Points as Queries: Weakly Semi-supervised Object Detection by Points

Liangyu Chen[1,2] * Tong Yang[1] * Xiangyu Zhang[1] Wei Zhang[2] † Jian Sun[1]
[1] MEGVII Technology [2] Fudan University
{chenliangyu,yangtong,zhangxiangyu,sunjian}@megvii.com weizh@fudan.edu.cn

Abstract

We propose a novel point annotated setting for the weakly semi-supervised object detection task, in which the dataset comprises small fully annotated images and large weakly annotated images by points. It achieves a balance between tremendous annotation burden and detection performance. Based on this setting, we analyze existing detectors and find that these detectors have difficulty in fully exploiting the power of the annotated points. To solve this, we introduce a new detector, Point DETR, which extends DETR by adding a point encoder. Extensive experiments conducted on MS-COCO dataset in various data settings show the effectiveness of our method. In particular, when using 20% fully labeled data from COCO, our detector achieves a promising performance, 33.3 AP, which outperforms a strong baseline (FCOS) by 2.0 AP, and we demonstrate the point annotations bring over 10 points in various AR metrics.

1. Introduction

Object detection is one of the fundamental problems in computer vision. Modern object detectors [12, 14, 15, 22, 29] have achieved great success with the help of tremendous annotated data. However, it is very costly to annotate a large amount of detection data. Specifically, for each object instance, a precise bounding box needs to be labeled manually and carefully, which is quite time-consuming: it takes 10-35 seconds [27, 24, 1] for labeling an object.

To reduce the cost of data annotation, weakly supervised object detection (WSOD) and semi-supervised object detection (SSOD) methods are proposed. Weakly supervised object detection methods [2, 11, 25, 36] utilize large data with weak annotations, such as image labels, which is far easier to collect than precisely annotated bounding boxes. The semi-supervised object detection methods [10, 17, 26, 28, 31] learn detectors with a small amount of box-level labeled images and large unlabeled images, where the cost of image annotation is small. Although these methods can reduce the cost of annotation significantly, their performance is far inferior to their supervised counterparts [14, 15, 29]. To make a trade-off between annotation cost and performance, weakly semi-supervised object detection methods (WSSOD) [33] are studied, which use small box-level labeled images as well as large weakly labeled images to learn detectors. However, image-level annotations in weakly annotated data are not optimal for object detection task since image labels do not contain the instance-level information of all objects. Motivated by [1], we annotate each instance in the image by one point (as shown in Figure 1d) instead of image-level annotation, for two main reasons. Firstly, compared with image-level annotation, points bring much richer information, not only the category of the object but also the strong prior of object location. Secondly, there is no strict requirement on point annotations, such as center points of objects. Thus, the increase in the cost of labeling is marginal compared with the image-level annotation [1]: 23.3 sec/image vs. 20.0 sec/image in VOC dataset [6].

Though the above new setting is better for weakly semi-supervised object detection, most recent detectors [14, 15, 29] have difficulty in predicting object boxes based on point annotations. In most detectors, FPN [14] is a basic component, which utilizes multi-level feature maps to predict object boxes. FPN can boost the performance of detectors, but it is incompetent to predict object boxes using point annotations since it is difficult to select the optimal box prediction from multi-level ones, predicted for a point annotation. For the single-level feature detectors, they may suffer from bad performance [20, 21, 22] or strict requirement on point annotations [5, 12, 35] even though they avoid choosing feature map levels.

Inspired by DETR [4] which achieves competitive performance with a single-level feature map, we propose a novel detector, Point DETR, by adding a point encoder to DETR in this paper. It can predict object boxes precisely from point annotations. Specifically, it uses a single-level feature map to predict object boxes, avoiding the multi-level selection problem and can predict object boxes with loose

*Equally contribution.
†Corresponding author.

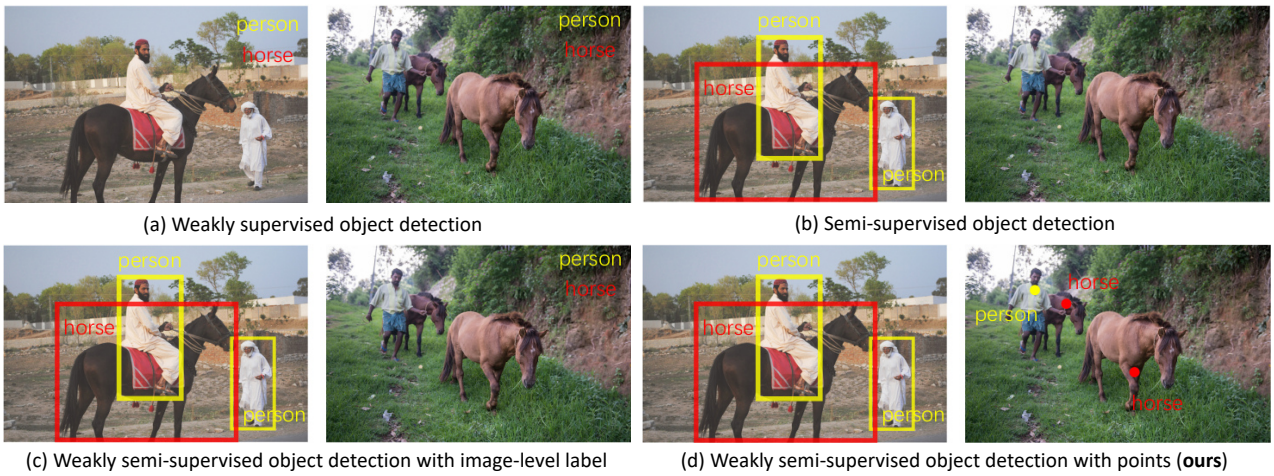


Figure 1. Different types of object detection settings to reduce the cost of data annotation

