2nd Place Solution SSLAD Track 1 - O2O Semi-Supervised Framwork

Beitong Zhou*, Donghao Gu*, Binbin Zhang*, Sanli Tang, Yunlu Xu✉, Zhanzhan Cheng, Yi Niu
AML Group, GOLDWAY ITS CO. LTD
Shanghai, China
xuyunlu1030@163.com

Abstract

*This report introduces our solution for the ICCV 2021 Workshop SSLAD Track 1 - 2D Object Detection. The goal of Semi-Supervised Object Detection (SS-OD) is to train a detector which exploits large amounts of unlabeled data with only a few data labeled. Till now, related methods can be roughly divided into self-training based methods and consistency-regularization based methods. In this work, we combined the above mentioned methods and proposed a holistic framework named as O2O (**O**ffline to **O**nline) for SS-OD problem. We adopted Cascade R-CNN with the Swin-Transformer backbone as our detector for training on SODA10M dataset. In virtue of the proposed SS-OD method, our best single model could reach a mAP of 81.35. With ensembles, we further improved the final mAP in the public leaderboard to 81.79 and achieved the 2nd place in the SSLAD Track 1 challenge.*

1. Introduction

The SSLAD 2D Object Detection challenge in ICCV 2021 is a 2D object detection task for autonomous driving. A large-scale dataset called SODA10M [5] with 10 million unlabeled images in total is provided. For labeled set, SODA10M annotates 5K images for training, 5K images for validation and 10K images for testing. The main task of this track is to exploit the massive unlabeled data together with labeled images to improve the performance and generalize on the test set. A straightforward way to address this challenge is to adopt Semi-Supervised Object Detection (SS-OD) methods.

While semi-supervised learning has been widely explored in image classification tasks [11, 13], few works have been focused on object detection due the difficulty of localizing and regressing the location of each object on images. Recent SS-OD methods can be roughly divided into self-training based methods and consistency-regularization based methods. The self-training based methods [12] generate pseudo labels from a pretrained model and then train the detector jointly with unlabeled and labeled data. We name those methods with a self-training pattern as Offline methods, as the pseudo labels are fixed during the semi-supervised training process. On the other head, consistency-regularization based methods [8, 15] dynamically generate pseudo labels or regularize the consistency of outputs with different data transformations. Most of them are in a Teacher-Student pattern with the teacher model updated via a Exponential Moving Average (EMA) of student weights. Those methods are named as Online methods as the predictions vary through iterations.

The different training patterns endue those methods with different characteristics. We argue that while pseudo labels generated by offline methods can be of high quality, the gain margin is limited with fixed predictions. For online methods, the model outputs gradually evolve through time if trained properly although the performance can be inferior in the early training stage. It's validated that self-training based methods showed inferior performance than consistency-regularization based methods and the performance couldn't improve with increased amount of unlabeled images in experiments [5]. Therefore, we proposed a holistic SS-OD framework called O2O (**O**ffline to **O**nline) to combine the advantages of above mentioned methods. We train our model in following steps: 1) train a baseline detector on labeled images, 2) first utilize the fixed pseudo labels generated on unlabeled images by the baseline detector to train the student detector, and then 3) switch to Teacher-Student training pattern after a period of iterations. Also, we can repeat the process a few times to further improve the performance as the work [14].

As in a 2D object detection task, it's equally important to design a strong detector for training on limited labeled images so that it can generalize well on unseen unlabeled data. We adopted Cascade R-CNN [2] with Swin Transformer [9] as backbones. Swin Transformer recently refreshes the SOTA results of multiple computer vision tasks including object detection and enables us to have a strong detection

*These authors contribute equally to this work.

baseline as well as generate high-quality pseudo labels. To summarize our whole solution:

- **Strong Detector**: We adopted Cascade R-CNN with Swin Transformer backbones, which provides a strong baseline and generalizes well on unseen images.
- **O2O Framework**: We proposed a SS-OD framework named as O2O combining the advantages of self-training based methods as well as consistency-regularization based methods to improve the detection performance.
- **Auto Ensemble**: We adopted a search-based ensemble mechanism to aggregate predictions by different detectors automatically.

2. 2D Object Detection

We use two-stage Cascade R-CNN [2] as our detector and adopted Swin Transformer [9] as the backbone feature extractor. The task of this challenge it to exploit massive unlabeled data with limited labeled images and we assume that the diversity of model architectures won't contribute much to achieving an impressive performance.

