Tunable Hybrid Proposal Networks for the Open World

Matthew Inkawhich[1] Nathan Inkawhich[2] Hai Li[1] Yiran Chen[1]
[1]Duke University [2]Air Force Research Laboratory
matthew.inkawhich@duke.edu

Abstract

Current state-of-the-art object proposal networks are trained with a closed-world assumption, meaning they learn to only detect objects of the training classes. These models fail to provide high recall in open-world environments where important novel objects may be encountered. While a handful of recent works attempt to tackle this problem, they fail to consider that the optimal behavior of a proposal network can vary significantly depending on the data and application. Our goal is to provide a flexible proposal solution that can be easily tuned to suit a variety of open-world settings. To this end, we design a Tunable Hybrid Proposal Network (THPN) that leverages an adjustable hybrid architecture, a novel self-training procedure, and dynamic loss components to optimize the tradeoff between known and unknown object detection performance. To thoroughly evaluate our method, we devise several new challenges which invoke varying degrees of label bias by altering known class diversity and label count. We find that in every task, THPN easily outperforms existing baselines (e.g., RPN, OLN). Our method is also highly data efficient, surpassing baseline recall with a fraction of the labeled data.

1. Introduction

The goal of object proposal generation is to detect and localize all potential objects of interest in an image. High-quality object proposals serve as the foundation for many vision-based applications including object detection [6, 16, 17, 36, 51], segmentation [3, 9, 21], object discovery [4, 8, 53], and visual tracking [37, 63]. Over recent years, heuristic-based object proposal algorithms [50, 60, 75] have been supplanted by deep learning-based solutions such as Region Proposal Network (RPN) [51] which provide superior recall and faster inference. Currently, there is a significant push towards creating models that can function in open-set [10, 55, 56] and open-world [5, 29] environments. Here, the deployed model will encounter known object classes from the labeled training distribution as well as novel classes. We refer to these instances as "in-distribution" (ID) and "out-of-distribution" (OOD) objects, respectively. An ideal object proposal model would detect all ID *and* OOD objects of interest with high confidence. However, most existing proposal networks overfit to the ID classes, meaning that if we deploy them in an open-world setting many OOD objects will go undetected [10]. In a real-world system this kind of mistake could have catastrophic consequences. While several recent works improve a classifier's ability to discern ID from OOD objects [11, 23, 24, 27, 44, 71], we argue that the *proposal network* is holding back open-set/world detection. Ultimately, if an OOD object is not confidently proposed, the region will never even reach the classification stage.

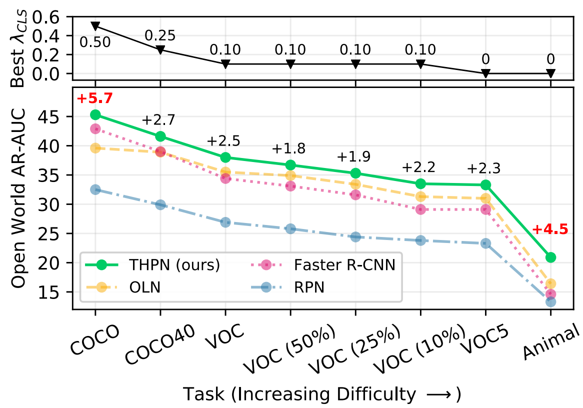


Figure 1. THPN's adjustability allows it to excel across a broad range of open world tasks. *Top:* optimal λ_{CLS} for each task. *Bottom:* recall comparison; annotations are THPN's margins over OLN [32].

The primary motivation for this work is to design a proposal network that is useful in a variety of real-world settings. To do this, we must expand the settings that we use to evaluate the models. Currently, the litmus test for open-set/world object detection performance involves training on one diverse natural imagery dataset and testing it on another (e.g., train on PASCAL VOC [12], test on COCO [43]) [10, 32, 33, 54]. While this style of evaluation is convenient, it emulates a mere sliver of potential open-world scenarios that we may encounter in the real-world. Existing evaluations make two key assumptions: (1) they assume we have access to a large diverse set of training data that contains exhaustive labels for nearly every class that we would want to detect during deployment; and (2) they assume that all open-world applications consider *every* unlabeled object to be of interest. In reality, both the quality of the labeled data and the desired behavior of the model can vary significantly depending on the application in which it is used. For example, in robotics applications it may be more important to localize all potential regions of interest, down to the level of ambient objects such as light switches and power outlets. However, in an application such as a vehicle identification system, it is more critical to detect novel types of vehicles than, for instance, buildings and trees. In this work, we design several novel challenges to simulate varying degrees of label bias to more rigorously evaluate our method. Specifically, a *training class diversity* challenge restricts ID class coverage, a *semi-supervised* challenge directly reduces the amount of labeled samples we have, and a *ships* challenge tests the models in a different domain with a uniquely constrained set of OOD objects of interest.

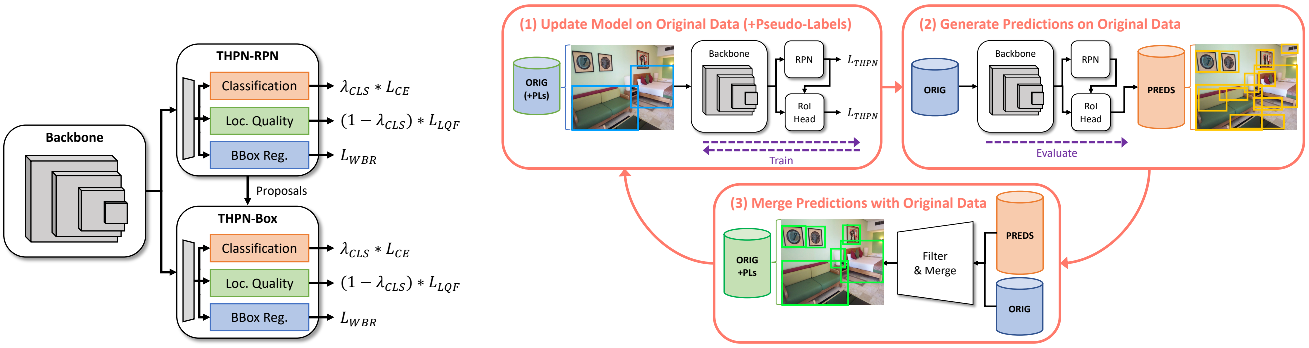
To address these challenges, we develop a Tunable Hybrid Proposal Network (THPN) that leverages two types of object representation: (1) classification-based objectness and (2) localization-based objectness. Classification-based objectness is employed in the canonical Region Proposal Network (RPN) [51, 61, 62], and frames object learning as a discriminatory task. This works well for detecting ID objects, but struggles to detect OOD objects as it explicitly learns that all non-labeled regions are *background* [13, 32, 33]. Localization-based objectness, introduced recently by Kim et al.'s Object Localization Network (OLN) [32], frames objectness as the localization quality [28, 59] between a region and any ground truth box. This approach promotes a less discriminative detector that more readily generalizes to dissimilar OOD classes. By using both representations simultaneously, THPN is capable of the best of both worlds. The behavior of THPN can be easily tuned with a single hyperparameter $\lambda_{CLS} \in [0, 1]$ which determines how significantly the model weights classification-based objectness versus localization-based objectness. Depending on the needs of the application, THPN can operate as a conservative ID expert using a large λ_{CLS}, an aggressive OOD object detector using a small λ_{CLS}, or anywhere in between. In addition, THPN uses a novel open-world-aware self-training procedure which bolsters the existing label set with high-quality pseudo-labels [26]. Unlike common self-training solutions [2, 26, 58], our approach does not require any auxiliary samples to generate pseudo-labels on, and does not require full retraining in each round. Finally, we develop a dynamic loss to address challenges such as class-imbalance and imperfect pseudo-label targets.