points, having no strict requirement on point annotations. Besides, it inherits the strong representation of DETR, having a good performance on object detection. But, different from DETR, we encode position and category of annotated points into object queries with the point encoder, which easily establishes one-to-one correspondences between points and object queries, being fit for box prediction based on points. In addition, to boost detection performance and make optimization easier, we do box predictions as offsets w.r.t. point position rather than make box predictions directly like DETR.

To show the superiority of our detector, we mainly evaluate our proposed detector on the MS-COCO dataset [16]. To make a fair comparison, we take FCOS [29] as the default baseline, which is regarded as a point-based detector. Following our proposed weakly semi-supervised object detection setting, object instances of small image data fraction ($5\% \sim 50\%$) are annotated fully and the rest are annotated by points. In these various settings with a different fraction of fully-annotated image data, our proposed detector outperforms other modern detectors, including multi-level feature detectors and single-level feature detectors. In particular, when using 20% fully labeled data from COCO, our detector outperforms FCOS and Faster R-CNN by 2.0 AP and 1.9 AP, respectively.

Our main contributions can be summarized as follows:

- We propose a potential and novel setting for the weakly semi-supervised object detection task, which comprises small fully annotated images and large weakly annotated images by points. Compared with the image-level data setting [1], this setting introduces weakly instance-level information with marginal annotation cost, which is fit for object detection. This provides a new perspective to improve detection performance with weakly annotated detection images.

- Based on the above setting, we analyze the drawbacks of existing modern object detectors and propose Point DETR, which is simple and easily implemented. The proposed detector takes object points as input, transforms these points into object queries, and predicts object box precisely for these queries, as shown in Figure 3.

- Extensive experiments on COCO dataset [16] are conducted to demonstrate the effectiveness of our proposed detector. Our detector outperforms most modern detectors in various data settings. We also do quantity and quality experiments to show our detector solves the problems suffered by most modern detectors.

2. Related Work

Supervised Object Detection: With the large-scale fully annotated detection data, existing modern detectors [12, 14, 15, 22, 29] have obtained great improvements in the object detection task. These detectors can be divided into two categories: two-stage detectors and one-stage detectors. FPN [14] is a popular two-stage detector, which predicts object proposals firstly and refines these proposals finally. Unlike two-stage detectors, one-stage detectors [12, 15, 29] directly outputs the classification and location of each object without refinement. Though achieving great success, these detectors are trained with a large amount of fully-annotated data, which is costly to annotate. Thus, there are many works proposed to reduce the annotation cost.

Semi-Supervised/Weakly Supervised Object Detection: Semi-supervised object detection (SSOD) [10, 17, 26, 28, 31] and weakly-supervised object detection (WSOD) [2, 11, 25, 36] are introduced to reduce the large cost of data annotation. The semi-supervised object detection methods learn detectors with a small amount of box-level labeled images and large unlabeled images. Jeong *et al*. [10] employ consistency constraints for object detection to exploit

unlabeled data. While, weakly supervised object detection methods utilize large data with weak annotations, such as image labels. Bilen *et al*. [2] learn an object detector under image-level supervision by combining region classification and selection. Furthermore, pursuing the performance of supervised detection and keeping the low cost of annotation, weakly semi-supervised object detection methods (WSSOD) [33] are studied, which use small box-level labeled images as well as large weakly labeled images to learn detectors. Unlike these semi-/weakly-supervised object detection, our proposed detector utilizes a new low-cost annotation: points, which provide instance location. Recently, UFO^2 [23] also uses point supervision as weak labels, but it does not explore the point information sufficiently as we shown in Section 4.3.

Point based Semi-Supervised Segmentation: Point supervision [1, 19, 34] has been employed by semantic segmentation. Bearman *et al*. [1] incorporate point supervision along with objectness prior to boost segmentation performance and alleviate annotation burden. Qian *et al*. [19] leverage semantic relationships among several labeled points to address the semantic scene parsing task. Different from these works, we focus on object detection task, where point-based detection has been explored little. Due to a lack of exploitation, existing detectors do not fit point-level annotation well.

DETR: Unlike existing detectors, DETR [4] removes the need for many hand-designed components like a non-maximum suppression procedure or anchor generation. By virtue of Transformer [30], DETR takes an image as input and directly outputs a fixed set of box predictions. For the point-based detection task, DETR has a beneficial characteristic: a single-level feature map, avoiding the multi-level selection problem. However, directly applied DETR into point-based detection task is not practical. Object queries in DETR are general embeddings and have no specific point information. Conversely, our detector encodes the position and category of annotated points into object queries with the point encoder and establishes one-to-one correspondences between point annotations and object queries.

