

# RepPoints: Point Set Representation for Object Detection

Ze Yang<sup>1†\*</sup> Shaohui Liu<sup>2,3†\*</sup> Han Hu<sup>3</sup> Liwei Wang<sup>1</sup> Stephen Lin<sup>3</sup>  
<sup>1</sup>Peking University <sup>2</sup>Tsinghua University  
<sup>3</sup>Microsoft Research Asia

yangze@pku.edu.cn, blueber2y@gmail.com, wanglw@cis.pku.edu.cn  
{hanhu, stevelin}@microsoft.com

## Abstract

Modern object detectors rely heavily on rectangular bounding boxes, such as anchors, proposals and the final predictions, to represent objects at various recognition stages. The bounding box is convenient to use but provides only a coarse localization of objects and leads to a correspondingly coarse extraction of object features. In this paper, we present **RepPoints** (representative points), a new finer representation of objects as a set of sample points useful for both localization and recognition. Given ground truth localization and recognition targets for training, RepPoints learn to automatically arrange themselves in a manner that bounds the spatial extent of an object and indicates semantically significant local areas. They furthermore do not require the use of anchors to sample a space of bounding boxes. We show that an anchor-free object detector based on RepPoints, implemented without multi-scale training and testing, can be as effective as state-of-the-art anchor-based detection methods, with 42.8 AP and 65.0 AP<sub>50</sub> on the COCO test-dev detection benchmark.

## 1. Introduction

Object detection aims to localize objects in an image and provide their class labels. As one of the most fundamental tasks in computer vision, it serves as a key component for many vision applications, including instance segmentation [34], human pose analysis [43], and visual reasoning [47]. The significance of the object detection problem together with the rapid development of deep neural networks has led to substantial progress in recent years [7, 12, 11, 38, 15, 3].

In the object detection pipeline, bounding boxes, which encompass rectangular areas of an image, serve as the basic element for processing. They describe target locations of objects throughout the stages of an object detector, from

\*Equal contribution. †The work is done when Ze Yang and Shaohui Liu are interns at Microsoft Research Asia.

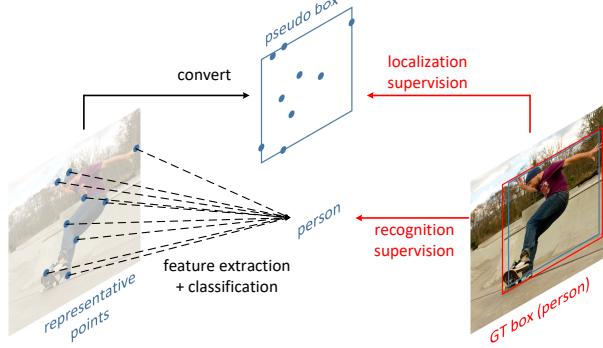


Figure 1. **RepPoints** are a new representation for object detection that consists of a set of points which indicate the spatial extent of an object and semantically significant local areas. This representation is learned via weak localization supervision from rectangular ground-truth boxes and implicit recognition feedback. Based on the richer RepPoints representation, we develop an anchor-free object detector that yields improved performance compared to using bounding boxes.

anchors and proposals to final predictions. Based on these bounding boxes, features are extracted and used for purposes such as object classification and location refinement. The prevalence of the bounding box representation can partly be attributed to common metrics for object detection performance, which account for the overlap between estimated and ground truth bounding boxes of objects. Another reason lies in its convenience for feature extraction in deep networks, because of its regular shape and the ease of subdividing a rectangular window into a matrix of pooled cells.

Though bounding boxes facilitate computation, they provide only a coarse localization of objects that does not conform to an object’s shape and pose. Features extracted from the regular cells of a bounding box may thus be heavily influenced by background content or uninformative foreground areas that contain little semantic information. This may result in lower feature quality that degrades classification performance in object detection.

In this paper, we propose a new representation, called **RepPoints**, that provides more fine-grained localization and

facilitates classification. Illustrated in Fig. 1, RepPoints are a set of points that learn to adaptively position themselves over an object in a manner that circumscribes the object’s spatial extent and indicates semantically significant local areas. The training of RepPoints is driven jointly by object localization and recognition targets, such that the RepPoints are tightly bound by the ground-truth bounding box and guide the detector toward correct object classification. This adaptive and differentiable representation can be coherently used across the different stages of a modern object detector, and does not require the use of anchors to sample over a space of bounding boxes.