THPN outperforms all baselines in all evaluation settings that we consider. On the common VOC→COCO open-set benchmark, where models are trained on VOC-class labels and tested on non-VOC COCO classes, THPN vastly improves upon RPN (+18.9% AR100) and OLN (+5.7% AR100). Fig. 1 shows a summary of results across several of our novel evaluation challenges in terms of ALL object recall. Note that THPN can easily surpass OLN in more difficult biased tasks without sacrificing performance on low-bias tasks. For example, THPN trained on a five-class subset of VOC achieves higher OOD recall than an OLN trained on the entire 20-class VOC subset. Furthermore, a THPN trained on a random 10% subset of the original VOC labels rivals the OOD recall of an OLN trained on 100% of the labels. On the *ships* challenge, THPN achieve a ∼3x recall improvement over Faster R-CNN on OOD ships. Overall, THPN's flexibility enables it to be a better general solution for open-set/world detection problems.

2. Related work

Class-agnostic object proposal. Early methods for class-agnostic object detection rely on handcrafted image features such as Gaussian filters and edges [1,34,50,60,75], but the advent of deep learning has rendered these heuristic-driven approaches obsolete [36, 51]. RPN and its variants [51, 61, 62] *learn* to identify a reduced set of regions that have a high likelihood of containing objects. RPN can be trained inline as part of a two-stage detection architecture [6, 21, 41, 51] to attain impressive accuracy on ID classes. The problem with RPN is that it overfits to the ID categories [32,33,54]. Object Localization Network (OLN) [32] combats this overfitting by replacing the classification heads of a class-agnostic Faster R-CNN with localization quality prediction heads to avoid treating OOD objects as *background*. Konan et al. [33] use unknown object masking to reduce the number of false-negative regions sampled during training. Finally, Saito et al. [54] use a background erasing augmentation and a multi-domain training strategy to reduce the bias of classification-based proposal networks. Uniquely, our solution combines both objectness representations with a novel self-training procedure to better address a variety of open-world scenarios.

Open-set/world detection. Unlike class-agnostic proposal networks, full object detection models also classify the objects. *Open-set* detectors accept that OOD objects will inevitably be encountered during inference, and attempt to flag them as *unknown*. Dhamija et al. [10] find that closed-set models frequently misclassify OOD objects as ID classes despite training with an explicit *background* class. Miller et al. [45, 46] use dropout sampling [15] to estimate uncertainty and reduce these open-set false positives. Recently, virtual outliers [11] and contrastive learning [20, 31], have been used to separate known and unknown instances in feature space. Joseph et al. [29] present the first attempt at an *open-world* detection system, which



(a) THPN's hybrid architecture has heads for both classification-based and localization-based objectness. The total objectness loss is a linear interpolation between the CLS and LQ head losses.

(b) Our self-training algorithm consists of three stages. First, update the model on the original data (plus pseudo-labels we may have). Second, use the trained model to generate predictions on the *original* images. Third, filter and merge the predictions with the original ground truth labels to create an updated label set with pseudo-labels (PLs). Repeat this process iteratively.

Figure 2. Overview of THPN's architecture and training procedure.

not only detects OOD objects, but also incrementally learns the newly encountered classes on the fly. Since then, several works have incrementally improved open-world detection [19, 66, 70, 74]. Critically, these open-set/world detectors rely on a classification-based RPN to provide proposals for both ID and OOD objects, meaning that many OOD objects are likely to go undetected. In this work, we focus on the development of a more powerful proposal network, which can be integrated into such systems in the future.

Self-training. Self-training is a powerful technique in semi-supervised learning, where only a subset of the training data has labels [2, 47]. Based on the classic concept of a delay-feedback algorithm [14, 57], self-training uses a trained model to generate pseudo-labels on unlabeled data which are then used to bolster the existing training set, leading to better subsequent models [2, 26, 76]. While self-training is most commonly employed in image classification tasks [65,67,68], some works have used it for the object detection task [30, 52, 58, 69, 76] to improve closed-set performance in semi-supervised and few-shot scenarios. Different from previous works, we apply the principles of self-training to the open-world proposal problem. We find that this setting allows us to loosen many of the requirements made by existing approaches [26,47,58]. For example, our method requires no auxiliary data, and iteratively fine-tunes the same model rather than fully re-training it.

3. Learning open-set proposals

To build intuition, we formalize the open-set object proposal problem. Generally, in an object detection task we have a set of *known* (ID) object classes $\mathcal{K} = \{1, 2, \ldots, C\} \subset \mathbb{N}^+$ that we have labels for. Typically there are also a considerable number of unlabeled instances of *unknown* (OOD) classes $\mathcal{U} = \{C+1, \ldots\} \subset \mathbb{N}^+$ that coexist with the known instances in the images. The goal of the open-set object proposal task is to train a model $\mathcal{M}$ parameterized by θ to detect and localize *all* object instances of potential interest in a test set (i.e., all instances in the set $\mathcal{K} \cup \mathcal{U}$). For a given test image X, the proposal network's function is $\mathcal{M}(X;\theta) = \{[x, y, w, h, s]_{j=1\ldots N}\}$, where x, y, w, and h denote the center coordinates, width, and height of the bounding box, respectively. The predicted "objectness" score $s \in [0, 1]$ is the confidence that box j contains an object. Although the proposal task differs from the full open-set detection task (in which the model also predicts the class of each object), most current state-of-the-art open-set/world detection systems rely on proposal networks to produce high-recall candidate regions [10, 20, 29, 74]. Ultimately, the upper bound of performance achievable by such systems is premised on the recall of the proposal network.