3. Method

In this section, we first introduce the task of weakly semi-supervised object detection (WSSOD) with point annotations and discuss why existing object detectors can not fit this task well. Next, in order to solve it, we illustrate our novel detector, Point DETR, in detail.

WSSOD with point annotations: WSSOD generally uses a small set of instance-level labeled images and tremendous weakly image-level labeled images as training data (Figure 1c). However, for object detection, image-level labeled images do not fit WSSOD well, since it can not provide instance information. This raises a natural question: is there a new data annotation for weakly labeled images, which has instance information without a large annotation burden? In this paper, we introduce point annotation for weakly labeled images.

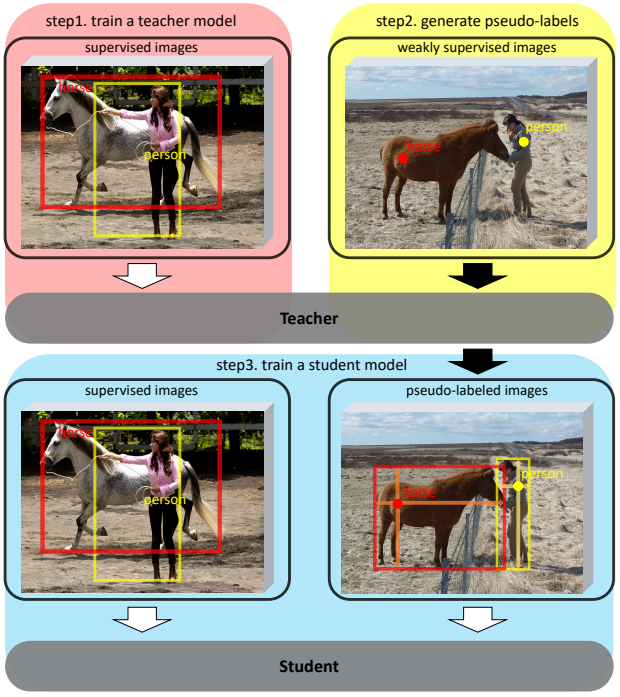


Figure 2. Overall framework. The white arrows represent the training stage, and the black arrows represent the inference stage. The steps of the framework are represented by red, yellow, blue rounded rectangles respectively. Best viewed in color.

Point Annotations: It is introduced by Bearman *et al*. [1] for weakly semantic segmentation, but it has not been explored well in object detection. In object detection, we define point annotation as follows: it locates on the object and takes object class as its category. Thus, we represent an object as (x, y, c), where $(x, y) \in [0, 1]^2$ and c represent point location and object category, respectively. We must note that our method is robust to point location, as shown in Table 1e. Therefore, the point annotations can locate at the anywhere of objects. In this way, we can alleviate the annotation burden.

Overall Framework: With this new setting that a small number of supervised images and a large number of weakly supervised images, we adapt self-training as our default training pipeline, which has made considerable progress in semi-supervised learning (*e.g*. Lee [13],Noise-Student [32],STAC [26]). The steps are summarized as follows:

1. **Train a teacher model** on available labeled images.

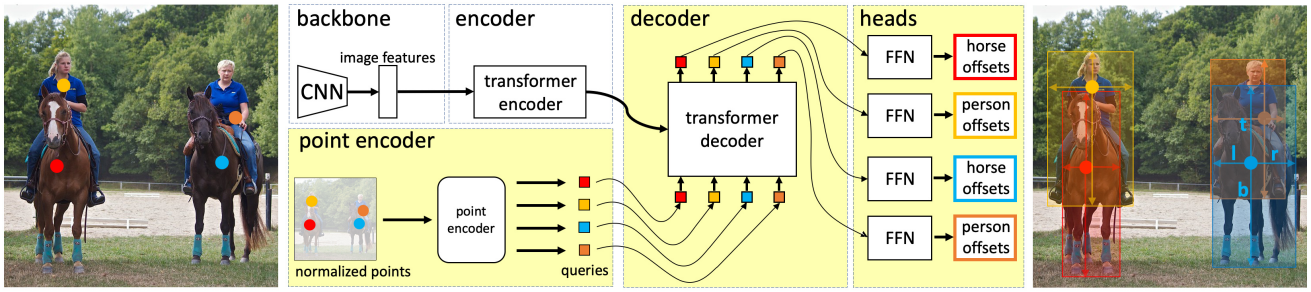


Figure 3. Point DETR takes the image and its corresponding object points as input. The object points are normalized to $[0, 1]^2$, and are encoded into object queries by the point encoder module. The transformer decoder takes the object queries and additionally attends to the image features (extracting by backbone and encoder). The output of the transformer decoder is passed to the head, generating box predictions. The box predictions are the relative offsets from the four sides of a bounding box to the point location. The components that are different from DETR are highlighted by light yellow.

2. **Generate pseudo-labels of weakly point annotated images** using the trained teacher model.