RepPoints differs from existing non-rectangular representations for object detection, which are all built in a bottom-up manner [44, 24, 54]. These bottom-up representations identify individual points (e.g., bounding box corners or object extremities) and rely on handcrafted clustering to group them into object models. Their representations furthermore either are still axis-aligned like bounding boxes [44, 24] or require ground truth object masks as additional supervision [54]. In contrast, RepPoints are learned in a top-down fashion from the input image / object features, allowing for end-to-end training and producing fine-grained localization without additional supervision.

To demonstrate the power of the RepPoints representation, we present an implementation within a deformable ConvNets framework [4], which provides recognition feedback suitable for guiding the adaptive sampling while maintaining convenience in feature extraction. This anchor-free detection system is found to have strong classification ability while also accurately localizing objects. Without multi-scale training and testing, our clean detector achieves 42.8 AP and 65.0  $AP_{50}$  on the COCO benchmark [29], not only surpassing all existing anchor-free detectors but also performing on-par with state-of-the-art anchor-based baselines.

## 2. Related Work

**Bounding boxes for the object detection problem.** The bounding box has long been the dominant form of object representation in the field of object detection. One reason for its prevalence is that a bounding box is convenient to annotate with little ambiguity, while providing sufficiently accurate localization for the subsequent recognition process. This may explain why the major benchmarks all utilize annotations and evaluations based on bounding boxes [8, 29, 23]. In turn, these benchmarks motivate object detection methods to use the bounding box as their basic representation in order to align with the evaluation protocols.

Another reason for the dominance of bounding boxes is that almost all image feature extractors, both before [45, 5] and during the deep learning era [22, 40, 42, 16], are based on an input patch with a regular grid form. It is thus con-

venient to use the bounding box representation to facilitate feature extraction [12, 11, 38].

Although the proposed *RepPoints* has an irregular form, we show that it can be amenable to convenient feature extraction. Our system utilizes RepPoints in conjunction with deformable convolution [4], which naturally aligns with RepPoints in that it aggregates information from input features at several sample points. Besides, a rectangular *pseudo box* can be readily generated from RepPoints (see Section 3.2), allowing the new representation to be used with object detection benchmarks.

**Bounding boxes in modern object detectors.** The best-performing object detectors to date generally follow a multi-stage recognition paradigm [27, 4, 14, 26, 2, 41, 30], and the bounding box representation appears in almost all stages: 1) as pre-defined anchors that serve as hypotheses over the bounding box space; 2) as refined object proposals connecting successive recognition stages; and 3) as the final localization targets.

1) *Bounding boxes as anchors.* Significant improvements in object detection have been achieved through the use of dense predefined anchors. Most existing detectors [38, 28] use anchors of different aspect ratios and scales. Recently, there is a trend towards better modeling of anchors, such as by guided anchoring [46], learning anchor functions [50], and adaptive optimization [53].

In contrast, with the proposed *RepPoints*, we show that an anchor-free detector can reach performance comparable to modern detectors that rely on a large set of dense anchors.

2) *Bounding boxes as object proposals.* Progressive refinement of object proposals is crucial to the success of multi-stage object detectors. The improved localization increases the quality of extracted features, leading to improved detection performance from more recognition stages [2]. However, the mechanics of bounding box refinement can be seen as non-intuitive, where the width/height change is directly regressed from an exponential function on input object features.

With RepPoints, refinement of object localization corresponds to adjusting the positions of its sample points towards more semantically meaningful features, which both aids in classification and naturally determines the object’s spatial extent.

3) *Bounding boxes as the final target localization.* Existing methods usually adopt the same bounding box regression method as used for object proposals to produce the final target localization. Recently, some alternatives have been presented for this purpose, using IoU-Net [20], Generalized IoU loss [39] or consistent optimization [21].

**Other representations for object detection.** To address limitations of rectangular bounding boxes, there have been

some attempts to develop more flexible object representations. These include an elliptic representation for pedestrian detection [25] and a rotated bounding box to better handle rotational variations [17, 55].

Other works aim to represent an object in a bottom-up manner. Early bottom-up representations include DPM [9] and Poselet [1]. Recently, bottom-up approaches to object detection have been explored with deep networks [24, 54]. CornerNet [24] first predicts top-left and bottom-right corners and then employs a specialized grouping method [32] to obtain the bounding boxes of objects. However, the two opposing corner points still essentially model a rectangular bounding box. ExtremeNet [54] is proposed to locate the extreme points of objects in the x- and y-directions [33] with supervision from ground-truth mask annotations. In general, bottom-up detectors benefit from a smaller hypothesis space (for example, CornerNet and ExtremeNet both detect 2-d points instead of directly detecting a 4-d bounding box) and potentially finer-grained localization. However, they have limitations such as relying on handcrafted clustering or post-processing steps to compose whole objects from the detected points.