4. Tunable Hybrid Proposal Network (THPN)

Our primary goal with THPN is to introduce a flexible proposal network that can be readily adapted to many open-world environments. Controllable by a single hyper-parameter, our idea is to allow the user to adjust the model's willingness to detect OOD objects that are dissimilar to the labeled classes depending on their application's requirements To achieve this, we develop a novel training algorithm (Sec. 4.1), model architecture (Sec. 4.2), and dynamic loss (Sec. 4.2). Sec. 4.3 contains implementation details.

4.1. Self-training procedure

One major drawback of existing proposal networks is that their generalization is largely dependent on the quantity and diversity of the labeled training data. Self-training can significantly mitigate this issue by artificially adding labels to the dataset. Self-training is the process of training a model on available labeled data, running inference on unlabeled inputs to generate high-quality pseudo-labels, and training a new model on the union of the original training data and the pseudo-labeled set [26]. While self-training is most commonly used for semi-supervised learning of closed-set tasks [26, 47, 58], we are the first to tailor this powerful regularization for open-world object proposals. Specifically, we develop a three-stage self-training algorithm that is outlined in Fig. 2b. The overall workflow is as follows. In Stage 1, we train the model on the original labeled data; in Stage 2, we evaluate the trained model *on the original training images* to generate predictions; and in Stage 3, we filter predictions by score and merge the highest scoring predictions with the original ground truth labels. We can repeat this loop by training the model again on the *updated* label set to incrementally improve pseudo-label quality and thus subsequent model generalization.

Note that unlike existing self-training implementations, our method does not require the user to cull auxiliary unlabeled data. This is because in virtually all real-world detection data there exists a multitude of unlabeled OOD (and ID) objects that coexist in the same images as the labeled ID instances. Also note that existing self-training work [26, 47, 58] retrains the model "from scratch" in each round. This approach is very expensive as it involves training for $(r*E)$ epochs, where r is the number of self-training rounds and E is the number of epochs in the standard training schedule. A more efficient approach is to repeatedly fine-tune the same model with the updated label set. We observe fine-tuning convergence within $E/4$ epochs, so the total training cost of THPN is $(E + r * (E/4))$ epochs.

Another important design detail is how we "filter & merge" newly proposed boxes into the ground-truth label set in Stage 3. First, to avoid adding redundant labels we discard all predictions that overlap a ground truth box by 0.7 IoU. Next, we filter the remaining labels based on predicted objectness. While previous methods use simple thresholding of confidence [26, 58, 69], we find that this approach does not provide enough granular control over the amount of predictions we allow to become pseudo-labels because DNNs are notoriously poorly calibrated [18]. Instead, we take the top P non-overlapping predictions, where P is p% of the number of original training instances (p is a hyperparameter). With this approach, we can precisely control the amount of pseudo-labels we add relative to the number of original ground truth labels, making performance consistent regardless of the dataset size or the objectness metric used.

4.2. Model architecture and losses

There are two known meta-strategies for learning-based proposal networks which are differentiated by how a region's "objectness" is quantified. *Classification-based* approaches, such as RPN and class-agnostic Faster R-CNN [29, 51], directly predict a region's likelihood of containing an ID object. These models are trained to explicitly discriminate ID objects vs. background, meaning any OOD objects present in the training images are learned as negatives (i.e., background). Thus, these models significantly overfit to the training classes [13, 32, 33]. Alternatively, *localization-based* (i.e., classification-free) models [32] predict a region's localization quality (e.g., centerness [59], IoU [28]) with respect to the nearest ground-truth box and treat this as a notion of objectness. In essence, this changes the task from "*What is the likelihood this region contains an object?*" to "*How well does this region localize the nearest object?*". Because predicting localization quality is not discriminatory, the model is not explicitly biased towards the ID classes. While this allows for better OOD detection, it comes at the cost of reduced ID proficiency. For more details on these methods, see Appendix A.

The key insight of our work is that the best objectness representation to use is dependent on the data and the desired behavior of the system. For example, applications that prioritize ID recall would benefit from classification-based objectness, while applications that require detecting *all* objects would benefit from localization-based objectness. Our solution is to leverage a hybrid objectness representation that can be readily tuned to suit the full spectrum of applications and environments. To realize this design we use a two-stage detection architecture, where a first stage THPN-RPN produces a set of reasonable candidate regions, and a second stage THPN-Box refines these candidate regions and makes the final objectness prediction. To allow THPN to use both objectness representations, we use three prediction heads in THPN-RPN and THPN-Box (see Fig. 2a). For each anchor, a classification head (CLS) predicts the likelihood that a region contains an object, a localization quality head (LQ) predicts a quality score (i.e., centerness [59] in THPN-RPN and IoU [28] in THPN-Box), and a bounding box regression head (BOX) predicts the box offsets.

The loss function for both THPN stages is defined as:

$$\begin{aligned} L_{THPN}\big(\{c_i\}, \{q_j\}, \{t_i\}\big) = \\ \lambda_{CLS} \frac{1}{N_{CLS}} \sum_i L_{CE}(c_i, c_i^*) \\ +(1-\lambda_{CLS}) \frac{1}{N_{LQ}} \sum_j L_{LQF}(q_j, q_j^*) \\ +\lambda_{BOX} \frac{1}{N_{BOX}} \sum_i c_i^* L_{WBR}(t_i, t_i^*). \end{aligned} \tag{1}$$

Due to the fact that we use two different sets of sampled anchors (based on different sampling criteria) to compute the losses, we use i to denote the indexes of anchors for the CLS and BOX heads, and j to denote the indexes of anchors for the LQ head. Thus, c_i, q_j, and t_i are the predicted object likelihood, localization quality score, and box offsets, respectively, and c_i^*, q_j^* and t_i^* are the corresponding targets. The total loss is composed of three terms. The first is the cross-entropy loss L_{CE}

from the CLS head; the second is the Localization Quality Focal Loss L_{LQF} (detailed below) from the LQ head; and the third is Weighted Box Regression Loss L_{WBR} (also detailed below) from the BOX head. Importantly, the first two terms together represent the total objectness loss, which can be balanced using the λ_{CLS} hyperparameter. By adjusting $\lambda_{CLS} \in [0, 1]$, the user can significantly alter the behavior of the resulting model. The smaller λ_{CLS} is set, the more the model is incentivized by localization quality, increasing the propensity of detecting diverse OOD objects. During inference-time (and training time to collect proposals for THPN-Box), we use the same linear interpolation to blend the predicted scores from the CLS and LQ heads. The final scores are computed by $\mathtt{s} = \lambda_{\text{CLS}} * \mathtt{cls\_scores} + (1 - \lambda_{\text{CLS}}) * \mathtt{lq\_scores}$.

