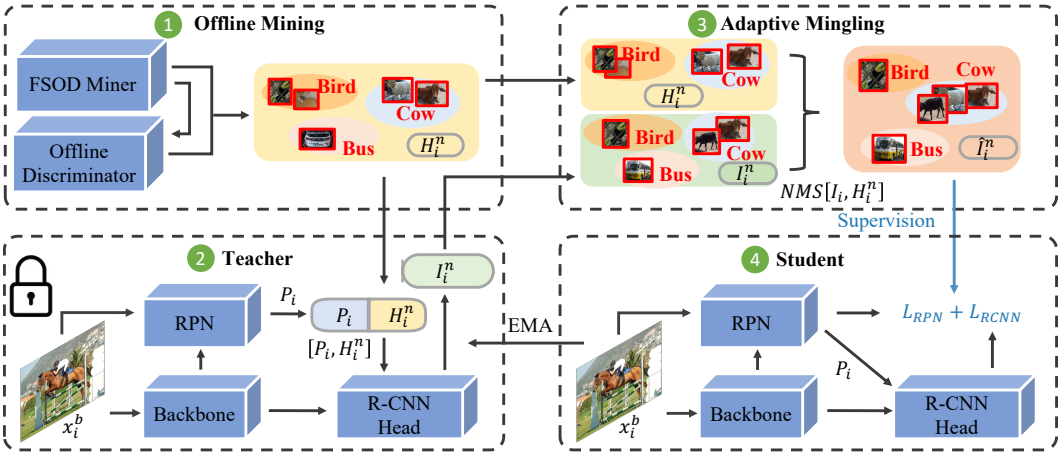

Figure 2: **Method Overview**. *MINI* mines implicit novel instances with unified offline and online mining mechanisms. The pipeline of *MINI* is following: 1) An FSOD detector is used to discover initial implicit novel instances. The offline mining mechanism leverages an offline discriminative model to calibrate classification confidences of these mined novel instances. 2) In the online mining mechanism, the teacher model mines implicit novel instances at each training iteration. 3) The offline and online mined novel instances are combined with an adaptive mingling design. 4) The student model takes implicit novel instances as ground-truths and updates the parameters of the teacher model via EMA.

novel instances will be filtered out by a class-wise adaptive threshold scheme. In the **online mining** mechanism (Sec. 3.3), MINI employs a teacher-student framework to simultaneously update the FSOD network and mine implicit novel instances on the fly. In **adaptive mingling** design (Sec. 3.3), MINI combines and adaptively balances the offline and online mined instances, where the offline mined novel instances dominate the early stage of training, while the proportion of the online mined novel instances gradually increases.

3.1 FSOD Detector as Initial Miner

In this section, we aim to obtain an object detector that has some basic ability to recognize novel classes. The initial FSOD network can be readily instantiated with different FSOD algorithms. For simplicity, we adopt the widely used TFA (Wang et al., 2020) in this work, which divides the whole training pipeline into two independent stages as follows,

Base Model Training Stage In the first base training stage, the whole model, including the box predictors (*i.e.*, the classifier and regressor) and the feature extractor (*i.e.*, the rest of the network) are jointly trained on the base dataset D^b with abundant annotations of base classes. To this end, the base model learns a general feature representation ability and is ready to transfer to novel classes.

Few-Shot Fine-tuning Stage In the second few-shot fine-tuning stage, only the box predictor is fine-tuned on a small balanced training set that comprises both base and novel classes. The feature extractor will be frozen to preserve the pre-trained general knowledge and prevent the potential over-fitting on the scarce novel set.

3.2 Offline Mining Mechanism

After initializing, a FSOD model M^s that can detect novel categories. In this section, we aim to discover implicit novel instances from the base dataset D^b with M^s in an offline manner. Specifically, we perform inference of M^s over each base set image $x_i^b \in D^b$. The mining process can be formulated as follows,

$$P_i = \phi^{RPN}(x_i^b)\ , \quad Y_i^n = \phi^{RCNN}(x_i^b, P_i)\ , \quad H_i^n = \{y_{ij}^n | s_{ij} >= \delta\}\ . \tag{1}$$

The RPN ϕ^{RPN} first predicts a set of proposals P_i, the R-CNN ϕ^{RCNN} classifies and regresses each proposal $p_{ij} \in P_i$, then some post-processing procedures, *e.g.*, NMS, are applied to yield the inference results $Y_i^n = \{(s_{ij}, b_{ij}, l_{ij})\}$ on novel classes C^n, where s_{ij}, b_{ij}, l_{ij} denotes the predicted score, bounding box and label of j-th candidate instance on i-th image, respectively. A score

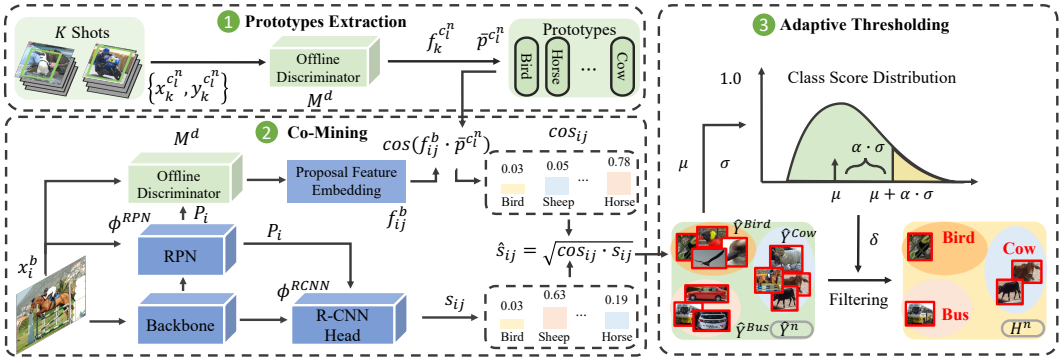

Figure 3: Pipeline of offline mining mechanism. Offline discriminator first extracts class prototypes from D^n. The FSOD model performs inference on D^b, the offline discriminator calibrates its scores via calculating cosine similarities between the class prototypes and the box features. Adaptive thresholding then computes class-wise statistics from calibrated boxes to determine a proper threshold to filter low-quality mined instances.

confidence threshold δ is set to harvest boxes with high confidence in Y_i^n. The selected instances $H_i^n = \{y_{ij}^n | s_{ij} >= \delta\}$ can then be used as the offline mined novel instances.