Basic Architecture. The detector architecture we use is actually the same as that of the original paper. It should be noticed that when training on only labeled images to get a baseline detector, we add auxiliary mask heads like HTC [3] did to help improve the detection performance by learning hybrid tasks. The feeded masks are forged by simply filling pixels within annotated bounding boxes and this design gives us a 0.75% boost in mAP. In semi-supervised training, the mask heads are removed to reduce GPU memory usage. Other improvements such as replacing L1 loss with GIoU loss [10] also works well in our training.

Data Augmentation. With limited labeled images and complex driving scenes, overfitting is introduced unavoidably in this task. Except for standard augmentation strategies such as multi-scale training and flip, we also use more radical augmentations like color jitters and MixUp [16]. Those augmentations are helpful especially for adjusting to lighting changes in different scenes and detecting blocked objects. Other techniques like Copy-Paste [4] and Mosaic augmentations [1] couldn't lead to stable improvements.

TTA and Multi-class NMS. We empirically adopted standard test-time augmentation strategies, including multi-scales and image flip. The chosen scale ratios are $1.0, 1.1, 1.2$, and thus the TTA results are the average of 6 predictions in total. In the post-processing stage of object detection, NMS or soft-NMS is commonly used to filter invalid bounding boxes. We've found in our experiments that a direct use of soft-NMS increases the final mAP by a minor improvement. The performance of pedestrian, cyclist and tricycle actually degrades with the use of soft-NMS while the mAP of remaining threes increase. After analyzing, we assume that soft-NMS could retrieve more miss detections for categories with higher IoU threshold, which is 0.7 for car, truck and tram. While soft-NMS would lead to more false detections for categories with 0.5 IoU threshold, we merge the post-processing results of soft-NMS and NMS. This gives us a $0.3 \sim 0.4\%$ boost in mAP.

3. O2O SS-OD Framework

Our SS-OD framework are shown in Figures 1 and we've divided the whole training process into three different parts.

3.1. Stage-0: Baseline Training

This stage refers to training on labeled images to offer a good initialization for semi-supervised training as well as high-quality offline pseudo labels. In this stage, we add auxiliary mask heads to help learn latent knowledge and remove those heads in all other stages. For labeled data $\boldsymbol{D}_s = \{\boldsymbol{x}_i^s, \boldsymbol{y}_i^s\}_{i=1}^{N_s}$, we follow standard training schemes and the supervised loss consists of three parts:

$$\mathcal{L} = \sum_i \mathcal{L}_{rpn}(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s) + \mathcal{L}_{roi}(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s) + \mathcal{L}_{mask}(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s) \quad (1)$$

The loss $\mathcal{L}_{rpn}$ and $\mathcal{L}_{roi}$ both include regression loss and classification loss. The trained weights θ would be loaded for student model θ_s and teacher model θ_t in the next stage training.

3.2. Stage-1: O2O Semi-Supervised Training

To exploit large amounts of unlabeled images, we proposed O2O semi-supervised framework to combine offline and online methods. Offline pseudo labels generated by baseline detector are first used to stabilize the model performance. As the training processes, the model could no longer learn useful knowledge from fixed pseudo labels and then we switch to a teacher-student pattern to continue training with the help of online pseudo labels.

Data Selection. Considering 10 million unlabeled images are available, we've tried different strategies to sample unlabeled data with a fixed amount to maximize the semi-supervised performance, such as sample uniformly from different annotated tags, but found out that random sampling is good enough for training. With collected data, we sample the labeled and unlabeled images separately to balance the ratio of those two in a single batch so that the training performance won't be affected much by the errors in pseudo labels. Especially, the sampling ratio $1:1$ is used, which means that the labeled set would be iterated much more times than unlabeled set.

Offline Pseudo Labels. To get high-quality pseudo labels, we use TTA strategy mentioned in section 2 to boost the accuracy. In order to prevent the detrimental effects introduced by noisy pseudo labels, it's critical to set a proper confidence threshold σ to filter false detections with low