3. **Train a student model** with fully labeled images and pseudo-labeled images.

The overall framework is shown in Figure 2. For most self-training based detection methods, hyper-parameters are selected carefully since they must keep true object boxes and screen out false ones as much as possible. Instead, we can directly predict the corresponding object box for each point annotation without duplicate object boxes. Although choosing hyper-parameters is no longer an obstacle to performance, predicting object boxes from point-level annotations with existing detectors remains a problem.

Discussion on Existing Detectors: Existing detectors can be divided into two categories: multi-level feature detectors and single-level feature detectors. For multi-level detectors (*e.g.* FCOS[29]), it is difficult for them to predict object boxes with point annotations since point annotations do not have feature-level information, which is used to select one prediction from multi-level box predictions (Figure 8b). On the other hand, single-level feature detectors (*e.g.* Faster R-CNN[22]) suffer from the bad performance or strict requirement on point annotations though avoiding choosing feature map levels (Figure 8c). For more experiments see Section 4.3.

3.1. Point DETR

To avoid the drawbacks of existing detectors in the WSSOD with point annotations task, we introduce a novel detector, Point DETR: adding a point encoder to DETR. It transforms point annotations into object queries, extracts image features for each object query, and outputs the corresponding object box. Next, we introduce a key element of Point DETR, point encoder, which is critical to the WSSOD with point annotations task.

DETR: We begin by reviewing DETR [4], which is an end-to-end set-based object detector. DETR consists of a CNN backbone, an encoder-decoder transformer, and a prediction head. DETR first extracts a single-level 2D feature map from the CNN backbone, flattens it, and supplements it with a positional encoding. Then, the encoder-decoder transformer takes as input a fixed set of object queries (learned positional embeddings) and attends to 1D image feature embeddings. Finally, the output embeddings of the transformer are passed to the prediction head that predicts either a detection (class and bounding box) or a "no object" class.

Point DETR: Point DETR, as shown in Figure 3, adopts most components of DETR. To fit the point annotated images, Point DETR has a special module, point encoder. Point encoder can encode the point annotations into object queries, which are taken as input by the transformer decoder. Unlike the object queries in DETR that are learned positional embeddings, these object queries are specific instance embeddings which contain position and category information of object instances. Thus, these object queries have a one-to-one correspondence with object instances. Moreover, the number of object queries varies with the number of object instances in an image instead of a fixed number(e.g. 100) like DETR.

During training, we simply define the loss of each object query as $\mathcal{L} = \mathcal{L}_{box}$, since we already have category for each object query and only need to regress the object box. The bounding-box loss $\mathcal{L}_{box}$ is identical as it defined in DETR. But, for the box prediction $\hat{b}_i$, it calculated by $\hat{b}_i = \hat{b}_i^{init} + \Delta\hat{b}_i$, where $\hat{b}_i^{init} \in [0, 1]^4$ is (x, y, x, y), (x, y) is the location of point annotation and $\Delta\hat{b}_i \in [0, 1]^4$ is the relative offsets w.r.t. the point location (x, y) following FCOS [29]. In our experiments, we show this way of regression can alleviate the mismatch between point annotation and object box, see Section 4.3.

Point Encoder: In point DETR, how to encode point annotations into object queries is critical for point encoder.

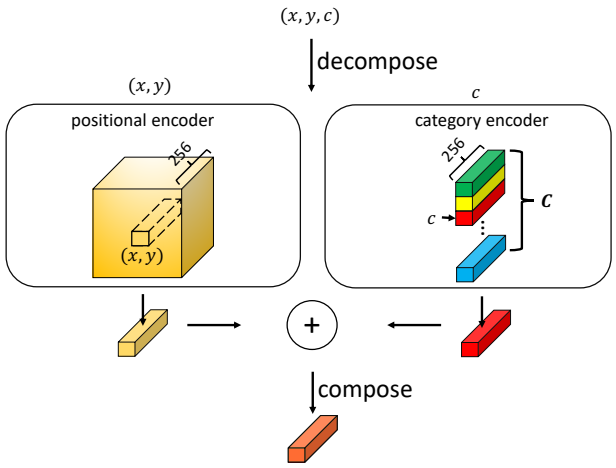


Figure 4. Point encoder. For each point (x, y, c), it encodes the position (x, y) and category c separetely, and then takes the element-wise addition as the point embedding.

As shown in Figure 4, a point annotation (x, y, c) is decomposed to a 2D coordinate $(x, y) \in [0, 1]^2$ and category index c. Based on (x, y), the position embedding $e_{pos} \in \mathbb{R}^{256}$ is extracted from fixed spatial positional encodings [30, 18, 4], which is the same as one used in the transformer encoder. For category embedding $e_{cat} \in \mathbb{R}^{256}$, it is obtained from predefined learnable category embeddings by category index, *i.e.* c. In the end, we fuse these embedding to get the object query by sum operation.

Though point encoder is simple and easily implemented, it bridges the divisions between point annotations and object queries. In the experiments, we show the essentials of every component (positional encoder and category encoder) in point encoder, see section 4.3.