To determine the offline mining threshold δ, a simple solution may use a fixed high threshold to filter out unreliable boxes, but it is not feasible for FSOD. The severe data scarcity and extreme class imbalance make the predicted novel scores of M^s exhibit a large variance and tend to be generally low, hence the fixed high confidence threshold fails to deal with different novel classes. On the other hand, the pervasive misclassification of the novel classifier of M^s results in massive false positives in H_i^n. Towards the aforementioned drawbacks, we introduce **co-mining with offline discriminator** to promote the discriminative ability of the classifier. An **adaptive thresholding** is also introduced to find a proper threshold for different novel classes.

Co-Mining with Offline Discriminator Given merely the novel class training samples of K-shots, it is challenging to acquire a discriminative classifier. Inspired by (Qiao et al., 2021; Kaul et al., 2022), we propose to leverage an offline discriminative model M^d, *e.g.*, MoCo v2 (Chen et al., 2020), to mine implicit novel instances with the initial FSOD model M^s collaboratively.

As shown in Fig. 3, given K novel samples $\{(x_k^{c_l^n}, b_k^{c_l^n})\}$ for the novel class $c_l^n \in C^n$ from the novel dataset D^n, we first forward the novel set image $x_k^{c_l^n}$ through the offline discriminative model M^d, then employ RoIAlign (He et al., 2017) to extract the area bounded by the ground-truth box $b_k^{c_l^n}$ as follows,

$$f_k^{c_l^n} = \text{RoIAlign}(M^d(x_k^{c_l^n}), b_k^{c_l^n}) , \quad \bar{p}^{c_l^n} = \frac{1}{K}\sum_{k=1}^{K} f_k^{c_l^n} , \tag{2}$$

where $f_k^{c_l^n}$ denotes the feature embedding of k-th novel sample, the class prototype $\bar{p}^{c_l^n}$ is the mean feature over all K instances for the class c_l^n. During inference in Equ. 1, the RPN first predicts a set of proposals P_i. We then compute the feature embedding f_{ij}^b of j-th proposal $p_{ij} \in P_i$ on i-th base set image $x_i^b \in D^b$ similar to Equ. 2, and cosine similarity score is computed with the class prototype $\bar{p}^{c_l^n}$ for each novel class c_l^n,

$$f_{ij}^b = \text{RoIAlign}(M^d(x_i^b), p_{ij}) , \quad cos_{ij}^{c_l^n} = \frac{\tau \cdot {f_{ij}^b}^T \cdot \bar{p}^{c_l^n}}{||f_{ij}^b|| \cdot ||\bar{p}^{c_l^n}||} , \tag{3}$$

where τ is a temperature factor. We concatenate all cosine similarities of N novel classes and apply the calibration as follows,

$$cos_{ij} = [cos_{ij}^{c_1^n}, \cdots, cos_{ij}^{c_N^n}] , \quad \hat{s}_{ij} \leftarrow \sqrt{cos_{ij} \cdot s_{ij}} , \quad \hat{Y}_i^n = \{(\hat{s}_{ij}, b_{ij}, l_{ij})\} , \tag{4}$$

where $[\cdots]$ denotes concatenation operation. It is noted we only apply the calibration to each novel class. To this end, the inference results will be updated to $\hat{Y}_i^n = \{(\hat{s}_{ij}, b_{ij}, l_{ij})\}$.

Adaptive Thresholding To filter out candidate instances of low quality in $\hat{Y}_i^n$, we propose an adaptive thresholding scheme to obtain a proper δ according to the class-wise score distributions.

As shown in Fig. 3, for each novel classes $c_l^n \in C^n$, we first extract its candidate instances set $\hat{Y}_i^{c_l^n} = \{\hat{s}_i^{c_l^n}, b_i^{c_l^n}\}$ from $\hat{Y}_i^n$. After then, the mean $\mu^{c_l^n}$ and deviation $\sigma^{c_l^n}$ will be computed based on classification scores $\{\hat{s}_i^{c_l^n}\}$. To this end, we can compute the final confidence threshold and harvest high-quality predict results as follows,

$$\delta^{c_l^n} = \mu^{c_l^n} + \alpha \cdot \sigma^{c_l^n} , \quad H_i^{c_l^n} = \{\hat{y}_i^{c_l^n} | \hat{s}_i^{c_l^n} >= \delta^{c_l^n}\} , \quad H_i^n = \{H_i^{c_l^n}\} , \tag{5}$$

where α is a coefficient that controls the magnitude of deviation offset to decide the number of kept instances. It is noted we further clamp the maximum number to be N_{max}. Intuitively, the score mean $\mu^{c_l^n}$ is a measure of the transferring hardness of novel class c_l^n, and $\sigma^{c_l^n}$ indicates the compactness of intra-class score distribution. The $\delta^{c_l^n}$ leverages both the $\mu^{c_l^n}$ and $\sigma^{c_l^n}$ to adaptively distinguish the reliable implicit novel instances without introducing computational cost.

3.3 Online Mining Mechanism and Adaptive Mingling Design

With the offline mined implicit novel instances H^n, we are ready to re-train a new detector with satisfactory performance on novel classes. However, these instances are sourced from a static offline teacher M^s with limited precision, and cannot be updated as the model improves, which hinders further performance improvements. Hence we introduce an online mining mechanism to update H^n on the fly. Specifically, we adopt a teacher-student learning paradigm as shown in Fig. 2. The teacher shares the same network architecture with the student model. During training, the teacher mines implicit novel instances I^n at each training iteration, and its parameters are updated by the student via EMA. The slowly updated teacher can be considered a temporal model ensemble of the student at different iterations, hence detecting implicit novel instances more accurately.

Nevertheless, the teacher is initialized from M^s of limited precision on novel classes, the online mined novel instances I^n can be very noisy, especially at the beginning of the training. Thus, we devise an adaptive mingling scheme to combine the offline and online mined novel instances, where the offline mined novel instances H^n warm up the early training and the online mined novel instances I^n gradually substitute the offline mined instances to further improve the performance. We also introduce an IoU branching mechanism to improve the quality of online mined novel instances.