Localization quality focal loss. A key challenge that proposal networks encounter is data imbalance. The source of imbalance in our case is two-fold: (1) the natural training distribution is often long-tailed, and (2) the pseudo-labels may only cover a handful of samples from each OOD class. By failing to account for this imbalance we risk overfitting the LQ head to the frequently occurring training classes. To combat this, we devise a Localization Quality Focal Loss (L_{LQF}) which dynamically weights the loss contribution of each sampled region (i.e., anchor) based on the correctness of the model's predicted quality score for that region. Specifically, the loss of the j^{th} sampled anchor is:

$$L_{LQF}(q_j, q_j^*) = \left|q_j^* - q_j\right|^{\gamma} L_{BCE}(q_j, q_j^*) \qquad (2)$$

where q_j and q_j^* are the predicted and target localization quality for the given anchor, respectively. γ is a hyperparameter to scale the significance of the weighting (we use γ=2). While inspired by the original Focal Loss [42], L_{LQF} makes a critical modification to allow it to be used with floating-point targets. Also, while L_{LQF} bears similarity to the recently proposed QFL [38, 39], the goal of L_{LQF} is different as it encourages accurate localization quality predictions on difficult pseudo-label targets.

Weighted box regression loss. Another unique challenge that we face is imperfect pseudo-labels. Particularly when dealing with unseen object categories, it is not safe to assume that the pseudo-label bounding boxes will be of hand-crafted quality. Because box targets are represented as fixed Dirac delta distributions [39] with no encoding of uncertainty, we must be judicious with how much we optimize against certain pseudo-label targets. Naively training on flawed boxes will hinder the model's ability to make fine-grained localization adjustments. We address this problem with the L_{WBR} loss, which scales the box regression loss from different pseudo-labels depending on their estimated quality during pseudo-label generation. To scale the loss, we downweight the contribution from anchors matched to pseudo-label targets by the respective pseudo-label's score.

Split	Method	OOD AR10	OOD AR100
VOC	RPN [51]	7.4	20.0
	GA-RPN [62]	11.9	27.7
	Cascade RPN [61]	12.6	27.7
	Faster R-CNN [51]	11.6	25.1
	FCOS [59]	10.5	24.4
	FCOS-OWP [33]	14.5	31.3
	LDET [54]	18.2	30.8
	OLN [32]	18.4	33.2
	THPN ($\lambda_{CLS} = 0$)	**21.6**	**38.9**

Table 1. Results on the *COCO benchmark* challenge.

Then the loss from the i^{th} anchor is:

$$L_{WBR}(t_i, t_i^*) = s_i^{\beta} L_1(t_i, t_i^*). \qquad (3)$$

Recall, t_i and t_i^* are the predicted and target box offsets, respectively. Here, $s_i \in [0, 1]$ is the quality score predicted for pseudo-label t_i^* in Stage 2. Note that for ground-truth targets, we assume s_i=1. The hyperparameter β scales how severely we downweight the loss from anchors matched to lower scoring targets (we use β=2). Intuitively, this objective encodes uncertainty into each pseudo-label's box coordinates based on its predicted quality.

4.3. Implementation details

THPN is built on the PyTorch-based [48] mmdetection library [7]. We use a ResNet-50 [22] with a Feature Pyramid Network (FPN) [41] as a backbone. We also use one anchor per feature location and $\lambda_{BOX} = 10$ and $\lambda_{BOX} = 1$ for THPN-RPN and THPN-Box, respectively, in accordance with Kim et al. [32]. Multi-level features from the top scoring anchors are extracted with RoIAlign [21]. In this work, we train all THPN models using crop & zoom augmentations. We train for E=16 epochs initially, and E/4=4 epochs in each succeeding self-training round. We use r=3 self-training rounds per model and set p=30 to incur a 30% increase in total labels due to pseudo-labels. Note that in each round of self-training, we generate all new pseudo-labels instead of re-using them from previous rounds. Models are trained on four NVIDIA V100 GPUs with a batch-size of two images per device.

5. Experiments

To thoroughly evaluate the performance of THPN we consider four generalization challenges which go far beyond the evaluations of contemporary methods [29, 32, 33, 54]. Our core experimental methodology is to divide the COCO dataset [43] into several ID:OOD disjoint class splits, such that the union of the ID and OOD classes equals all 80 COCO classes. During training, we only assume access to labels of the ID classes in the training set. Importantly, *THPN is only ever exposed to images that contain at least one ID label during training and pseudo-label*

Split	Model	Images / Instances	OOD AUC	OOD AR10	OOD AR100	OOD AR1k	ID AUC	ID AR10	ID AR100	ID AR1k	ALL AUC	ALL AR10	ALL AR100	ALL AR1k
COCO40	Faster R-CNN	104k / 623k	26.6	17.5	36.0	51.4	44.4	41.4	58.3	63.2	39.0	33.8	51.7	60.0
	OLN	104k / 623k	33.1	25.8	44.8	54.6	42.1	34.6	57.2	65.0	38.9	30.5	53.3	62.2
	THPN ($\lambda_{CLS} = 0$)	104k / 810k	34.1	26.9	45.9	**56.0**	40.6	31.8	55.7	64.0	38.1	28.8	52.5	61.7
	THPN ($\lambda_{CLS} = 0.10$)	104k / 810k	**34.8**	**29.8**	**46.0**	55.3	44.0	39.6	58.1	64.6	40.7	35.1	54.3	62.0
	THPN ($\lambda_{CLS} = 0.25$)	104k / 810k	33.6	27.6	44.4	55.1	45.6	42.8	59.5	65.2	**41.6**	**37.1**	**54.9**	62.4
	THPN ($\lambda_{CLS} = 0.50$)	104k / 810k	30.9	22.5	42.1	54.5	**46.0**	**43.3**	**60.2**	**65.7**	41.3	36.5	54.7	**62.7**
VOC	Faster R-CNN	95k / 493k	19.3	11.6	25.1	42.4	46.7	45.1	60.7	64.7	34.4	29.9	44.8	55.1
	OLN	95k / 493k	24.8	18.4	33.2	45.0	44.8	40.1	59.3	66.1	35.5	29.1	47.5	56.9
	THPN ($\lambda_{CLS} = 0$)	95k / 641k	**28.8**	21.6	**38.9**	**49.7**	43.8	37.0	58.9	65.9	36.6	28.5	49.8	**59.0**
	THPN ($\lambda_{CLS} = 0.10$)	95k / 641k	27.9	**22.0**	37.1	48.0	46.8	44.2	60.9	66.5	**38.0**	32.9	**50.2**	58.5
	THPN ($\lambda_{CLS} = 0.25$)	95k / 641k	25.3	18.4	33.7	46.3	48.1	46.8	62.1	67.0	37.6	**33.1**	49.4	58.1
	THPN ($\lambda_{CLS} = 0.50$)	95k / 641k	22.2	14.7	29.6	44.8	**48.4**	**47.0**	**62.8**	**67.4**	36.6	32.1	48.0	57.7
VOC5	Faster R-CNN	74k / 357k	16.3	9.8	20.7	38.1	48.1	47.6	62.2	65.6	29.1	24.8	37.4	49.6
	OLN	74k / 357k	20.3	14.1	26.9	40.1	47.6	45.2	61.7	67.8	31.0	25.7	40.8	51.6
	THPN ($\lambda_{CLS} = 0$)	74k / 465k	**25.6**	**18.4**	**34.7**	**46.6**	45.8	41.1	60.6	66.9	**33.3**	26.2	**44.9**	**55.1**
	THPN ($\lambda_{CLS} = 0.10$)	74k / 465k	23.7	17.6	31.5	43.9	48.3	47.1	62.4	67.4	33.3	**28.4**	43.8	53.7
	THPN ($\lambda_{CLS} = 0.25$)	74k / 465k	21.8	15.5	28.6	42.3	49.4	49.2	63.5	67.7	32.7	28.3	42.6	52.9
	THPN ($\lambda_{CLS} = 0.50$)	74k / 465k	19.0	13.0	24.3	40.0	**49.7**	**49.5**	**63.9**	**68.1**	31.2	27.2	40.3	51.6
Animal	Faster R-CNN	24k / 63k	11.5	6.0	13.5	31.3	53.9	58.9	67.1	69.4	14.6	9.8	17.5	34.1
	OLN	24k / 63k	13.3	8.2	16.4	31.5	55.8	59.7	69.7	73.2	16.4	11.9	20.3	34.6
	THPN ($\lambda_{CLS} = 0$)	24k / 81k	**18.2**	10.1	**24.9**	**39.5**	54.5	57.4	68.7	72.3	**20.9**	13.5	**28.1**	**42.0**
	THPN ($\lambda_{CLS} = 0.10$)	24k / 81k	17.0	**10.3**	22.9	36.6	56.1	60.6	69.9	73.0	19.8	**13.9**	26.3	39.3
	THPN ($\lambda_{CLS} = 0.25$)	24k / 81k	16.1	10.1	20.9	35.4	56.5	61.7	70.2	73.0	19.0	13.8	24.5	38.2
	THPN ($\lambda_{CLS} = 0.50$)	24k / 81k	14.7	8.7	18.7	34.3	**56.6**	**61.9**	**70.3**	**73.0**	17.8	12.6	22.5	37.1