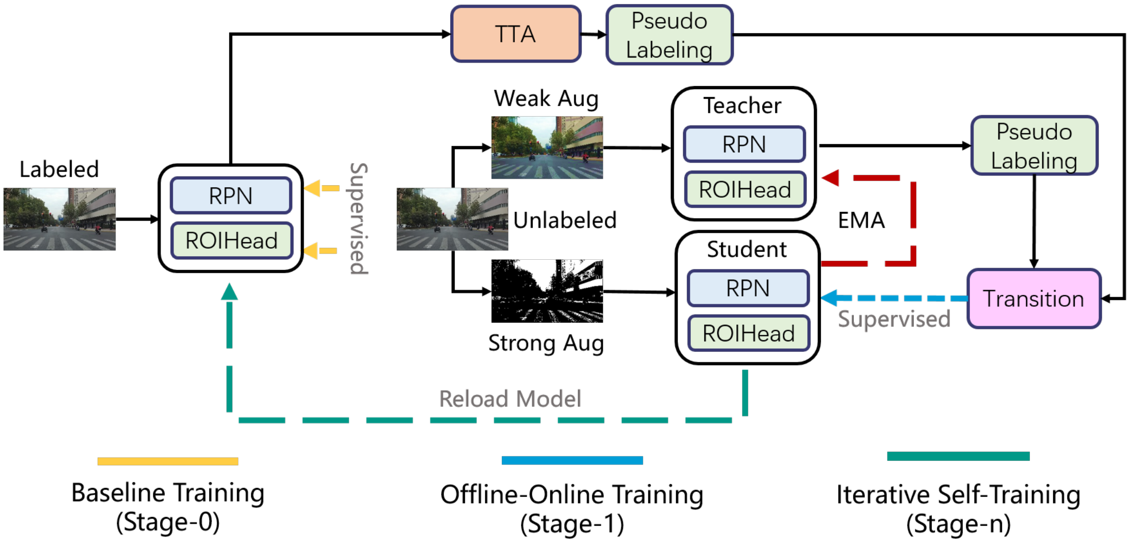

Figure 1. O2O Semi-Supervised Object Detection framework.

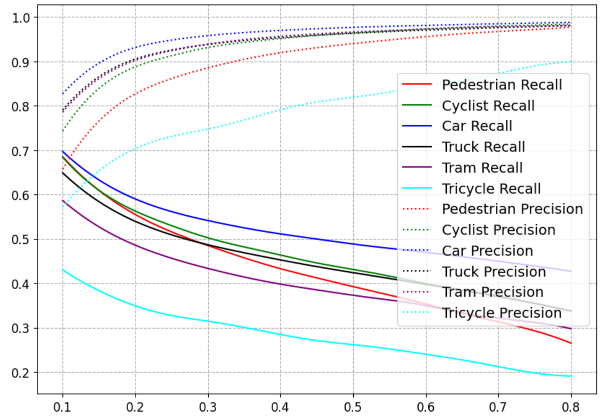

Figure 2. Recalls and precisions for each category under different confidence thresholds.

confidence, while keeping sufficient positive detections. We analyze the recalls and precisions for categories under different score thresholds as shown in Figure 2. The tendency of recalls and precisions become steady with the increase of the confidence threshold and we simply choose 0.5 as threshold for offline pseudo labels as we find that further tuning the parameters only brings negligible improvements.

Online Pseudo Labels. When generating online pseudo labels, we adopt teacher-student pattern where the teacher model is updated by exponential moving average of the student model. The slowly updated teacher model θ_t ensures stabler and better detection performance than the student model θ_s.

$$\theta_t = \alpha\theta_t + (1-\alpha)\theta_s \tag{2}$$

As the false detections would easily be amplified through EMA updates, we set a higher threshold of 0.7 for filtering online pseudo labels in order to relieve the confirmation bias or error accumulation problem.

Weak-Strong Augmentation. It is common to use the weak-strong augmentation scheme in semi-supervised classificatoin tasks [11]. In our proposed framework, we randomly apply weak and strong augmentations A_{rand} on labeled images considering that the labeled data should stabilize the training process as well as avoid overfitting. For unlabeled data, the teacher model is fed with weakly augmented images to generate pseudo labels and the student model is trained with strong augmentations A_s for regularization purpose.