Adaptive Mingling Design During training, given a base set image x_i^b, the teacher first online mines novel instances I_i^n with a similar procedure with Equ. 1, and we mingle the online mined novel instances I_i^n with the offline mined novel instances H_i^n as follows,

$$Y_i^n = \phi^{RCNN}(x_i^b, [P_i, H_i^n]) , \quad I_i^n = \{y_{ij}^n | s_{ij} >= \delta\} , \quad \hat{I}_i = NMS([I_i^n, H_i^n]) . \tag{6}$$

We concatenate H_i^n with P_i and I_i^n before and after the R-CNN head, respectively. Here P_i is RPN proposals predicted by the teacher model. We argue these two concatenations play an important role from two aspects. 1) At the beginning of the training, due to the poor-learned RPN and R-CNN, the high confidence threshold δ can filter out almost all of the novel instances, so that I_i^n degrades to an empty set, hence only H_i^n is remained to provide training signal to warm up the beginning training of the student. 2) As the training process proceeds, the online teacher becomes more and more discriminative. By presenting H_i^n as extra proposals, the R-CNN head of the teacher will calibrate some misclassification and also discover missed instances in H_i^n. The mingled novel instances I_i^n of novel classes C^n will then be combined with annotations B_i^b of base classes C^b as ground-truths during the training of the student model.

IoU Branching Correction To further improve the quality of online mined novel instances, we notice the model trained under low data regime cannot well recognize precisely-localized boxes, hence we introduce IoU Branching mechanism to better mine high-quality novel instances. Specifically, we introduce an extra IoU branch that parallels to the original R-CNN head to learn to predict the IoU between predicted boxes and ground truths. The structure is the same as the original R-CNN branch, *i.e.*, two fully-connected (FC) layers and followed by an IoU predictor (a single FC layer). During mining, we combine the classification scores with IoU scores in Equ. 1 as follows,

$$s'_{ij} = \sqrt{s_{ij} \cdot iou_{ij}} , \quad I_i^n = \{y_{ij} | s'_{ij} >= \delta\} , \tag{7}$$

where iou_{ij} denotes the predicted IoU score of j-th proposal on i-th image. A standard MSE loss is adopted to optimize the IoU branch. All modules of the R-CNN head are jointly optimized by the

following loss in an end-to-end manner:

$$\mathcal{L}_{RCNN} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \beta \cdot \mathcal{L}_{IoU} \ , \tag{8}$$

where β denotes the loss weight of the loss of the IoU branch.

Finetuning on Few Shot Novel Samples After training FSOD model on mined implicit novel instances with optimizing all parameters, our MINI has achieved strong performance on novel classes. Due to the mined novel instances inevitably have some noise, we can further boost the performance of MINI via finetuning on few shot novel ground-truth. Specifically, we finetune the box predictor of MINI on a small balanced training set of both base and novel ground-truth, finally achieving new state-of-the-art performance on FSOD tasks.

4 Experiments

In this section, we first outline the datasets and benchmark protocols in Sec 4.1, the implementation details in Sec 4.2. Then, we compare our approach with the latest methods of FSOD in Sec 4.3. Finally, we make an extensive ablation study about different components in Sec 4.4.

4.1 Datasets and Evaluation Protocols

We follow the same data split and evaluation protocols used in (Wang et al., 2020) for fair comparisons. All experiments are evaluated on both PASCAL VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) datasets. Please refer to Appendix G for the detailed protocols.

4.2 Implementation Details

We implement our method based on MMDetection (Chen et al., 2019) and MMFewShot (mmfewshot Contributors, 2021). We employ the Faster R-CNN (Ren et al., 2015) with Feature Pyramid Network (Lin et al., 2017a) and ResNet-101 (He et al., 2016) as base model. Please refer to Appendix F for the detailed settings.

4.3 Main Results

In Table 1, we compare *MINI* with latest FSOD methods on the PASCAL VOC benchmark. In all splits and shots, *MINI* achieves new SOTA performance and outperforms the second-best by a large margin. Specifically, *MINI* boosts the current SOTA by 8.6, 7.9, 4.7, 5.3, 8.1 and 9.7, 7.1, 8.9, 6.8, 9.9 and 8.9, 8.2, 5.0, 4.7, 6.7 for shot 1, 2, 3, 5, 10 on novel split 1, 2 and 3, respectively. The significant performance improvements are consistent across different shots and splits. Compared with the offline mining counterpart (Kaul et al., 2022) (third-to-last line of Table 1), *MINI* boosts the performance by 17.5, 21.0, 13.6, 11.5, 10.5 and 19.0, 24.4, 11.6, 12.3, 13.1 and 16.6, 13.8, 9.0, 7.3 and 10.8 for shot 1, 2, 3, 5, 10 on novel split 1, 2 and 3, respectively. Similar performance gains can be observed on the MS COCO benchmark. As shown in Table 2, *MINI* outperforms all FSOD methods by a large margin with the COCO-style AP metric. Concretely, our method achieves 21.8 and 27.3 and boosts the SOTA performance by 2.4 and 2.8 for $K = 10$ and 30, respectively. The superior performance on both datasets suggests *MINI* can generalize well under different datasets. Besides, we also explore the cross dataset setting that we mine extra novel instances from some external unlabeled datasets, which further boost the performance, *e.g.*, boost from 21.8, 27.3 to 23.6, 29.3 for COCO shot 10 and 30, respectively. Please refer to Appendix B for details.

4.4 Ablation Study

In this section, we conduct thorough ablation studies on each component of our approach. Unless otherwise specified, all experiments are conducted on novel split 1 of PASCAL VOC benchmark.

Component Analysis Table 3 shows the effectiveness of each component. We start with the pure offline mining baseline (first row on the left) with a fixed confidence threshold 0.7, which improves the TFA baseline (third row in Table 1) on high-shots but degrades the performance on low-shots.