Table 2. Results on the *training class diversity* challenge.

generation. Thus, our implementation of THPN does not use any unlabeled training images, just like any non-self-trained baseline. In Sec. 5.1, we consider the common VOC→COCO benchmark. Sec. 5.2 and Sec. 5.3 cover our *training class diversity* and *semi-supervised* challenges, respectively. In Sec. 5.4, we test THPN on an open-set ship detection task. Finally, Sec. 5.5 contains an analysis and ablation study of several model design choices.

5.1. COCO benchmark challenge

The first challenge we consider is the cross-category generalization task which has been used as the main benchmark by various recent open-world proposal works [32,33,54]. In this task, we consider the 20 VOC [12] classes to be ID and the 60 remaining (non-VOC) classes to be OOD. We train a model on ID labels only and evaluate by computing Average Recall (AR@k detections per image) [25] on the OOD instances in the validation set. We do not consider Average Precision (AP) as it is unfair to penalize false positives unless the dataset is exhaustively labeled. Performance on this task signifies a model's ability to generalize to unseen classes. Tab. 1 contains the results. We set $\lambda_{CLS} = 0$ in this test to maximize OOD performance. THPN outperforms all baselines, surpassing the strongest (OLN) by +3.2% AR10 and +5.7% AR100. In Appendix B, we evaluate THPN against several learning-free methods such as Selective Search [60] and EdgeBoxes [75], and find that THPN beats the strongest baseline by over 2x.

5.2. Training class diversity challenge

While the *COCO benchmark* challenge provides some notion of a model's open-world aptness, it is a fairly optimistic scenario. Even though there are only 20 training classes, they cover a wide range of COCO's semantic "superclasses" like animal, vehicle, and household-object. A model trained on these classes is exposed to a variety of scene types (e.g., indoor, outdoor, etc.), thus improving its generalization [64]. Also, while OOD recall alone is important, it does not tell the full story of a model's performance. It is equally critical to measure the model's recall of ID objects, and ultimately the recall of ALL object classes (ID *and* OOD). Our hypothesis is that in the case of strong label bias, existing proposal networks will struggle to generalize to OOD instances without sacrificing ID performance. Meanwhile, THPN's ability to leverage both classification-based and localization-based objectness, as well as high-quality pseudo-labels, will enable it to excel. To test this hypothesis, we curate four ID class splits with increasing difficulty/bias: Half of COCO (COCO40), VOC classes (VOC), a sample of five VOC classes (VOC5), and a highly biased split of only animal classes (Animal). See Appendix H for the exact classes used. Note that AUC serves as summary metric of AR over several k thresholds (10–1000) [32].

Tab. 2 shows the results of this experiment. Note that the results can be interpreted differently depending on the goal of the user. If the goal is to maximize OOD performance, THPN with a small λ_{CLS} (≤ 0.25) outperforms the baselines in all cases. Interestingly, the margins of improvement of OLN over Faster R-CNN *decrease* as we increase label bias (e.g., +8.8% AR100 on COCO40 down to +2.9% AR100 on Animal), while THPN's margins over Faster R-CNN *increase* (e.g., +10.0% AR100 on COCO40 up to +11.4% AR100 on Animal for $\lambda_{CLS} = 0$). This finding confirms our hypothesis that THPN models are far less