The final semi-supervised loss is the sum of labeled loss and unlabeled loss. For unlabeled data $\boldsymbol{D}_u = \{\boldsymbol{x}_i^u\}_{i=1}^{N_u}$, the pseudo labels $\hat{\boldsymbol{y}}_i^u$ would switch from fixed predictions generated offline by baseline detector and to online predictions made by the EMA teacher model after a period of iterations.

$$\mathcal{L}_s = \sum_i \mathcal{L}_{rpn}(A_{rand}(\boldsymbol{x}_i^s), \boldsymbol{y}_i^s) + \mathcal{L}_{roi}(A_{rand}(\boldsymbol{x}_i^s), \boldsymbol{y}_i^s) \tag{3}$$

$$\mathcal{L}_u = \sum_i \mathcal{L}_{rpn}(A_s(\boldsymbol{x}_i^u), \hat{\boldsymbol{y}}_i^u) + \mathcal{L}_{roi}(A_s(\boldsymbol{x}_i^u), \hat{\boldsymbol{y}}_i^u) \tag{4}$$

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u \tag{5}$$

3.3. Stage-n: Iterative Self-Training

After the last stage of semi-supervised training, we would have a much better model than the baseline detector. Thus, it becomes possible to further improve the performance by putting back the trained semi-supervised model as the baseline detector and repeat the above process for more rounds. It should be noticed that after a round of semi-supervised training, the model becomes much more confident on unlabeled images and generate high-confidence predictions. This can by risky as the predefined threshold could no longer work especially for teacher-student training. As a result, we simply use the fixed pseudo labels generated by the trained model from the last stage instead of online predictions for fine-tuning.

4. Auto Ensemble

We design a two-stage auto ensemble scheme, in which the proposals of all models are fused together and feed to ROI Heads respectively to produce final results. Considering the computational consumption required by semi-supervised training, we choose only 4 models differing slightly in training schedules and unlabeled data for ensemble. We follow the same TTA strategy as mentioned above, and thus this leads to totally 24 predictions.

RPN Heads Ensemble. All predictions are simply concatenated together and the scores are lowered by soft-NMS while keeping as many as positive detections as possible. For each, we retain top 1000 bboxes with highest scores and feed them to the following ROI heads.

ROI Heads Ensemble. The predictions of each fed proposal in ROI heads would be weighted average into one single result. For every model, the weights of each category are first searched to find the best matches, which maximize the detection performance on the given validation set [7]. The ensemble process is presented as:

$$(x,y) = \frac{w_1 \cdot (x_1,y_1) + w_2 \cdot (x_2,y_2) + \cdots + w_n \cdot (x_n,y_n)}{w_1 + w_2 + \cdots + w_n} \tag{6}$$

$$s = \left(\frac{w_1 \cdot s_1 + w_2 \cdot s_2 + \cdots + w_n \cdot s_n}{w_1 + w_2 + w_n}\right)^p \tag{7}$$

where (x, y) and s indicate the coordinates and confidence of the ensemble bounding box. The w_i is the average weight and $p \in (0, 1]$ is used to control the confidence distribution, which helps retrieve more positive detections especially combined with soft-NMS. First, we would search for post-processing method and IoU thresholds on validation set. And then the simulated annealing method [6] is used to search for parameters w and p.

5. Experiments

5.1. Implementation Details

Baseline Training. The baseline experiment is conducted on 8 NVIDIA V100 GPUs with a batch size of 16. In the training phase, we utilize multi-scale training and the image size ranges from 960 × 640 to 2880 × 1920. The SGD optimizer with initial learning rate 0.03, momentum 0.9 and weight decay 0.0001 is used for training. The augmentation strategy includes mixup, cutmix and random color jitters. The training epoch is set to 50 with the learning rate decayed by a factor of 0.1 at epoch 33 and 44.

	validation mAP
Basline (Cascade R-CNN)	67.32
+ color jitters	67.81 (+0.49)
+ MixUp / CutMix	68.75 (+0.94)
+ 50 epochs training	69.48 (+0.73)
+ multi-scale training	70.83 (+1.35)
+ HTC mask heads	71.47 (+0.64)

Table 1. Performance of techniques of a single model with single scale testing on the validation set trained on training set.

The Cascade R-CNN detector uses Swin Transformer as backbone with window size 12 × 12, drop path rate 0.3 and ImageNet pretrained weights for initialization. The baseline bbox head uses GIoU loss for regression and cross entropy loss for classification. We keep the mask heads of HTC in baseline training and remove them in the following semi-supervised training for saving GPU memories.

Semi-Supervised Training. In Stage-1, we load baseline weights to initialize the student model and teacher model. The offline pseudo labels are generated by the baseline detector and the confidence threshold is set to 0.5 to filter low-confidence false detections. In teacher-student training, the EMA update ratio α is set to 0.999 and a confidence threshold 0.7 is used for online pseudo labels. We randomly sample 200k unlabeled images for training and the epoch is set to 15. The Considering the large number of unlabeled images, we conduct experiment on 32 NVIDIA V100 GPUs and the batch size is chosen as 64. The pseudo labels switch from offline to online at epoch 5 and the learning rate decays by a factor of 0.1 at epoch 7 and 12.