4. Experiments

We evaluate our models on the COCO 2017 detection dataset [16] with synthetic point annotations (details in section 4.1). We report the standard COCO metrics including AP (averaged over IoU thresholds), AP_{50}, AP_{75}. In addition, to show the quality of generated pseudo-boxes, we also calculate the $mIoU$ between the generated pseudo-boxes and the ground truth bounding boxes.

With existing detectors that can not be directly applied to our point annotated settings, we make some modifications to existing detectors: FCOS and Faster R-CNN. These modified detectors are denoted as FCOS† and Faster R-CNN†, respectively. For FCOS†, we separately extract point features from multi-level feature maps by bilinear interpolation [9] and predict the corresponding object box, finally use the box prediction with the highest point category score as the pseudo-box. As for Faster R-CNN†, we extract point features from one-level feature map, and then predict boxes for different anchors, finally use one with the highest point category score as the pseudo-box.

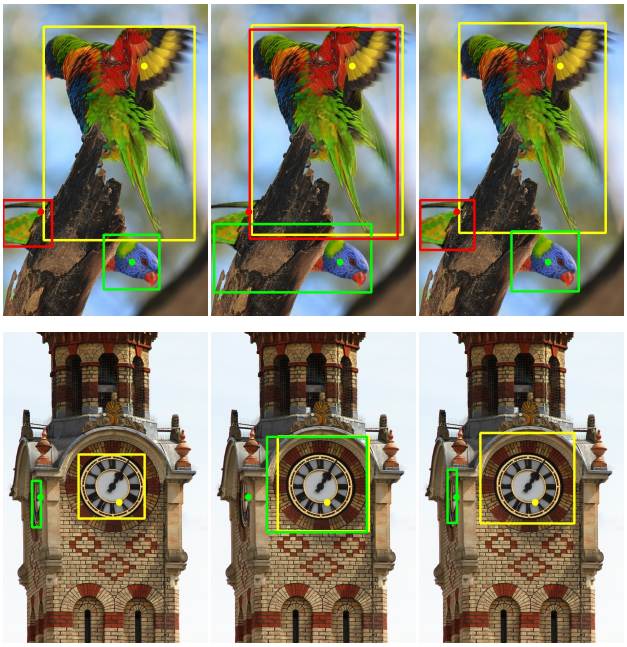


Figure 5. **Absolute vs. Relative Regression:** Different colors to distinguish instances and the color of the point annotation is consistent with its corresponding box. Best viewed in color.

4.1. Implementation Details

We use ResNet-50 [8] as the default backbone for different detectors and set the hyper-parameters following these detectors.

Dataset: We train the model with 118k training images and evaluate the performance of the detectors on the remaining 5k val images. Specially, for our point annotated setting, we randomly sample 5%, 10%, 20%, 30%, 40%, 50% of training images as the fully labeled set and use the rest of the images as a weakly labeled set. In this paper, we noted them as different data settings for simplicity, *e.g.* 20% data setting. For the weakly labeled set, we synthesize the point annotations for each object as follows: (a) if the object has instance segmentation, randomly sample a point from the instance mask as the point annotation for the object; (b) if not, simply randomly sample a point in its bounding box.

Training: In our framework, there are two models: the teacher model and student model. Our teacher model includes Point DETR, FCOS†, and Faster R-CNN†. While we simply choose FCOS as the default student model since the student model is only used to evaluate the effectiveness of the teacher model. We show by experiments (in section 4.3) that our method is robust to the architecture of the student model.

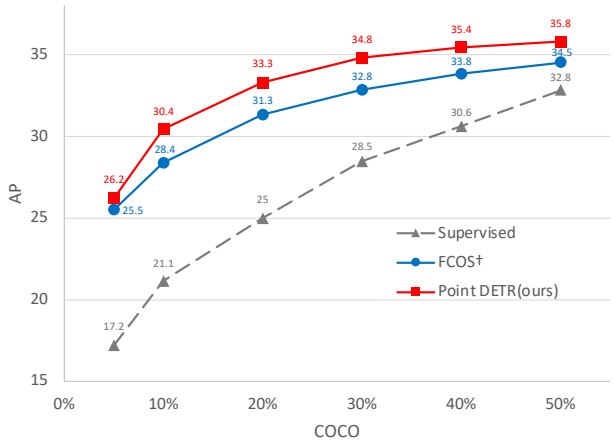


Figure 6. Comparison in APs of the student model (*i.e*. FCOS) for different methods on MS-COCO. "Supervised" refers to the student models trained on labeled data only.

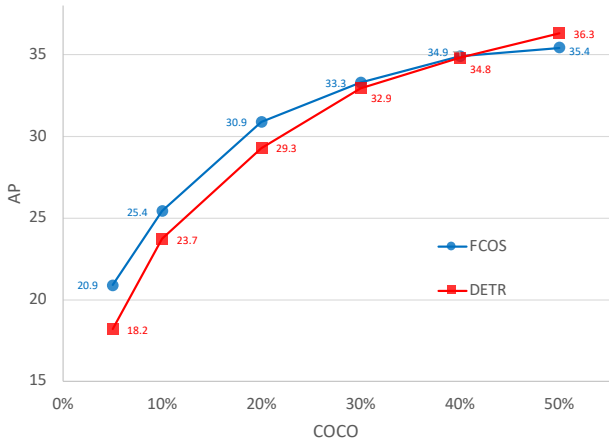


Figure 7. Comparison in APs of FCOS and DETR to demonstrate the improvement comes from our method rather than a stronger teacher model. FCOS trained in DETR augmentation for a fair comparison. In most cases ($5\% \sim 40\%$), FCOS has a better performance than DETR.