Similar to these bottom-up works, *RepPoints* is also a flexible object representation. However, the representation is constructed in a top-down manner, without the need for handcrafted clustering steps. *RepPoints* can automatically learn extreme points and key semantic points without supervision beyond ground-truth bounding boxes, unlike ExtremeNet [54] where additional mask supervision is required.

**Deformation modeling in object recognition.** One of the most fundamental challenges for visual recognition is to recognize objects with various geometric variations. To effectively model such variations, a possible solution is to make use of bottom-up composition of low-level components. Representative detectors along this direction include DPM [9] and Poselet [1]. An alternative is to implicitly model the transformations in a top-down manner, where a lightweight neural network block is applied on input features, either globally [19] or locally [4].

*RepPoints* is inspired by these works, especially the top-down deformation modeling approach [4]. The main difference is that we aim at developing a flexible object representation for accurate *geometric localization* in addition to semantic feature extraction. In contrast, both the deformable convolution and deformable ROI pooling methods are designed to improve feature extraction only. The inability of deformable ROI pooling to learn accurate geometric localization is examined in Section 4. In this sense, we expand the usage of adaptive sample points in previous geometric modeling methods [19, 4] to include finer localization of objects.

### 3. The RepPoints Representation

We first review the bounding box representation and its use within multi-stage object detectors. This is followed by a description of RepPoints and its differences from bounding boxes.

#### 3.1. Bounding Box Representation

The bounding box is a 4-d representation encoding the spatial location of an object,  $\mathcal{B} = (x, y, w, h)$ , with  $x, y$  denoting the center points and  $w, h$  denoting the width and height. Due to its simplicity and convenience in use, modern object detectors heavily rely on bounding boxes for representing objects at various stages of the detection pipeline.

**Review of Multi-Stage Object Detectors** The best performing object detectors usually follow a multi-stage recognition paradigm, where object localization is refined stage by stage. The role of the object representation through the steps of this pipeline is as follows:

$$\begin{array}{ccc} \text{bbox anchors} & \xrightarrow{\text{bbox reg.}} & \text{bbox proposals (S1)} \\ & \xrightarrow{\text{bbox reg.}} & \text{bbox proposals (S2)} \\ & \dots & \\ & \xrightarrow{\text{bbox reg.}} & \text{bbox object targets} \end{array} \quad (1)$$

At the beginning, multiple anchors are hypothesized to cover a range of bounding box scales and aspect ratios. In general, high coverage is obtained through dense anchors over the large 4-d hypothesis space. For instance, 45 anchors per location are utilized in RetinaNet [28].

For an anchor, the image feature at its center point is adopted as the object feature, which is then used to produce a confidence score about whether the anchor is an object or not, as well as the refined bounding box by a bounding box regression process. The refined bounding box is denoted as “bbox proposals (S1)”.

In the second stage, a refined object feature is extracted from the refined bounding box proposal, usually by ROI-pooling [11] or ROI-Align [14]. For the two-stage framework [38], the refined feature will produce the final bounding box target by bounding box regression. For the multi-stage approach [2], the refined feature is used to generate intermediate refined bounding box proposals (S2), also by bounding box regression. This step can be iterated multiple times before producing the final bounding box target.

In this framework, bounding box regression plays a central role in progressively refining object localization and object features. We formulate the process of bounding box regression in the following paragraph.

**Bounding Box Regression** Conventionally, a 4-d regression vector  $(\Delta x_p, \Delta y_p, \Delta w_p, \Delta h_p)$  is predicted to map the

current bounding box proposal  $\mathcal{B}_p = (x_p, y_p, w_p, h_p)$  into a refined bounding box  $\mathcal{B}_r$ , where

$$\mathcal{B}_r = (x_p + w_p \Delta x_p, y_p + h_p \Delta y_p, w_p e^{\Delta w_p}, h_p e^{\Delta h_p}). \quad (2)$$