In Stage-n, we use the semi-supervised model trained in last stage to update offline pseudo labels on the same unlabeled images and load it to finetune for one more round. Online pseudo labels are not used here and the initial learning rate is set to be 0.003.

5.2. Results

We first study the effectiveness of different techniques on training the baseline detector and the results are shown in Table 1. It is easy to see that augmentation strategies to avoid overfitting are necessary for training a good baseline detector as the training set only consists of 5000 labeled images.

In Table 2, we validate the performance of different parts in O2O semi-supervised framework. For quick experiments, we use the 5k labeled images and randomly sampled 50k unlabeled images for training. The sharp improvement on validation set should be attributed partly to the different data distributions of training set and validation set. For example, there are no night scenes in the training set and the

	Offline PL	Weak-Strong Aug	Online PL	Iterative Training	validation mAP
Baseline					71.47
	✓				73.94 (+2.47)
	✓	✓			74.85 (+3.38)
	✓	✓	✓		76.15 (+4.68)
	✓	✓	✓	✓	76.67 (+5.20)

Table 2. Performance of O2O semi-supervised framework with single scale testing on the validation set. For quick experiments, the sampled amount of unlabeled images is 50k.

	Car	Truck	Tram	Pedestrian	Cyclist	Tricycle	mAP
Basline (train+val)	89.90	82.50	78.40	77.20	83.00	44.20	75.87
Best Model	92.40	86.90	83.70	82.10	86.90	56.10	81.35
Ensemble	92.89	87.49	84.55	82.51	87.25	56.08	81.79

Table 3. Model Performance on the test set. The results of every category are shown in the table.

predicted pseudo labels help learn the latent knowledge to detect on night images. When adding the validation set to the labeled data, the improvement becomes less remarkable on the test set. For iterative training, we only train the semi-supervised model for one more round, which means that this is the performance of the Stage-2. As the marginal benefits introduced by better semi-supervised models decrease, the performance of further training stabilizes. Training for a one more stage is enough with limited computational resources.

We show our results on the test set in Table 3. The baseline in trained together with the labeled training set and validation set. The 200k sampled unlabeled images are used for semi-supervised training and we've found that increase the sample number to 500k couldn't improve the performance much. We assume that the extra sampled images become homogenous and carry less useful information. Our best single model is achieved after the Stage-2 training with TTA and post-processing. Finally, we ensemble 4 models with different unlabeled images and training schedules. The designed auto ensemble scheme gives us another 0.44% boost in the test mAP.

5.3. Things Don't Work

Except for semi-supervised training, we've also tried other standard detection strategies to complement our predictions, such as expert models and small object detections. For expert models, different resampling methods are used to improve the detection ability on the tricycle category but none of them are effective enough. We hypothesise that with only a small amount of labeled images, it is hard to train an expert model using only resampling methods to cover the whole feature space of a certain category, as the backgrounds and objects are varied. We also trained another model only for detecting small objects in the image but found out that much more false detections were introduced and the whole performance degraded.

6. Conclusion

In this report, we present our method for the semi-supervised object detection task. The proposed O2O framework which combines offline and online methods shows its effectiveness in our experiments. Also, we build a strong baseline detector using augmentation strategies and design a useful auto ensemble scheme. The above works led us to the 2nd place in the SSLAD Track 1 Challenge.

References

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

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

[3] Kai Chen, Wanli Ouyang, Chen Change Loy, Dahua Lin, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, and Jianping Shi. Hybrid task cascade for instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4974–4983, 2019.

[4] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[5] Jianhua Han, Xiwen Liang, Hang Xu, Kai Chen, Lanqing Hong, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Xiaodan Liang, and Chunjing Xu. Soda10m: Towards large-scale object detection benchmark for autonomous driving. *arXiv preprint arXiv:2106.11118*, 2021.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. *Optimization by simulated annealing*, volume 220. 1987.

[7] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place solutions for openimage2019 – object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2020.

[8] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[10] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.

[11] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608, 2020.

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

[13] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR (Workshop)*, 2017.

[14] Qizhe Xie, Minh-Thang Luong, Eduard 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)*, pages 10687–10698, 2020.

[15] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *arXiv: Computer Vision and Pattern Recognition*, 2021.

[16] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2017.