For the training of the teacher model, it is simple for FCOS† and Faster R-CNN†. We train them with their default training settings. For a fair comparison, we also use data augmentation as shown in [4]. For Point DETR, it follows most of the training settings used in [4] with several differences: we train the model for 108 epochs on 8 GTX 1080Ti GPUs, with 2 images per GPU. To ensure training stability, we use a warmup scheme [7] in the first epoch. The learning rate is reduced by a factor of 10 at epoch 72 and 96, respectively. In the training, we randomly sample a point in each bounding box and transform points into point annotations. With these point annotations, we train Point DETR as shown in Figure 3.

For the default student model, we combine the fully labeled images and pseudo-labeled images generated by the teacher model to train the student, as showed in Figure 2.

4.2. Main Results

We first show the effectiveness of Point DETR on different data split settings, see Figure 6. We train the student model (*i.e*. FCOS) only with the fully annotated images (noted as "Supervised"). By comparing "Supervised" with the student model trained with pseudo-boxes, we can evaluate the benefits brought by the pseudo-boxes. Point DETR and FCOS† outperform "Supervised" by a large margin. This demonstrates that images with point annotations can improve the performance of the detection task. Furthermore, Point DETR outperforms FCOS† by a considerable margin.

Next, we verify the factors that contribute to the great performance of our method. We compare the accuracies of FCOS and DETR as shown in Figure 7, DETR performs worse than FCOS in most settings. Given that our method based on DETR achieves greater performance, we can conclude that the high accuracy of our method does not mainly benefit from its strong representation. Moreover, we conduct quality and quantity experiments to show the superiority of our method on pseudo-object boxes, see Figure 8. FCOS†, a multi-level feature detector, can not predict object boxes well due to FPN, and Faster R-CNN†, a single-level feature detector, also has difficulty regressing box owing to poor representation. But, Point DETR can generate a more precise object box than other detectors. Specifically, the $mIoU$ of Point DETR is larger than FCOS† and Faster R-CNN† by 6.3 and 5.3, respectively. Based on the above experiments, our method achieves considerable performance mainly by generating precise pseudo-object boxes from point annotations.

4.3. Ablation Experiments

We conduct the ablation experiments at 20% data setting. Results are shown in Table 1 and discussed in detail next.

Point Encoder: Table 1a shows the effectiveness of the components in the point encoder module (as we shown in Figure 4) . Point DETR with only positional embeddings outperforms one with only category embeddings and point DETR has a severe loss in AP (18.6 points) without positional embeddings. Based on that our method only regresses the object boxes, this suggests that it is difficult to learn the relative offsets of a point with respect to the bounding box without positional embeddings. We also find that adding category embeddings to positional embeddings can boost the performance by 2 points. We conjecture this improvement is caused by the that category embeddings can provide object prior, such as object shape.

Student Model: For student model, we use FCOS [29] as the default detector. To exploit robustness of our approach,

	pos?	cate?	AP	AP_{50}	AP_{75}
Point Encoder		✓	14.7	34.3	10.4
	✓		31.3	51.0	32.6
	✓	✓	**33.3**	**53.5**	**34.8**

(a) **Point Encoder:** The effectiveness of positional encoder and category encoder.

Teacher	Student	AP	AP_{50}	AP_{75}
FCOS†	RetinaNet	30.4	49.9	31.6
Ours		**32.5**	**52.8**	**33.7**
FCOS†	FCOS	31.3	50.7	32.6
Ours		**33.3**	**53.5**	**34.8**

(b) **Student Model:** RetinaNet [15] as the student model demonstrates the effectiveness of our approach is not related to the student model.

	AP	AP_{50}	AP_{75}
Faster R-CNN†	31.4	51.6	32.6
Ours	**33.3**	**53.5**	**34.8**

(c) **Single-Level Detector:** Point DETR vs. Faster R-CNN†.

	Supervised	AP	AP_{50}	AP_{75}
UFO^2	29.1	30.1	-	-
Ours	28.1	**33.5**	53.8	34.8

(d) **Comparison with UFO^2 [23]:** "Supervised" refers to the model trained with fully labeled data only.