Table 1: Performance (novel AP50) across three splits on PASCAL VOC dataset. **Red**/**Blue** denote best and second-best results, respectively

Method/Shot		Novel Split 1					Novel Split 2					Novel Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW (Kang et al., 2019)	*ICCV 19*	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
Meta R-CNN (Yan et al., 2019)	*ICCV 19*	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA w/ cos (Wang et al., 2020)	*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
MPSR (Wu et al., 2020)	*ECCV 20*	41.7	-	51.4	55.2	61.8	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7
FSCE (Sun et al., 2021)	*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
SRR-FSD (Zhu et al., 2021)	*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 (Li et al., 2021a)	*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
FADI (Cao et al., 2021)	*NeurIPS 21*	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	55.0	59.6
DeFRCN (Qiao et al., 2021)	*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
Label,Verify (Kaul et al., 2022)	*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
Multi-Faceted (Wu et al., 2022)	*ECCV 22*	**63.4**	**66.3**	**67.7**	**69.4**	**68.1**	**42.1**	**46.5**	**53.4**	**55.3**	**53.8**	**56.1**	**58.3**	**59.0**	**62.2**	**63.7**
MINI (Ours)		**72.0**	**74.2**	**72.4**	**74.7**	**76.2**	**51.8**	**53.6**	**62.3**	**62.1**	**63.7**	**65.0**	**66.5**	**64.0**	**66.9**	**70.4**

Table 2: Performance on MS COCO dataset. **Red**/**Blue** denote best and second-best results, respectively

Method	nAP		nAP50		nAP75		Method	nAP		nAP50		nAP75	
	10	30	10	30	10	30		10	30	10	30	10	30
Meta R-CNN (Yan et al., 2019)	8.7	12.4	19.1	25.3	6.6	10.8	FADI (Cao et al., 2021)	12.2	16.1	22.7	29.1	11.9	15.8
TFA w/ cos (Wang et al., 2020)	10.0	13.7	19.1	24.9	9.3	13.4	DeFRCN (Qiao et al., 2021)	18.5	22.6	-	-	-	-
MPSR (Wu et al., 2020)	9.8	14.1	17.9	25.4	9.7	14.2	Label (Kaul et al., 2022)	17.8	**24.5**	30.9	41.1	17.8	25.0
SRR-FSD (Zhu et al., 2021)	11.3	14.7	23.0	29.2	9.8	13.5	Multi-Faceted (Wu et al., 2022)	**19.4**	22.7	-	-	20.2	23.2
CME (Li et al., 2021a)	15.1	16.9	24.6	28.0	16.4	17.8	**MINI (Ours)**	**21.8**	**27.3**	**38.0**	**44.9**	**21.5**	**28.5**

Table 3: Effectiveness of each component. ODC, AT, TS, AM, IB, FT denotes offline discriminator co-mining, adaptive threshold, teacher-student framework, adaptive mingling, iou branching and fine-tuning, respectively

Offline		nAP50					Online			FT	nAP50				
ODC	AT	1	2	3	5	10	TS	AM	IB		1	2	3	5	10
✗	✗	19.3	48.5	57.1	65.2	66.9	✓	✗	✗	✗	0.0	1.8	16.7	46.7	44.9
✓	✗	34.8	54.7	60.2	65.8	66.8	✓	✓	✗	✗	68.3	71.0	70.7	71.3	72.8
✗	✓	58.1	63.3	62.5	67.7	67.5	✓	✓	✓	✗	69.9	72.5	71.7	72.7	73.8
✓	✓	63.5	67.7	66.8	70.3	68.8	✓	✓	✓	✓	72.0	74.2	72.4	74.7	76.2

Our adaptive thresholding (AT) rescues the performance degradation and improves the performance on all shots, suggesting it is vital to set a proper threshold. Offline discriminator co-mining (ODC) results in decent gains in lower shots but lower gain in higher shots, *e.g.*, +5.4 and +1.3 for $K = 1$ and 10, which suggests the offline discriminator is a good enhancement to TFA in low-shots but share a similar discriminative ability on high-shots. We then turn to the online mining part. The pure online mining with a teacher-student (TS) framework always fails to converge due to the noisy training signals. Our adaptive mingling (AM) unifies the offline and online mining adaptively, stabilizing the training process. Based on adaptive mingling, the iou branching (IB) further improves the quality of mined instances. Finally, we fine-tune (FT) the re-trained model on the balanced set to mitigate the side effect of the inaccurate supervision from mined implicit instances.

Hyper-parameters Ablation Please refer to Appendix H for detailed hyper-parameters ablation.

Flexibility of Adaptive Thresholding To understand how adaptive thresholding works, we study how threshold δ varies among different shots K and classes in Fig. 4. We can see δ well characterize the transfer hardness among different classes and shots. On the one hand, as shot grows, the classification scores should be higher since the classifier learns better. Adaptive thresholding decides to steadily increase δ to rigorous the mined novel instances to suppress false positives. On the other hand, the classifier tends to predict higher scores for those novel classes that are similar to base classes (Cao et al., 2021), *e.g.*, "bus" is an easy class since it is similar to "car", but "bird" is a hard class since no base class is similar to it. Therefore, adaptive thresholding decides a higher δ for "bus" and a lower δ for "bird". Such flexibility leads to the strong robustness of our adaptive thresholding to fit in different scenarios.

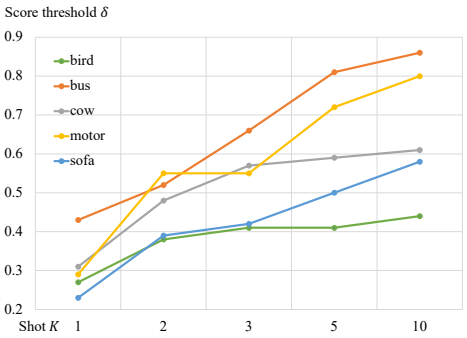

Figure 4: Adaptive threshold δ of different novel classes varies on PASCAL VOC Novel Split 1

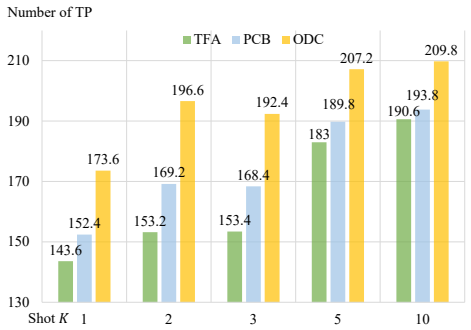

Figure 5: Comparison of the number of true positive (TP) of mined novel instances when using different mining methods