Given the ground truth bounding box of an object  $\mathcal{B}_t = (x_t, y_t, w_t, h_t)$ , the goal of bounding box regression is to have  $\mathcal{B}_r$  and  $\mathcal{B}_t$  as close as possible. Specifically, in the training of an object detector, we use the distance between the predicted 4-d regression vector and the expected 4-d regression vector  $\hat{\mathcal{F}}(\mathcal{B}_p, \mathcal{B}_t)$  as the learning target, using a smooth  $l_1$  loss:

$$\hat{\mathcal{F}}(\mathcal{B}_p, \mathcal{B}_t) = \left( \frac{x_t - x_p}{w_p}, \frac{y_t - y_p}{h_p}, \log \frac{w_t}{w_p}, \log \frac{h_t}{h_p} \right). \quad (3)$$

This bounding box regression process is widely used in existing object detection methods. It performs well in practice when the required refinement is small, but it tends to perform poorly when there is large distance between the initial representation and the target. Another issue lies in the scale difference between  $\Delta x, \Delta y$  and  $\Delta w, \Delta h$ , which requires tuning of their loss weights for optimal performance.

### 3.2. RepPoints

As previously discussed, the 4-d bounding box is a coarse representation of object location. The bounding box representation considers only the rectangular spatial scope of an object, and does not account for shape and pose and the positions of semantically important local areas, which could be used toward finer localization and better object feature extraction.

To overcome the above limitations, *RepPoints* instead models a set of adaptive sample points:

$$\mathcal{R} = \{(x_k, y_k)\}_{k=1}^n, \quad (4)$$

where  $n$  is the total number of sample points used in the representation. In our work,  $n$  is set to 9 by default.

**RepPoints refinement** Progressively refining the bounding box localization and feature extraction is important for the success of multi-stage object detection methods. For RepPoints, the refinement can be expressed simply as

$$\mathcal{R}_r = \{(x_k + \Delta x_k, y_k + \Delta y_k)\}_{k=1}^n, \quad (5)$$

where  $\{(\Delta x_k, \Delta y_k)\}_{k=1}^n$  are the predicted offsets of the new sample points with respect to the old ones. We note that this refinement does not face the problem of scale differences among the bounding box regression parameters, since the offsets are at the same scale in the refinement process of *RepPoints*.

**Converting RepPoints to bounding box** To take advantage of bounding box annotations in the training of RepPoints, as well as to evaluate RepPoint-based object detectors, a method is needed for converting RepPoints into a bounding box. We perform this conversion using a pre-defined converting function  $\mathcal{T} : \mathcal{R}_P \rightarrow \mathcal{B}_P$ , where  $\mathcal{R}_P$  denotes the RepPoints for object  $P$  and  $\mathcal{T}(\mathcal{R}_P)$  represents a *pseudo box*.

Three converting functions are considered for this purpose:

- $\mathcal{T} = \mathcal{T}_1$ : **Min-max function.** Min-max operation over both axes are performed over the *RepPoints* to determine  $\mathcal{B}_p$ , equivalent to the bounding box over the sample points.
- $\mathcal{T} = \mathcal{T}_2$ : **Partial min-max function.** Min-max operation over a subset of the sample points is performed over both axes to obtain the rectangular box  $\mathcal{B}_p$ .
- $\mathcal{T} = \mathcal{T}_3$ : **Moment-based function.** The mean value and the second-order moment of the RepPoints is used to compute the center point and scale of the rectangular box  $\mathcal{B}_p$ , where the scale is multiplied by globally-shared learnable multipliers  $\lambda_x$  and  $\lambda_y$ .

These functions are all differentiable, enabling end-to-end learning when inserted into an object detection system. In our experiments, we found them to work comparably well.

**Learning RepPoints** The learning of RepPoints is driven by both an object localization loss and an object recognition loss. To compute the object localization loss, we first convert *RepPoints* into a *pseudo box* using the previously discussed transformation function  $\mathcal{T}$ . Then, the difference between the converted *pseudo box* and the ground-truth bounding box is computed. In our system, we use the smooth  $l_1$  distance between the top-left and bottom-right points to represent the localization loss. This smooth  $l_1$  distance does not require the tuning of different loss weights as done in computing the distance between bounding box regression vectors (i.e., for  $\Delta x, \Delta y$  and  $\Delta w, \Delta h$ ). Figure 3 indicates that when the training is driven by this combination of object localization and object recognition losses, the extreme points and semantic key points of objects are automatically learned ( $\mathcal{T}_1$  is used in transforming RepPoints to pseudo box).