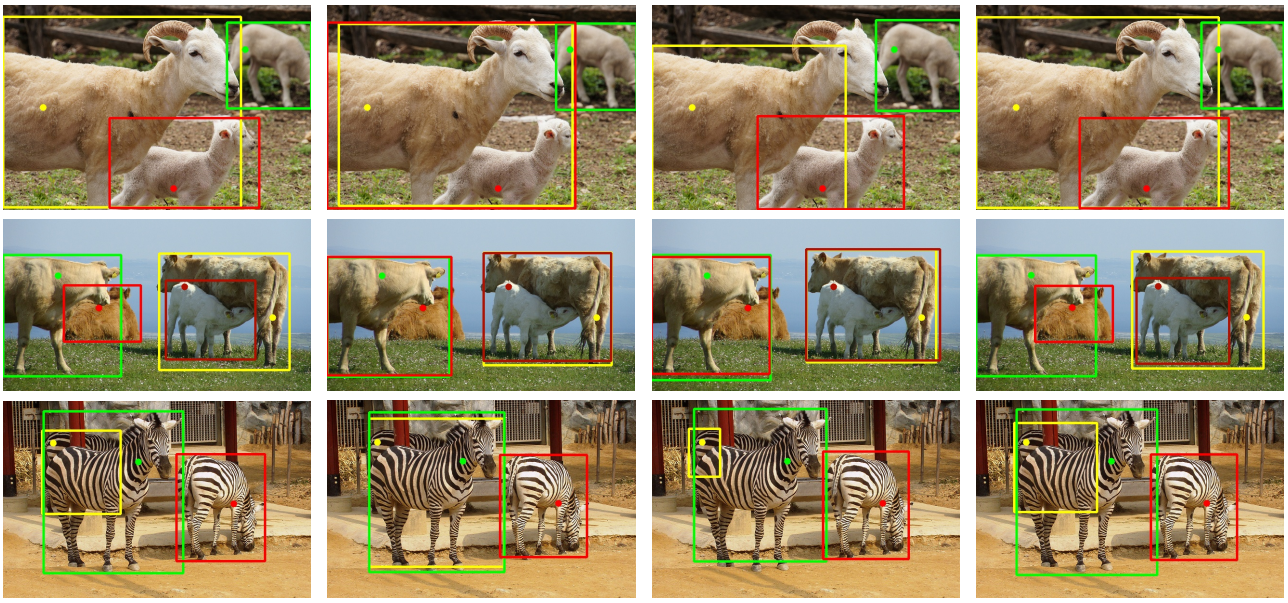
	center?	AP	AP_{50}	AP_{75}
Ours		**33.3**	53.5	**34.8**
Ours	✓	33.3	**53.6**	34.6

(e) **Point Location:** The effectiveness of the point location.

	points?	score?	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AR_1	AR_{10}	AR_{100}	AR_s	AR_m	AR_l
DETR			19.1	33.2	18.7	5.6	20.2	31.3	22.9	32.8	33.6	12.0	35.0	51.0
		✓	**26.9**	43.8	**27.5**	9.2	27.7	**39.9**	24.2	33.2	33.6	12.0	35.0	51.0
Ours	✓		26.8	**52.3**	24.2	**12.6**	**29.5**	38.9	**30.7**	**44.0**	**44.5**	**22.8**	**46.9**	**63.9**
Δ			-0.1	+8.5	-3.3	+3.4	+1.8	-1.0	+6.5	+10.8	+10.9	+10.8	+11.9	+12.9

(f) **Point Annotations:** To confirm the benefits of point annotations, we compare Point DETR (with points) vs. DETR (without points) by analyzing the generated boxes with respect to ground truth boxes. With AR far exceeding DETR, our AP remains comparable.

Table 1. **Ablations.** All ablation experiments are conducted at 20% data setting except (d).



(a) Ground truth. (b) FCOS†. *mIoU*: 57.1. (c) Faster R-CNN†. *mIoU*: 58.1. (d) Point DETR (**ours**). *mIoU*: 63.4.

Figure 8. Visualized results of FCOS†, Faster R-CNN† and Point DETR (**ours**). The $mIoU$ between the ground truth boxes and pseudo-boxes on the entire weakly labeled images are provided. Different colors to distinguish instances, and the color of the point annotation is consistent with its corresponding box. Best viewed in color.

we replace FCOS with RetinaNet [15]. In Table 1b, we find that our method has a 2.1 AP gain over baseline. This demonstrates that our method is robust to the student model.

Single-Level Detector: We compare Point DETR with single-level feature detectors and choose Faster R-CNN† as the default single-level feature detector. As shown in Table 1c, Point DETR outperforms Faster R-CNN† by 1.9 points. This highlights that effectiveness of Point DETR.

Comparison with UFO^2 [23]: To show the effectiveness of our method, we compare Point DETR with UFO^2. For fair comparison, we train Point DETR following the dataset split in UFO^2: COCO-35 (fully labeled images) and

COCO-80 (point labeled images). As shown in Table 1d, our method has inferior performance than UFO^2 when is only trained on COCO-35, but it outperforms UFO^2 by 3.4 points adding COCO-80. This indicates that our method can make better use of point annotation information.