Split	Model	Images / Instances	OOD AUC	OOD AR10	OOD AR100	OOD AR1k	ID AUC	ID AR10	ID AR100	ID AR1k	ALL AUC	ALL AR10	ALL AR100	ALL AR1k
VOC (50%)	Faster R-CNN	75k / 246k	18.7	11.7	24.1	40.9	44.8	42.7	58.5	63.1	33.1	28.5	43.2	53.6
	OLN	75k / 246k	23.8	17.7	31.7	43.8	44.4	39.5	58.8	65.7	34.9	28.5	46.7	56.3
	THPN ($\lambda_{CLS}=0$)	75k / 320k	**27.9**	**21.2**	**37.4**	**48.2**	43.7	38.6	58.0	65.0	36.2	29.4	**48.7**	**57.8**
	THPN ($\lambda_{CLS}=0.10$)	75k / 320k	25.7	19.6	34.0	45.9	46.1	44.1	59.7	65.7	**36.7**	**32.0**	48.1	57.2
	THPN ($\lambda_{CLS}=0.25$)	75k / 320k	23.8	16.8	31.6	45.2	47.1	**45.7**	60.8	66.1	36.5	32.0	47.7	57.1
	THPN ($\lambda_{CLS}=0.50$)	75k / 320k	21.3	13.8	28.1	43.7	**47.1**	45.4	**61.1**	**66.3**	35.5	30.9	46.4	56.6
VOC (25%)	Faster R-CNN	56k / 123k	17.9	11.2	22.9	39.2	42.7	40.1	55.8	60.9	31.6	27.0	41.1	51.6
	OLN	56k / 123k	21.9	16.6	28.8	40.7	43.2	38.3	57.1	64.1	33.4	27.5	44.5	54.0
	THPN ($\lambda_{CLS}=0$)	56k / 160k	**26.2**	**19.7**	**34.9**	**46.3**	43.1	38.6	56.9	64.0	35.2	28.9	**47.0**	**56.4**
	THPN ($\lambda_{CLS}=0.10$)	56k / 160k	24.2	17.9	32.1	44.8	44.7	42.4	58.0	64.2	**35.3**	30.5	46.4	55.9
	THPN ($\lambda_{CLS}=0.25$)	56k / 160k	22.7	15.9	30.2	43.7	45.5	**43.9**	58.8	64.4	35.1	**30.6**	46.0	55.5
	THPN ($\lambda_{CLS}=0.50$)	56k / 160k	20.6	13.4	26.9	43.0	**45.6**	43.6	**59.0**	**64.7**	34.3	29.7	44.7	55.4
VOC (10%)	Faster R-CNN	33k / 49k	16.2	10.4	20.5	35.8	39.5	36.2	51.8	57.8	29.1	24.5	37.9	48.4
	OLN	33k / 49k	19.8	15.2	25.7	37.3	40.8	36.3	53.6	61.0	31.3	26.0	41.3	50.8
	THPN ($\lambda_{CLS}=0$)	33k / 64k	**24.0**	**18.1**	**31.8**	**43.5**	41.0	36.5	54.0	61.6	33.1	27.1	**44.0**	**53.9**
	THPN ($\lambda_{CLS}=0.10$)	33k / 64k	23.0	17.2	30.2	42.4	42.3	39.5	55.0	61.9	**33.5**	**28.6**	43.9	53.6
	THPN ($\lambda_{CLS}=0.25$)	33k / 64k	20.9	14.8	27.4	40.5	43.0	**40.9**	55.5	62.0	33.0	28.6	43.0	52.8
	THPN ($\lambda_{CLS}=0.50$)	33k / 64k	19.0	12.4	25.0	39.7	**43.0**	40.5	**55.8**	**62.1**	32.3	27.6	42.1	52.5

Table 3. Results on the *semi-supervised* challenge.

Split	Model	Images / Instances	OOD AUC	ID AUC	ALL AUC
Military	Faster R-CNN	1.5k / 4.7k	11.8	65.6	33.3
	OLN	1.5k / 4.7k	23.0	69.0	41.3
	THPN ($\lambda_{CLS}=0$)	1.5k / 6.1k	**29.0**	68.8	**44.7**
	THPN ($\lambda_{CLS}=0.10$)	1.5k / 6.1k	26.6	**69.3**	43.6
Civilian	Faster R-CNN	1.0k / 4.4k	16.0	33.4	24.6
	OLN	1.0k / 4.4k	38.7	35.6	36.9
	THPN ($\lambda_{CLS}=0$)	1.0k / 5.7k	**49.8**	**36.5**	**42.8**
	THPN ($\lambda_{CLS}=0.10$)	1.0k / 5.7k	46.3	36.4	41.0

Table 4. Results on the *ships* challenge.

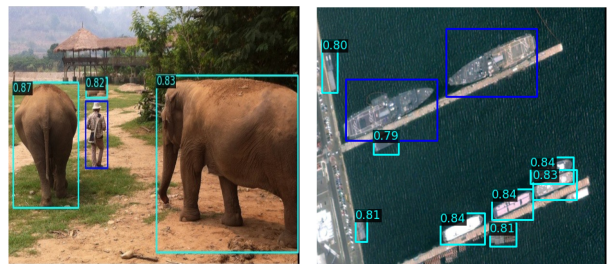


Figure 3. THPN training samples (p=30%). Blue boxes are ID labels and cyan boxes are pseudo-labels with objectness $s \in [0, 1]$.

prone to overfitting than the baselines. If the goal is to maximize ID performance, THPN can also be beneficial. With a larger λ_{CLS}, THPN can outperform Faster R-CNN on low-bias splits and OLN on high-bias splits. Finally, all THPN variants outperform both baselines in terms of ALL recall, but the choice of λ_{CLS} can make a large difference. On COCO40, where 72% of total instances are from ID classes, users should choose a larger λ_{CLS} as ID performance has more influence on ALL recall. On more biased tasks, tasks with more OOD samples than ID samples, or tasks where OOD recall is paramount, a small λ_{CLS} is more appropriate. These results showcase the power of allowing the user to influence the ID/OOD tradeoff depending on their needs.

5.3. Semi-supervised challenge

Another challenging yet realistic scenario that is not considered by existing open-world detection works is a partially labeled training dataset that only contains labels for a subset of the existing ID instances. In this challenge, we assume a fraction of the original VOC-class instances are labeled. We randomly subsample each class's label count by the same percentage. Our hypothesis is that THPN's self-training procedure will allow it to generate pseudo-labels on both unlabeled OOD *and* unlabeled ID instances, leading to drastically improved overall recall in these cases.

Tab. 3 contains the results of this challenge. We consider having 50%, 25%, and 10% of available labels (to avoid redundancy the 100% results can be found in Tab. 2 under "VOC"). In terms of OOD generalization, THPN with $\lambda_{CLS} = 0$ performs significantly better than OLN. Importantly, as we reduce the amount of labeled data, THPN's margin of improvement over OLN increases (e.g., +5.7% AR100 on VOC-100% up to +6.1% AR100 on VOC-10%). For ID performance, we again find a benefit to using a larger λ_{CLS} to use a more classification-based objectness. Overall, we find that the best setting to optimize ALL-AUC on all splits is $\lambda_{CLS} = 0.10$. Under this setting, a THPN trained with 25% of labeled samples (and 59% of images) can outperform a Faster R-CNN trained with all available data! This finding indicates that the optimal λ_{CLS} is influenced more by class diversity than label quantity. Overall, we believe our model's ability to gracefully deal with partially labeled datasets is a key advantage.

5.4. Ships challenge

To examine versatility, our final challenge is to consider a domain outside of natural imagery. We use the ShipRSImageNet dataset [73], which contains satellite imagery of oceanic regions around the world, with annotations for both military and civilian/merchant ships. Detection models in this domain are challenged with limited data and significant

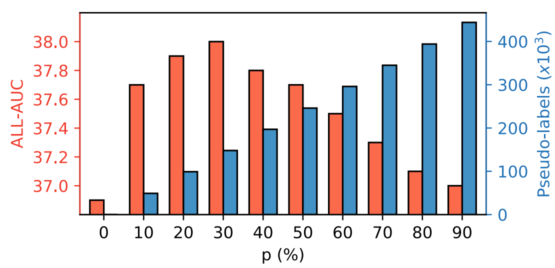


Figure 4. Hyperparameter p's impact on ALL recall and pseudo-label count for a THPN trained on the VOC split.