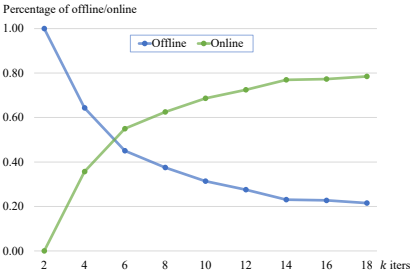

Figure 6: Percentage of offline and online mined instances kept for training at different iterations

Table 4: Ablation study for whether enhancing RPN or R-CNN of the online teacher with offline mined novel instances

RPN	R-CNN	nAP50 1	2	3	5	10
✗	✗	0.0	0.0	0.0	0.0	0.0
✓	✗	0.0	0.0	0.0	0.0	0.0
✗	✓	69.5	72.1	70.9	71.5	72.9
✓	✓	**69.9**	**72.5**	**71.7**	**72.7**	**73.8**

Explore Offline Discriminator Co-Mining In this section, we explore how many novel instances can be found by different mining methods. We compare TFA (Wang et al., 2020), PCB (Qiao et al., 2021) and our Offline Discriminator Co-mining (ODC), respectively, the result is shown in Fig 5. We only count true positives among mined novel instances that overlap a GT bounding box with IoU $>= 0.5$. Although both PCB and ODC can increase the number of TPs, the increment of ODC is more significant, especially in low-shot scenarios, *e.g.*, +9.2, +16.0 and +30.0, +43.4 for PCB and ODC in shots 1 and 2, respectively. This suggests the effectiveness of ODC. Please refer to Appendix E for more detailed comparison between PCB and ODC.

Complementivity between Offline and Online Mining To understand how adaptive mingling balances online and offline mined instances, we record the number of these two types of instances kept after the NMS in Equ. 6 at different iterations in Fig. 6. At the beginning of the training, the online teacher mines no instances and offline instances are mainly kept for training. This well explains the first row of Tab. 4. In the second row, although introducing the offline mined novel instances to the RPN, the performance is still bad because the pseudo labels are from the R-CNN Head of the noisy online teacher. As the training process proceeds, online instances gradually dominate kept instances, which demonstrates a better online teacher can discover more diverse novel instances than the initial FSOD model. The last row of Tab. 4 shows enhancing the RPN can bring further gains.

5 Conclusion

In this paper, we propose *Mining Implicit Novel Instances (MINI)* to better tackle FSOD. MINI unifies an offline mining mechanism and an online mining mechanism. The offline mining mechanism leverages a self-supervised discriminator to collaboratively mine implicit novel instances with a trained FSOD model. Taking the mined novel instances as auxiliary training samples, the online mining mechanism takes a teacher-student framework to simultaneously update the FSOD model and the mined implicit novel instances on the fly. *MINI* achieves new SOTA performance on various benchmarks, which demonstrates its effectiveness.

REFERENCES

Yuhang Cao, Jiaqi Wang, Ying Jin, Tong Wu, Kai Chen, Ziwei Liu, and Dahua Lin. Few-shot object detection via association and discrimination. In *Advances in Neural Information Processing Systems*, 2021.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

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

Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, 2017.

Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

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

Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *IEEE International Conference on Computer Vision*, 2019.

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 *IEEE International Conference on Computer Vision*, 2019.

Prannay Kaul, Weidi Xie, and Andrew Zisserman. Label, verify, correct: A simple few shot object detection method. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2021a.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2021b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017b.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, 2016.

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 *International Conference on Learning Representations*, 2021.

mmfewshot Contributors. Openmmlab few shot learning toolbox and benchmark. https://github.com/open-mmlab/mmfewshot, 2021.

Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *IEEE International Conference on Computer Vision*, 2021.

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*, 2015.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *IEEE International Conference on Computer Vision*, 2019.

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

Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Peng Tang, Chetan Ramaiah, Yan Wang, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. In *IEEE Winter Conference on Applications of Computer Vision*, 2021a.

Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021b.

Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, 2020.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

Shuang Wu, Wenjie Pei, Dianwen Mei, Fanglin Chen, Jiandong Tian, and Guangming Lu. Multi-faceted distillation of base-novel commonality for few-shot object detection. *European Conference on Computer Vision*, 2022.

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

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. In *IEEE International Conference on Computer Vision*, 2021.

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 *IEEE International Conference on Computer Vision*, 2019.

Lu Zhang, Shuigeng Zhou, Jihong Guan, and Ji Zhang. Accurate few-shot object detection with support-query mutual guidance and hybrid loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Chenchen Zhu, Fangyi Chen, Uzair Ahmed, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Table 5: Performance comparison between TFA and *MINI*. We use the same base and novel set as PASCAL VOC novel split1 with the shot1 setting, but we exclude images that contain a selected novel class c^n from the base dataset to simulate there is no co-occurred c^n, and the excluded class is marked as red. For example, the "bird" row indicates we exclude all the images containing "bird" instances from the base dataset

(a) TFA

Exclude	nAP	Bird	Bus	Cow	Motor	Sofa
Bird	37.2	27.1	63.8	33.0	45.0	17.2
Bus	37.9	24.0	39.4	39.0	59.7	27.5
Cow	38.5	22.6	54.9	43.0	54.9	16.9
Motor	36.6	18.4	57.7	41.2	42.7	23.0
Sofa	41.8	27.3	70.0	37.1	49.3	25.1

(b) MINI

Exclude	nAP	Bird	Bus	Cow	Motor	Sofa
Bird	65.6	33.0	84.5	70.7	77.9	61.6
Bus	64.1	56.9	49.1	77.1	76.5	61.1
Cow	65.4	57.8	84.1	46.1	77.0	62.2
Motor	67.5	60.6	83.9	73.9	52.3	66.8
Sofa	65.7	62.4	83.5	75.8	74.9	32.0

Bus Motorbike Cow

Figure 7: Examples of mined similar instances from excluded class dataset, *e.g.*, "wheel" is mined for novel class "bus", "bicycle" is mined for novel class "motorbike", "horse" is mined for novel class "cow".

