A Example forgetting in object detection

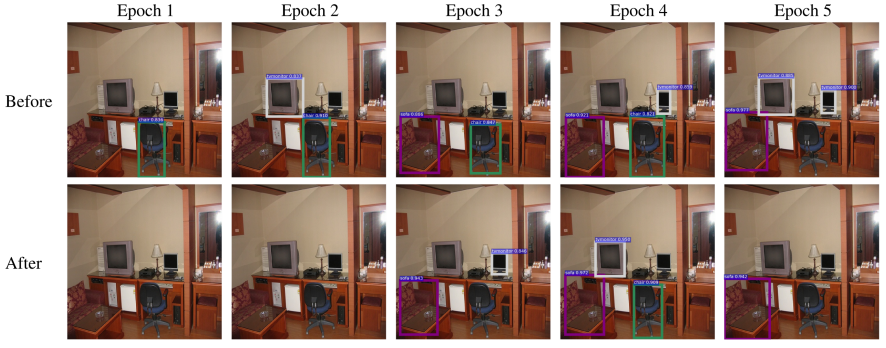


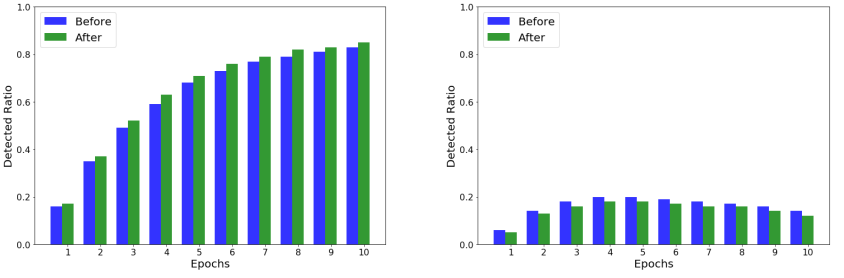
Figure 7: Detections on a PASCAL VOC train set image missing annotations throughout training: only the sofa in the lower left has a label. Each column shows detections directly before (top) and after (bottom) the model is trained on the image shown, for each epoch. While the sofa is consistently detected (purple box) after being learned, the unlabeled objects (2 monitors, a chair) are repeatedly found and then suppressed after being trained upon.

In a recent study of training dynamics of neural network classifiers, Toneva et al. [47] defined a "forgetting event" as a training example switching from being classified correctly by the model to being classified incorrectly during training. It was found that certain examples were forgotten more frequently than others while others were never forgotten (termed "unforgettable"), with the degree of forgetting for individual examples being consistent across neural network architectures and random seeds. When visualized, the forgotten examples tend to have atypical or uncommon characteristics (e.g., pose, lighting, angle), relative to "unforgettable" examples. Interestingly, a significant number of "unforgettable" examples could be removed from the training set with only a marginal reduction in test accuracy, if the "hard" examples were kept. This implies that the "hard" examples play a role akin to support vectors in max-margin learning, while easier "unforgettable" examples have little effect on the final decision boundary.

Within the context of object detection datasets, we hypothesize that unlabeled object instances form a similar group of hard examples that are also learned and then forgotten throughout training. Unlike the inter-batch catastrophic forgetting in [47], however, where hard examples are learned while part of the current minibatch and then forgotten while learning other examples, unlabeled samples in object detection are learned from other examples and then *suppressed* after incurring misclassification losses during training (see Figure 7).

Unlabeled instances strongly resemble positive examples throughout the rest of the dataset, but their lack of labels mean that the typical PN classification objective incentivizes learning them as negatives. Given that hard examples have a strong influence on classifier boundaries, having unlabeled examples trained as negatives may prove especially detrimental to training.

We perform a similar study as [47] and investigate forgetting events on PASCAL VOC [12] by tracking detection rates of labeled and unlabeled instances in the training set throughout learning. In particular, an object is considered detected if the detector produces a bounding box with intersection over union (IoU) of at least 0.5 and the classifier is at least 80% confident in the correct class. We track whether or not an object was detected directly before the image it belongs to is trained upon, and then again after the gradients have been applied.



(a) Labeled objects (b) Unlabeled objects

Figure 8: Detection rates of objects before and after training on their corresponding images for (a) labeled instances and (b) instances with labels withheld during training.

These indicator variables are then combined across objects for each epoch and reported as a percentage. While PASCAL VOC does naturally have unlabeled instances, we do not have access to these without a re-labeling effort. As such, we remove 10% of object annotations randomly across all object classes during training, and use them to calculate detection rates for this experiment.

Detection rates for labeled and unlabeled objects over time are shown in Figure 8. As expected, the model learns to detect a higher percentage of labeled instances over time, and objects are overall more likely to be detected immediately after the detector trains on them. Despite not having an explicit learning signal, unlabeled objects are still learned throughout training, but at a lower rate than labeled ones. In contrast with labeled objects, unlabeled object detections are discouraged with each PN gradient, leading to a dip in overall detection rates immediately after training. Despite this, overall detection rates of unlabeled objects grows through the first 5 epochs of training, implying a repeated cycle of learning unlabeled objects from other intra-class examples, forgetting them when explicitly trained against them, and then learning them again. Given the undesirability of this forced suppression of detected objects, we seek a method to remedy this behavior.

B Recall of Region Proposals

To investigate the effect of the PU risk estimator on the quality of the proposals from RPN stage, we examine the recall of the top 500 proposals, compared with the complete annotations. Higher recall means there are more proposals that match with the full-labeled annotations. In Figure 9, the recalls using the PU risk estimator are higher than those using the PN risk estimator. This illustrates the inclusion of more object proposals that are not included by the PN risk estimator because the corresponding ground truth annotations are missing.

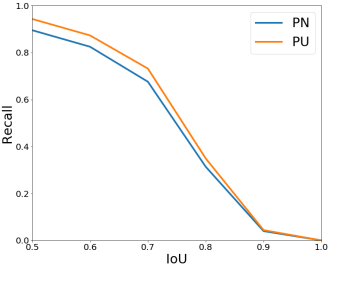


Figure 9: Recall of top 500 proposals from RPN after training on PASCAL VOC2007 when $\rho = 0.5$.

C OpenImages

OpenImages [23, 26] is a large object dataset consisting of 15.4 million bounding boxes from 600 classes across 1.9 million images. In order to achieve its scale, the labeling effort was crowd-sourced to a large number of human annotators. As pointed out in [12], even increasing from 10 classes of objects in PASCAL VOC2006 to the 20 in VOC2007 resulted in a substantially larger number of labeling errors, as it became more difficult for human annotators to remember all of the object classes. With 500 classes, this problem is worse by an order of magnitude for OpenImages. While the creators of OpenImages designed an annotator training process to insure quality, there still are many examples of missing labels. As such, PU learning as proposed is especially appropriate, even when considering full labels.

As in Section 4.2, we train a ResNet101 Faster R-CNN object detector with both PN and PU classification losses. Given the large size of the dataset, we restrict our analysis to 50 of the most prevalent classes, and subsample 140K images from the dataset containing at least one of the selected classes. Of these 140K images, we train on 100K with *full* annotations, and hold out 10K for validation and 30K as our test split. We observe that, all other things equal, switching to our proposed PU approach results in an increase of $+3.0$, $+3.0$, and $+5.0$ for mAPs with IoU thresholds $\{0.25, 0.50, 0.75\}$ (see Table 2).

Table 2: Detector performance on a subset of OpenImages at various IoU thresholds.

Method	AP_{25}	AP_{50}	AP_{75}
PN	37.7	33.6	20.7
PU	**40.7**	**36.6**	**25.7**