Model	OOD-AUC	ID-AUC	ALL-AUC
THPN ($\lambda_{CLS} = 0.10$)	27.9	46.8	38.0
No self-training	25.8 (-2.1)	46.5 (-0.3)	36.9 (-1.1)
No *LQ* head	20.3 (-7.6)	47.9 (+1.1)	35.6 (-2.4)
No *CLS* head	28.8 (+0.9)	43.8 (-3.0)	36.6 (-1.4)
No L_{LQF}	27.2 (-0.7)	46.1 (-0.7)	37.3 (-0.7)
No L_{WBR}	27.5 (-0.4)	46.6 (-0.2)	37.6 (-0.4)
Finetune → Retrain	27.4 (-0.5)	46.0 (-0.8)	37.3 (-0.7)
1x schedule	27.4 (-0.5)	46.1 (-0.7)	37.4 (-0.6)
No data aug.	25.1 (-2.8)	47.5 (+0.7)	37.2 (-0.8)

Table 5. Ablation study on the VOC split.

variations from natural conditions (e.g., reflection, weather, lighting). For this experiment, we consider two splits: (1) train on only military vessels and (2) train on only civilian vessels. From either perspective, being able to detect *all* ships in a given area is a critically important behavior. We argue that this challenge represents a pragmatic view of open-world detection; where the goal is not to detect *every* type of object (e.g., cars, buildings), but only instances of a particular superclass (i.e., ships).

For consistency, we use the same models and experimental settings as our COCO-based challenges. A summary of the results are in Tab. 4 and the full results are in Appendix E. Note that localization-based objectness is particularly effective for generalizing to OOD ships. Our observation is that in low-data conditions, classification-based objectness learners tend to severely overfit to specific features of the ID classes. For this reason, OLN transcends Faster R-CNN in terms of OOD recall, and $\lambda_{CLS} = 0$ is the optimal setting for THPN. This is an eye-opening result as many recent works achieving state-of-the-art performance on ship detection tasks use a variant of Faster R-CNN [35, 40, 72, 73]. While localization-based objectness provides one advantage, our self-training procedure boosts performance much higher than a vanilla OLN. On the Civilian split, THPN improves OOD recall over OLN and Faster R-CNN by +11.1 and +33.8 AUC, respectively! In terms of ID recall, THPN with a small λ_{CLS} can slightly exceed OLN. As a result, THPN's recall on ALL ships is vastly superior to either baseline. This result portrays THPN's capability across domains, and call into question the common algorithms used for remote sensing detection.

5.5. Analysis of model design

Training a THPN involves making a few key design decisions. First, we explore the implications of hyperparameter p (i.e., how many pseudo-labels to allow) on ALL recall for the VOC split in Fig. 4. We find that using $p = 30\%$ is a good rule of thumb in all scenarios, though the performance is not very sensitive to p. We visualize two pseudo-labeled samples from THPN with p=30% in Fig. 3. Notice that the pseudo-labels are of high quality in both domains, achieving a reasonable balance of recall and precision on OOD objects. See Appendix F for more pseudo-labeled samples.

Tab. 5 contains an ablation study for THPN on the VOC split. Self-training is responsible primarily for improved OOD performance, however THPN ($\lambda_{CLS} = 0.10$) without self-training still outperforms OLN significantly (OLN achieves 24.8 OOD-AUC, 44.8 ID-AUC, 35.5 ALL-AUC). As expected, removing the LQ head ($\lambda_{CLS} = 1$) results in much worse OOD recall with a slight benefit to ID recall, while removing the CLS head ($\lambda_{CLS} = 0$) yields worse ID recall but improved OOD performance. Our two dynamic loss functions, L_{LQF} and L_{WBR}, also play an important role especially for OOD detection. In additional experiments we find that these losses become more important in more biased class splits. For example, on the Animal split, removing these two losses leads to a 2.4 OOD-AUC reduction. Interestingly, we find that our finetuning-based self-training is not only more efficient, but better performing than the conventional retraining-based approach. Finally, while the longer (initial) 16-epoch training schedule is beneficial for both OOD and ID recall, the crop & zoom augmentations mainly benefit OOD generalization.

6. Conclusion

In the scope of open-world detection tasks, the variation of data bias and desired model behavior renders static proposal networks insufficient. In this work, we instead introduce a powerful new class of proposal solution that can be easily adjusted to suit the gamut of challenging open-world scenarios. Our novel evaluation challenges test models in a variety of conditions, ranging from large-scale academic tasks, to tasks with severe degrees of ID class bias and partial labels. We also demonstrate our model's superiority in realistic remote sensing applications. THPN's superior recall of both ID and OOD objects has the potential to enhance a variety of open-world applications, and we hope that our evaluation protocols can serve as touchstones to inspire even more robust models in the future.

Acknowledgements

This work is supported by NSF CNS-2112562, IIS-2140247, and AFRL FA8750-21-1-1015.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2189–2202, 2012. 2

[2] Massih-Reza Amini, Vasilii Feofanov, Loïc Pauletto, Emilie Devijver, and Yury Maximov. Self-training: A survey. *CoRR*, abs/2202.12040, 2022. 2, 3

[3] Pablo Arbelaez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir D. Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3378–3385. IEEE Computer Society, 2012. 1

[4] Aditya Arun, C. V. Jawahar, and M. Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9432–9441. Computer Vision Foundation / IEEE, 2019. 1

[5] Abhijit Bendale and Terrance E. Boult. Towards open world recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1893–1902. IEEE Computer Society, 2015. 1

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6154–6162. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 2, 13

[7] 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. 5, 14

[8] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1201–1210. IEEE Computer Society, 2015. 1

[9] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3992–4000. IEEE Computer Society, 2015. 1

[10] Akshay Raj Dhamija, Manuel Günther, Jonathan Ventura, and Terrance E. Boult. The overlooked elephant of object detection: Open set. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1010–1019. IEEE, 2020. 1, 2, 3

[11] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. VOS: learning what you don't know by virtual outlier synthesis. *CoRR*, abs/2202.01197, 2022. 1, 2

[12] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 1, 6, 14

[13] 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, CVPR 2021, virtual, June 19-25, 2021*, pages 4527–4536. Computer Vision Foundation / IEEE, 2021. 2, 4

[14] Stanley C. Fralick. Learning to recognize patterns without a teacher. *IEEE Trans. Inf. Theory*, 13(1):57–64, 1967. 3

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016. 2

[16] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1440–1448. IEEE Computer Society, 2015. 1, 13