APPENDIX

APPENDIX A: ROBUSTNESS OF MINI

Although co-occurrence widely exists in benchmark datasets, there may be a case that the novel class does not co-occur with base classes. In this section, we test the robustness of *MINI* in such a case. Specifically, we manually remove images that contain a selected novel class from the original base dataset of PASCAL VOC Novel Split1 with the Shot1 setting, and keep the novel dataset unchanged. We then train a TFA and *MINI* on this processed dataset, the results are shown in Table 5. Surprisingly, even the base dataset does not contain the removed novel class, our *MINI* can still significantly improve the performance for the excluded class, *e.g.*, boost nAP50 by 5.9 (from 27.1 to 33.0) for "Bird" and "9.7" (39.4 to 49.1) for bus. So what instances are mined by *MINI* for these excluded novel classes? We draw some examples in Fig. 7. We can see these mined novel instances share a strong texture or shape similarity with the exclude class, *e.g.*, the wheel of the base class "aeroplane" is also a part for novel class "bus", the shape of the base class "bicycle" is similar to the novel class "motorbike", the texture of the base class "horse" is similar to the novel class "cow". We conjecture learning from these similar instances of base classes can also promote the feature representation ability of the corresponding novel classes.

APPENDIX B: GENERALIZING TO EXTERNAL DATASETS

Currently, we only mine implicit novel instances from the base dataset, can we generalize *MINI* to external unlabeled datasets in a cross-domain manner? In this section, we explore two such settings, which adopt MS COCO (Lin et al., 2014) and Object365 (Shao et al., 2019) as external datasets for the original base set PASCAL VOC (Everingham et al., 2010) and MS COCO, respectively. The results are shown in Tab. 6. We adopt same hyper-parameters and mine 100 and 2000 instances on extra datasets for each novel class in Tab. 6a and Tab. 6b, respectively. Mining only from the base or extra set can both significantly improve the performance, but the performance of the extra set is inferior to the base set due to the domain gap between datasets. Moreover, mining from both sets can

Table 6: Generalizing *MINI* to mine novel instances from other unlabeled datasets

(a) PASCAL VOC

Base Set	Extra Set	nAP50				
VOC	COCO	1	2	3	5	10
✗	✗	41.9	49.1	49.9	58.0	58.4
✓	✗	72.0	74.2	72.4	74.7	76.2
✗	✓	62.9	69.3	68.2	72.9	72.4
✓	✓	**73.7**	**75.9**	**76.5**	**78.1**	**77.1**

(b) MS COCO

Base Set	Extra Set	nAP	
COCO	Object365	10	30
✗	✗	10.4	14.7
✓	✗	21.8	27.3
✗	✓	21.2	26.4
✓	✓	**23.6**	**29.3**

Table 7: Performance comparison with pure offline and online mining on PASCAL VOC dataset

Method	nAP50					Method	nAP50				
	1	2	3	5	10		1	2	3	5	10
Offline, $\delta = 0.5$	38.8	59.2	60.2	64.8	66.4	Online, $\delta = 0.5$	3.9	13.2	15.1	11.7	6.2
Offline, $\delta = 0.7$	19.3	48.5	57.1	65.2	66.9	Online, $\delta = 0.7$	1.8	3.6	22.3	52.0	60.6
Offline, $\delta = 0.9$	0.0	7.1	16.6	22.6	57.5	Online, $\delta = 0.9$	0.0	3.6	8.9	25.4	47.6
Offline, adaptive δ	**58.1**	**63.3**	**62.5**	**67.7**	**67.5**	Ours	**72.0**	**74.2**	**72.4**	**74.7**	**76.2**

further bring considerable gains, which demonstrates *MINI* can well generalize to external datasets and discover valuable instances to enhance the original model.

Appendix C: Comparison with Pure Offline and Online Mining

In this section, we compare *MINI* with pure offline and online mining mechanisms. We employ STAC (Sohn et al., 2020) and Unbiased Teacher (UB-T) (Liu et al., 2021) as representative methods for offline and online, respectively. As shown in Table 7, the performance of both pure offline and online mining are far behind *MINI*. We adopt the same hyper-parameters setting with the official paper except for the confidence threshold δ. The original STAC adopts $\delta = 0.9$. We notice such a high threshold can filter all novel instances, decreasing δ from 0.9 to 0.5 can significantly boost performance in lower shots, *e.g.*, 0.0, 7.1 to 38.8, 59.2 for shots 1 and 2, respectively. But it also degrades the performance in higher shots, *e.g.*, nAP50 drops 0.4 and 0.5 when decreasing δ from 0.7 to 0.5 for shots 5 and 10, respectively, since it will result in more false positives. Replacing the fixed threshold with our adaptive threshold improves the performance on all shots, especially in low-shot scenarios. For online mining, we initialize both teacher and student with TFA (Wang et al., 2020) in the burn in stage (Liu et al., 2021). We also adopt Focal Loss (Lin et al., 2017b) as in (Liu et al., 2021), we notice it is insufficient to resolve the severe data scarcity and extreme class imbalance in FSOD. By adaptive mingling the offline and online mined novel instances, the proposed *MINI* significantly outperforms these two counterparts, demonstrating the superiority of our method.

Appendix D: Comparison between different offline discriminators

The offline mining mechanism leverages an offline discriminative model to collaboratively mine implicit novel instances with the trained FSOD network, in this section we conduct a comparison between different offline discriminators. As shown in Table 8, overall MoCo v2 is better than ImageNet pre-train, especially in low-shot, *e.g.*, $K = 1, 2, 3$; but slightly inferior in higher shots, *e.g.*, $K = 10$. We adopt MoCo v2 as our offline discriminator. More powerful offline discriminators may further improve the performance, we leave it as our future works.

Appendix E: Comparison between Offline discriminator Co-Mining (ODC) and PCB

In this section, we first revisit the inference procedure of a conventional two-stage detector, then we compare the technique difference between Offline Discriminator Co-Mining (ODC) and PCB.

Table 8: Performance comparison between different offline discriminators in offline mining. For ImageNet Pre-train, we adopt a supervised-trained ResNet-50. For MoCo v2 Chen et al. (2020), we adopt a self-supervised trained ResNet-50.