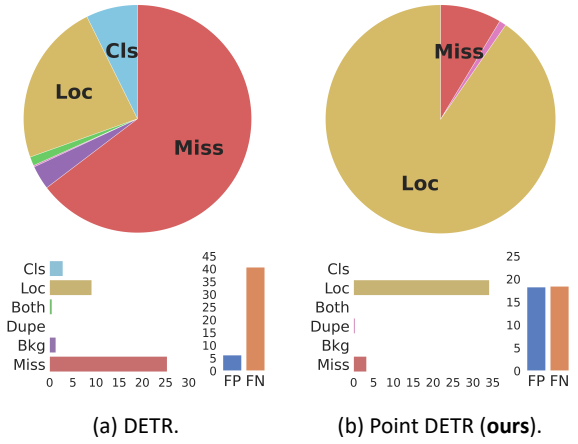
Point Location: To validate that our method is robust to the point location, we compare performances between two point location schemes: center point and arbitrary point on objects. As shown in Table 1e, our method has comparable performance between these two point location scheme.

Absolute vs. Relative Regression: Our method use relative regression to predict object boxes. In Figure 5, we compare our relative regression with absolute regression used in DETR. Absolute regression incorrectly matches the point with the bounding box that does not correspond (*e.g.* the green clock in Figure 5b) in some cases. Compared with absolute regression, relative regression has little mismatch problem between point and object box, we attribute it to its use of the prior knowledge: the point is *in* the bounding box.

Point Annotations: To evaluate the effectiveness of point annotations, we compare our approach with the method without point annotations. For a fair comparison, we use DETR as the method without point annotations. We apply a self-training framework (following [26]) on DETR directly. We train DETR with only fully labeled images first, and then generate pseudo-boxes for weakly labeled images *without* point annotations. To remove duplicate boxes, we use a threshold $\tau = 0.7$ which results in the best box predictions on weakly labeled images. For the generated pseudo-object boxes, they do not have the one-to-one correspondence with point annotations. Thus, it is impractical to calculate the $mIoU$ between generated boxes and ground truth boxes. To make comparison available, we use standard COCO metrics instead of $mIoU$, as shown in Table 1f. Point DETR performs on par with DETR on mAP and outperforms DETR by a large margin in the recall. Specifically, Point DETR achieves over 10 points of improvements in various AR metrics (*e.g.* AR_s, AR_m, AR_l, AR_{100}) and its AP is comparable with DETR (26.8 vs. 26.9). Though Point DETR is 3.3 points AP_{75} lower than DETR, which is possibly explained by that high τ screens out low-quality boxes and remains high-quality boxes, the higher recall of Point DETR can offset this bad influence.

Additionally, we set the classification score of pseudo-boxes generated by DETR to a constant value like 0.5, which is consistent with our method. In this setting, the performance of DETR drops by a large margin and performs much worse than our method. This highlights that with point annotations, our method does not suffer from the quality of classification score.

We also analyze the errors of the generated boxes by TIDE [3] in Figure 9. Missed ground truths is the largest



(a) DETR. (b) Point DETR (**ours**).

Figure 9. Diagnosing the errors of generated pseudo-boxes by TIDE [3]. Different error types: **Cls**: localized correctly but classified incorrectly, **Loc**: classified correctly but localized incorrectly, **Both**: both cls and loc error, **Dupe**: duplicate detection error, **Bkg**: detected background as foreground, **Miss**: missed ground truth error.

issue for DETR, while it does not affect the performance of Point DETR greatly. This is explained by that with point annotations, Point DETR does not miss objects like DETR. In addition, unlike DETR, location error is the main challenge of Point DETR. Also, Point DETR also has duplicate detection errors. This is caused by those point annotations residing in multiple bounding boxes that would predict object boxes for wrong ground truths, which results in a ground truth that has multiple box predictions.

5. Conclusion

In this work, we verify the effectiveness of point annotations in the weakly semi-supervised detection task. We also show that the power of point annotations is hindered by existing detectors. In order to solve this, we propose the Point DETR which applies a point encoder to the point annotations to establish the one-to-one correspondence between point annotations and objects. Our approach is simple and implemented easily. We demonstrate its efficacy by the extensive experimental analysis showing that it achieves state-of-the-art performance.

Acknowledgments This work is supported by National Key R&D Program of China (2020AAA0105200).

References

[1] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[3] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. *arXiv preprint arXiv:2008.08115*, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

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

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[10] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in neural information processing systems*, pages 10759–10768, 2019.

[11] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1377–1385, 2017.

[12] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[13] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[17] Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Semi-supervised object detection with unlabeled data. In *VISIGRAPP (5: VISAPP)*, pages 289–296, 2019.

[18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.

[19] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8843–8850, 2019.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[21] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G Schwing, and Jan Kautz. Ufo ^{2}: A unified framework towards omni-supervised object detection. In *European Conference on Computer Vision*, pages 288–313. Springer, 2020.

[24] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2121–2131, 2015.

[25] Miaojing Shi, Holger Caesar, and Vittorio Ferrari. Weakly supervised object localization using things and stuff transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3381–3390, 2017.

[26] 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.

[27] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[28] Peng Tang, Chetan Ramaiah, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. *arXiv preprint arXiv:2001.05086*, 2020.

[29] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[31] Keze Wang, Xiaopeng Yan, Dongyu Zhang, Lei Zhang, and Liang Lin. Towards human-machine cooperation: Self-supervised sample mining for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1605–1613, 2018.

[32] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[33] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017.

[34] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12234–12244, 2020.

[35] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[36] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017.