[17] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587. IEEE Computer Society, 2014. 1

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. 4

[19] Akshita Gupta, Sanath Narayan, K. J. Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. OW-DETR: open-world detection transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 9225–9234. IEEE, 2022. 3, 22

[20] Jiaming Han, Yuqiang Ren, Jian Ding, Xingjia Pan, Ke Yan, and Gui-Song Xia. Expanding low-density latent regions for open-set object detection. *CoRR*, abs/2203.14911, 2022. 2, 3

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society, 2017. 1, 2, 5, 13

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 5

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1

[24] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 1

[25] Jan Hendrik Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):814–830, 2016. 6

[26] Dong hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, 2013. 2, 3, 4

[27] Nathan Inkawhich, Jingyang Zhang, Eric K. Davis, Ryan Luley, and Yiran Chen. Improving out-of-distribution detection by learning from the deployment environment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 15:2070–2086, 2022. 1

[28] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pages 816–832. Springer, 2018. 2, 4, 14, 21

[29] K. J. Joseph, Salman H. Khan, Fahad Shahbaz Khan, and Vineeth N. Balasubramanian. Towards open world object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5830–5840. Computer Vision Foundation / IEEE, 2021. 1, 2, 3, 4, 5, 13, 22

[30] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Label, verify, correct: A simple few shot object detection method. *CoRR*, abs/2112.05749, 2021. 3

[31] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[32] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *CoRR*, abs/2108.06753, 2021. 1, 2, 4, 5, 6, 13, 21

[33] Sachin Konan, Kevin J. Liang, and Li Yin. Extending one-stage detection with open-world proposals. *CoRR*, abs/2201.02302, 2022. 1, 2, 4, 5, 6, 21

[34] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 725–739. Springer, 2014. 2, 13, 14

[35] Durga Kumar and Xiaoling Zhang. Ship detection based on faster R-CNN in SAR imagery by anchor box optimization. In *2019 International Conference on Control, Automation and Information Sciences (ICCAIS), Chengdu, China, October 23-26, 2019*, pages 1–6. IEEE, 2019. 8

[36] Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. Deepbox: Learning objectness with convolutional networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2479–2487. IEEE Computer Society, 2015. 1, 2

[37] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised object discovery and tracking in video collections. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3173–3181. IEEE Computer Society, 2015. 1

[38] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss V2: learning reliable localization quality estimation for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11632–11641. Computer Vision Foundation / IEEE, 2021. 5

[39] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 5

[40] Yiding Li, Shunsheng Zhang, and Wen-Qin Wang. A lightweight faster R-CNN for ship detection in SAR images. *IEEE Geosci. Remote. Sens. Lett.*, 19:1–5, 2022. 8

[41] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944. IEEE Computer Society, 2017. 2, 5

[42] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017. 5

[43] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 1, 5

[44] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[45] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 2348–2354. IEEE, 2019. 2

[46] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1–7. IEEE, 2018. 2

[47] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *CoRR*, abs/2006.05278, 2020. 3, 4

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[49] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1990–1998, 2015. 14

[50] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(1):128–140, 2017. 1, 2, 13, 14

[51] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 1, 2, 4, 5, 13

[52] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA*, pages 29–36. IEEE Computer Society, 2005. 3

[53] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1939–1946. IEEE Computer Society, 2013. 1

[54] Kuniaki Saito, Ping Hu, Trevor Darrell, and Kate Saenko. Learning to detect every thing in an open world. *CoRR*, abs/2112.01698, 2021. 1, 2, 5, 6, 21

[55] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1757–1772, 2013. 1, 21

[56] Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability models for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2317–2324, 2014. 1

[57] H. J. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory*, 11(3):363–371, 1965. 3

[58] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *CoRR*, abs/2005.04757, 2020. 2, 3, 4

[59] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9626–9635. IEEE, 2019. 2, 4, 5, 14, 21

[60] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *Int. J. Comput. Vis.*, 104(2):154–171, 2013. 1, 2, 6, 13, 14

[61] Thang Vu, Hyunjun Jang, Trung X. Pham, and Chang Dong Yoo. Cascade RPN: delving into high-quality region proposal network with adaptive convolution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1430–1440, 2019. 2, 5

[62] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2965–2974. Computer Vision Foundation / IEEE, 2019. 2, 5

[63] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *CoRR*, abs/1501.04587, 2015. 1

[64] Rui Wang, Dhruv Mahajan, and Vignesh Ramanathan. What leads to generalization of object proposals? In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12536 of *Lecture Notes in Computer Science*, pages 464–478. Springer, 2020. 6, 13

[65] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan L. Yuille, and Fan Yang. Crest: A class-rebalancing self-training

framework for imbalanced semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10857–10866. Computer Vision Foundation / IEEE, 2021. 3

[66] Zhiheng Wu, Yue Lu, Xingyu Chen, Zhengxing Wu, Liwen Kang, and Junzhi Yu. UC-OWOD: unknown-classified open world object detection. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part X*, volume 13670 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2022. 3, 22

[67] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. Computer Vision Foundation / IEEE, 2020. 3

[68] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *CoRR*, abs/1905.00546, 2019. 3

[69] Qize Yang, Xihan Wei, Biao Wang, Xian-Sheng Hua, and Lei Zhang. Interactive self-training with mean teachers for semi-supervised object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5941–5950. Computer Vision Foundation / IEEE, 2021. 3, 4

[70] Jinan Yu, Liyan Ma, Zhenglin Li, Yan Peng, and Shaorong Xie. Open-world object detection via discriminative class prototype learning. In *2022 IEEE International Conference on Image Processing, ICIP 2022, Bordeaux, France, 16-19 October 2022*, pages 626–630. IEEE, 2022. 3, 22

[71] Jingyang Zhang, Nathan Inkawhich, Yiran Chen, and Hai Li. Fine-grained out-of-distribution detection with mixup outlier exposure. *CoRR*, abs/2106.03917, 2021. 1

[72] Shaoming Zhang, Ruize Wu, Kunyuan Xu, Jianmei Wang, and Weiwei Sun. R-cnn-based ship detection from high resolution remote sensing imagery. *Remote. Sens.*, 11(6):631, 2019. 8

[73] Zhengning Zhang, Lin Zhang, Yue Wang, Pengming Feng, and Ran He. Shiprsimagenet: A large-scale fine-grained dataset for ship detection in high-resolution optical remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 14:8458–8472, 2021. 7, 8

[74] Xiaowei Zhao, Xianglong Liu, Yifan Shen, Yixuan Qiao, Yuqing Ma, and Duorui Wang. Revisiting open world object detection. *CoRR*, abs/2201.00471, 2022. 3, 22

[75] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 391–405. Springer, 2014. 1, 2, 6, 13, 14

[76] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3