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 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.

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 discriminative 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 \( \lambda_{CLS} \), an aggressive OOD object detector using a small \( \lambda_{CLS} \), 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 \( \sim 3x \) 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...
The open-set object proposal task is to train a model to generate candidate regions for unlabelled data. Generally, in an object detection task, we have a set of ID and OOD 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 $\theta$ 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 $\mathcal{M}(X; \theta) = \{[x, y, w, h, s]_{j=1..N}\}$, where $x, y, w, h$ 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 fully 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 hyperparameter, 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.

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 $\mathcal{M}$ parametrized by $\theta$ 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 $\mathcal{M}(X; \theta) = \{[x, y, w, h, s]_{j=1..N}\}$, where $x, y, w, h$ 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 fully 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 hyperparameter, 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 \((r \times E)\) 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 \((E + r \times (E/4))\) 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 \(\rho\%\) of the number of original training instances (\(\rho\) 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:

\[
L_{THPN}(\{c_i\}, \{q_j\}, \{t_i\}) = \lambda_{CLS} \frac{1}{N_{CLS}} \sum_i L_{CE}(c_i, c_i^*) + (1 - \lambda_{CLS}) \frac{1}{N_{LQ}} \sum_j L_{LQF}(q_j, q_j^*) + \lambda_{BOX} \frac{1}{N_{BOX}} \sum_i c_i^* L_{WBBD}(t_i, t_i^*).
\]

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, \(c_i, q_j, t_i\) 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 $\lambda_{CLS}$ hyperparameter. By adjusting $\lambda_{CLS} \in [0, 1]$, the user can significantly alter the behavior of the resulting model. The smaller $\lambda_{CLS}$ 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 $s = \lambda_{CLS} \cdot cls\_scores + (1 - \lambda_{CLS}) \cdot lq\_scores$.

**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 to difficult pseudo-label targets. Particularly, the loss of the $i^{th}$ sampled anchor is:

$$L_{LQF}(q_i, q^*_i) = |q^*_i - q_i|^\gamma L_{BCE}(q_i, q^*_i)$$

where $q_i$ and $q^*_i$ are the predicted and target localization quality for the given anchor, respectively. $\gamma$ is a hyperparameter to scale the significance of the weighting (we use $\gamma=2$). 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.

$$L_{WBR}(t_i, t^*_i) = s_i^\beta L_1(t_i, t^*_i)$$

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 $\beta$ scales how severely we downweight the loss from anchors matched to lower scoring targets (we use $\beta=2$). 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

<table>
<thead>
<tr>
<th>Split</th>
<th>Method</th>
<th>OOD</th>
<th>AR10</th>
<th>AR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPN [51]</td>
<td></td>
<td>7.4</td>
<td>20.0</td>
<td></td>
</tr>
<tr>
<td>GA-RPN [62]</td>
<td></td>
<td>11.9</td>
<td>27.7</td>
<td></td>
</tr>
<tr>
<td>Cascade RPN [61]</td>
<td></td>
<td>12.6</td>
<td>27.7</td>
<td></td>
</tr>
<tr>
<td>Faster R-CNN [51]</td>
<td></td>
<td>11.6</td>
<td>25.1</td>
<td></td>
</tr>
<tr>
<td>VOC</td>
<td>FCOS [50]</td>
<td>10.5</td>
<td>24.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FCOS-OWP [33]</td>
<td>14.5</td>
<td>31.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDE [54]</td>
<td>18.2</td>
<td>30.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLN [32]</td>
<td>18.4</td>
<td>33.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>THPN ($\lambda_{CLS} = 0$)</strong></td>
<td><strong>21.6</strong></td>
<td><strong>38.9</strong></td>
<td></td>
</tr>
</tbody>
</table>
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 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 +11.4% AR100 on Animal for 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 $\lambda_{CLS} \leq 0.25$ 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...
drastically improved overall recall in these cases. Both unlabeled OOD and labeled training datasets that only contain 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.

Table 3 contains the results of this challenge. We consider OOD recall is paramount, a small bias splits and OLN on high-bias splits. Finally, all THPN variants outperform both baselines in terms of ALL recall, but the choice of $\lambda_{CLS}$ can make a large difference. On COCO40, where 72% of total instances are from ID classes, users should choose a larger $\lambda_{CLS}$ 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 $\lambda_{CLS}$ 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.

Table 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 $\lambda_{CLS}$ 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 $\lambda_{CLS}$ 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 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 $\lambda_{CLS}$ 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 $\lambda_{CLS}$ 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.
pseudo-labels are of high quality in both domains, achieving. 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. Table 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 $L_{WB}$, 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