Discriminative Model	nAP50				
	1	2	3	5	10
ImageNet Pre-train	62.3	67.2	66.4	**70.3**	**69.4**
MoCo v2	**63.5**	**67.7**	**66.8**	**70.3**	68.8

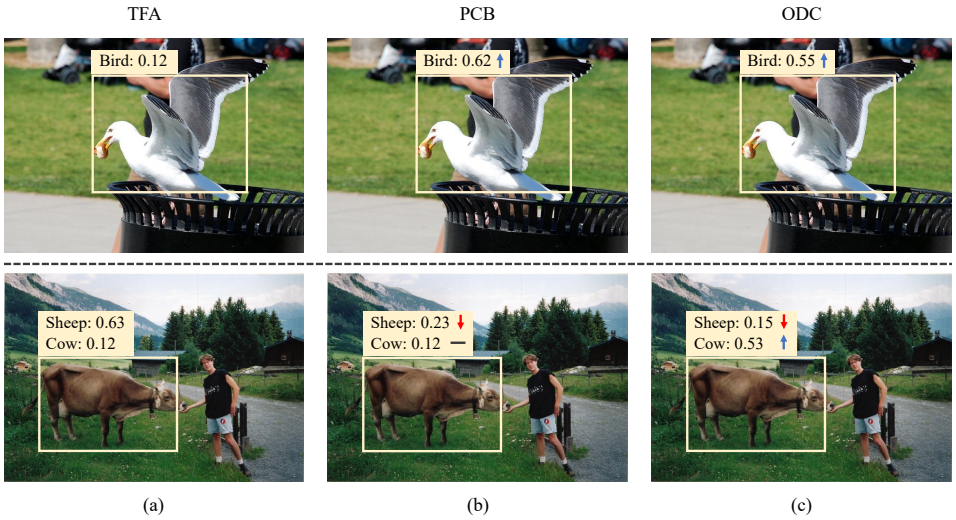

Figure 8: Demonstration of how PCB (Qiao et al., 2021) and ODC (Offline Discriminator Co-Mining) calibrate scores of the TFA (Wang et al., 2020) baseline. In the first row, TFA predicts the correct label "bird" for the instance, but with low scores. Both PCB and ODC boost the corresponding score, and remain the original correct label. In the second row, TFA predicts an incorrect label, *i.e.*, predict "sheep" for the instance, the true label should be "cow". PCB can only suppress the predicted wrong scores, but can not correct the wrong label. ODC correct the wrong label "sheep" into "cow".

Inference Procedure For a conventional two-stage detector, *e.g.*, Faster R-CNN (Ren et al., 2015), the inference procedure can be formularized as follows. Given an input image x, the Region Proposal Network (RPN) ϕ^{RPN} first predicts a set of proposals P, the number of proposals is denoted as N^P,

$$P = \phi^{RPN}(x), \quad |P| = N^P, \tag{9}$$

then R-CNN Head ϕ^{RCNN} classifies and regresses each proposal to yield the predicted results $\hat{Y}$,

$$Y = \phi^{RCNN}(x, P), \quad Y = \{y_i = (s_i, b_i) | s_i \in S, b_i \in B\}, \quad S \in R^{N^p \times C}, B^{N^P \times C \times 4}, \tag{10}$$

where C denotes the number of classes, and the number of predicted scores S and boxes B equals the number of proposals N^P. Then some post-processing procedures, *e.g.*, NMS, are applied to yield the final detection results $\hat{Y}$,

$$\begin{aligned} \hat{Y} = NMS(S, B), \quad \hat{Y} = \{\hat{y_i} = (\hat{s_i}, \hat{b_i}, \hat{l_i}) | \hat{s_i} \in \hat{S}, \hat{b_i} \in \hat{B}, \hat{l_i} \in \hat{L}\}, \\ \hat{S} \in R^{N^{keep}}, \hat{B} \in R^{N^{keep} \times 4}, \hat{L} \in R^{N^{keep}}, \end{aligned} \tag{11}$$

where $\hat{S}, \hat{B}, \hat{L}$ denotes the kept predicted scores, bounding boxes and labels. The number of predicted instances equals N^{keep}, which denotes the max predictions allowed per image, *e.g.*, for MS COCO Lin et al. (2014) $N^{keep} = 100$. We can see the R-CNN Head determines the label in Equ. 11.

Calibration of PCB For PCB, it calibrates the scores *after the R-CNN Head determines the labels*.

$$\begin{aligned} \hat{Y} = NMS(S, B), \quad \hat{Y} = \{\hat{y_i} = (\hat{s_i}, \hat{b_i}, \hat{l_i}) | \hat{s_i} \in \hat{S}, \hat{b_i} \in \hat{B}, \hat{l_i} \in \hat{L}\}, \\ \hat{S'} = PCB(x, \hat{Y}), \quad \hat{Y'} = \{\hat{y_i}' = (\hat{s_i}', \hat{b_i}, \hat{l_i}) | \hat{s_i}' \in \hat{S'}, \hat{b_i} \in \hat{B}, \hat{l_i} \in \hat{L}\}, \end{aligned} \tag{12}$$

Table 9: Performance comparison between PCB (Qiao et al., 2021) and ODC (Offline Discriminator Co-Mining) on PASCAL VOC Split1. Online Mining is not involved here for better comparison

Method	nAP50				
	1	2	3	5	10
PCB	60.4	64.8	64.1	68.1	68.2
ODC	**63.5**	**67.7**	**66.8**	**70.3**	**68.8**

here $\hat{S}'$ denotes the calibrated scores by PCB. For a predicted instances $\hat{y}_i$, if the predicted label $\hat{l}_i$ is correct (top of Fig 8(b)), PCB can boost the corresponding scores. However, if the label $\hat{l}_i$ is incorrect (bottom of Fig 8(b)), PCB can only suppress the corresponding scores, but can not change the label $\hat{l}_i$.

Calibration of ODC For ODC, it calibrates scores *before the R-CNN Head determines the labels*, it modifies scores of all novel classes of each proposal as follows,

$$\begin{aligned} &Y = \phi^{RCNN}(x, P), \quad Y = \{y_i = (s_i, b_i) | s_i \in S, b_i \in B\} \\ &S' = ODC(x, P, S), \quad \hat{Y}' = NMS(S', B), \quad \hat{Y}' = \{\hat{y_i}' = (\hat{s_i}', \hat{b_i}, \hat{l_i}')\}, \end{aligned} \tag{13}$$

here S' denotes the calibrated scores of each proposal by ODC. In this way, for each predicted score s_i of a proposal, our ODC can boost the score of the correct label but suppress others, hence helping the R-CNN heads correct the predicted labels. As shown in Fig 8(c), when TFA predicts a correct label (top), ODC boosts the corresponding scores. Moreover, when TFA predicts an incorrect label (bottom), ODC boosts the score of the correct label and suppress incorrect ones to help the R-CNN Head determines the label.

Performance Comparison Table 9 compares the performance between PCB and ODC when applied to the offline mining. For a better comparison, we don't involve online mining here. We can see that ODC outperforms PCB on all shots, which demonstrates the superiority of our ODC over PCB in the context of instances mining.

Appendix F: Implementation Details

We implement our method based on MMDetection (Chen et al., 2019) and MMFewShot (mmfewshot Contributors, 2021). We employ the Faster R-CNN (Ren et al., 2015) with Feature Pyramid Network (Lin et al., 2017a) and ResNet-101 (He et al., 2016) as base model. All models are trained on 8 Titan-XP GPUs with batch-size 16 (2 images per GPU), and optimized by a standard SGD optimizer with learning rate 0.02, momentum 0.9 and weight decay $10e^{-4}$. We strictly follow the protocol introduced by TFA (Wang et al., 2020) without any modifications to initialize the M^s. MoCo v2 (Chen et al., 2020) w/ ResNet-50 (He et al., 2016) is employed to co-mine novel instances with FSOD model, and we take the C4 feature, *i.e.*, the feature of the last layer of ResNet to compute the cosine similarity. α in Equ. 5 is set to be 1.5 for all experiments, and we limit the maximum of novel instances to be 300 and 3000 for PASCAL VOC and MS COCO, respectively. For the online learning stage, we follow unbiased teacher (Liu et al., 2016) to apply weak and strong augmentations to the teacher and student model, respectively. For PASCAL VOC all models are trained for 18k iterations and decayed at 12k and 16k, respectively, and the confidence threshold δ is set to be 0.7. For MS COCO all models are trained for 160k iterations and decayed at 110k and 145k, respectively, and the confidence threshold δ is set to be 0.8. In the last fine-tuning stage, for PASCAL VOC, we only fine-tune the box classifier, predictor and IoU predictor for 4k, 8k, 8k, 8k, 12k iterations for $K = 1, 2, 3, 5, 10$, respectively. For MS COCO, we fine-tune the whole R-CNN head for 4k, 8k iterations for $K = 10, 30$, respectively. The learning rate is set as 0.001 for both datasets.

Appendix G: Datasets and Evaluation Protocols

We follow the same data split construction and evaluation protocols used in (Wang et al., 2020) for fair comparisons. All experiments are evaluated on both PASCAL VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) datasets.

Table 10: Ablation study for hyper-parameters of different components. Varying α and N for adaptive thresholding in offline mining. Varying δ for online mining. Varying β for IoU branching

	nAP50						nAP50				
α	1	2	3	5	10	δ	1	2	3	5	10
0.0	**63.6**	**68.4**	66.7	69.6	67.7	0.5	19.0	23.7	13.4	14.2	18.4
1.5	63.5	67.7	**66.8**	**70.3**	**68.8**	0.7	**69.9**	**72.5**	**71.7**	**72.7**	**73.8**
3.0	59.8	59.4	65.9	69.6	67.2	0.9	65.2	70.5	69.8	71.2	71.4
N	1	2	3	5	10	β	1	2	3	5	10
150	62.4	65.9	66.0	68.9	67.7	0.5	**69.9**	72.5	**71.7**	**72.7**	**73.8**
300	63.5	67.7	**66.8**	**70.3**	**68.8**	1.0	69.4	**72.7**	69.8	34.1	31.6
450	**63.9**	**68.3**	66.3	68.8	67.3	2.0	69.1	72.0	30.4	33.0	72.3

PASCAL VOC has 20 classes, which are randomly split into 15 base classes and 5 novel classes. There are 3 different class splits, and we refer them as Novel Split 1, 2 and 3, respectively. For each split, there exists exhaustive base instances but only $K = 1, 2, 3, 5, 10$ annotated instances for novel classes. All instances are sampled from the union of VOC07 and VOC12 train/val set for training, and the model is tested on VOC07 test set. The standard PASCAL VOC metric, *i.e.*, Average Precision (IoU=0.5) for novel classes (nAP50) is reported.

MS COCO has 80 classes, 20 classes that overlap with PASCAL VOC are regarded as novel classes, the remaining 60 classes are considered as base classes. We evaluate our method for $K = 10, 30$ shots. And the standard COCO-style metric is adopted, which averages mAP of IoUs from 0.5 to 0.95 with an interval of 0.05. We also report nAP50 and nAP75, respectively.

APPENDIX H: HYPER-PARAMETERS ABLATION

In *MINI*, there are 4 hyper-parameters introduced, α and N for adaptive thresholding, δ for online mining, and β for IoU branching. Table 10 analyzes the effect of different choices of hyper-parameters. When studying α and N, we do not involve the online mining mechanism and the fine-tuning; when studying δ and β, we do not involve the fine-tuning. A smaller α and larger N will lead to more kept mined novel instances, which is beneficial in lower shots, *e.g.*, 1- and 2- shot, but can be harmful in higher shots since it may result in more false positives. We can observe the performance is not very sensitive to α and N, and we finally adopt $\alpha = 1.5$ and $N = 300$ for the offline mining. During online mining, it is necessary to set a relatively high δ. Because a too-small δ, *e.g.*, $\delta = 0.5$ can severely degrade the performance, as it will induce too many false positives to distract the learning of the student model. And we found a large β will disturb the training process, especially in higher shots. Through a coarse study, we adopt $\delta = 0.7$ and $\beta = 0.5$ for all experiments.