## 4. RPDet: an Anchor Free Detector

We design an anchor-free object detector that utilizes RepPoints in place of bounding boxes as its basic representation. Within a multi-stage pipeline, the object representa-

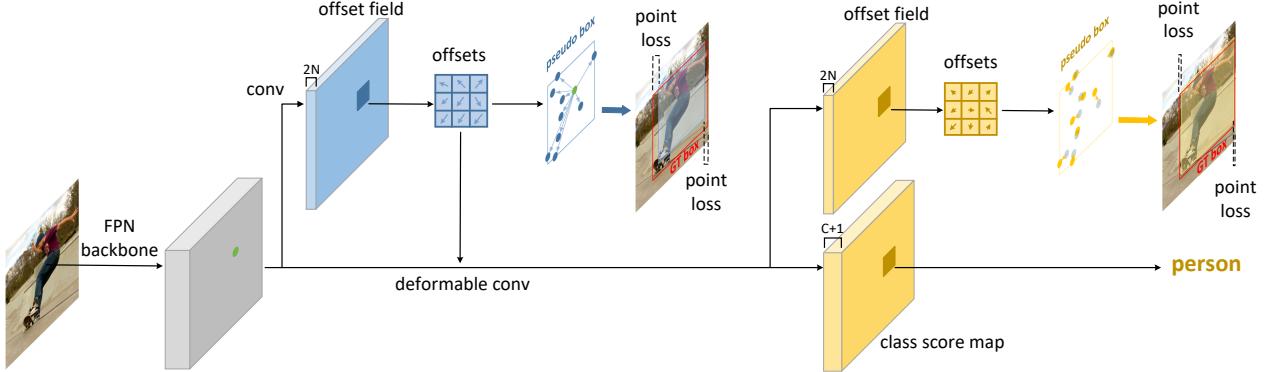


Figure 2. Overview of the proposed RPDet (RepPoints detector). While feature pyramidal networks (FPN) [27] are adopted as the backbone, we only draw the afterwards pipeline of one scale of FPN feature maps for clear illustration. Note all scales of FPN feature maps share the same afterwards network architecture and the same model weights.

tion evolves as follows:

$$\begin{array}{l}
 \text{object centers} \xrightarrow{\text{RP refine}} \text{RepPoints proposals (S1)} \\
 \xrightarrow{\text{RP refine}} \text{RepPoints proposals (S2)} \\
 \dots \\
 \xrightarrow{\text{RP refine}} \text{RepPoints object targets}
 \end{array} \quad (6)$$

Our RepPoints Detector (RPDet) is constructed with two recognition stages based on deformable convolution, as illustrated in Figure 2. Deformable convolution pairs nicely with RepPoints, as its convolutions are computed on an irregularly distributed set of sample points and conversely its recognition feedback can guide training for the positioning of these points. In this section, we present the design of RPDet and discuss its relationship to and differences from existing object detectors.

**Center point based initial object representation.** While predefined anchors dominate the representation of objects in the initial stage of object detection, we follow YOLO [35] and DenseBox [18] by using center points as the initial representation of objects, which leads to an anchor-free object detector.

An important benefit of the center point representation lies in its much tighter hypothesis space compared to the anchor based counterparts. While anchor based approaches usually rely on a large number of multi-ratio and multi-scale anchors to ensure dense coverage of the large 4-d bounding box hypothesis space, a center point based approach can more easily cover its 2-d space. In fact, all objects will have center points located within the image.

Center point based methods also face problems that limit its prevalence in modern object detectors. One is that two different objects may be located at the same position in a feature map, resulting in ambiguity of the recognition targets. In previous methods [35], this is mainly addressed by producing multiple targets at each position, which faces

another issue of vesting ambiguity<sup>1</sup>. In RPDet, we show that this issue can be greatly alleviated by using the FPN structure [27] for the following reasons: first, objects of different scales will be assigned to different image feature levels, which addresses objects of different scales and the same center points locations; second, FPN has a high-resolution feature map for small objects, which also reduces the chance of two objects having centers located at the same feature position. In fact, we observe that only 1.1% of objects in the COCO datasets [29] suffer from the issue of center points located at the same position when FPN is used.

Another issue is that it is hard for center point based methods to predict accurate object localization, due to the large variation in the spatial scope of objects. This issue can also be alleviated in our framework, through the flexibility of the *RepPoints* representation and the effective refinement of *RepPoints* in Eq.(6). In our experiments, we demonstrate that our proposed anchor-free object detector can be as effective as the state-of-the-art anchor-based methods.

It is worth noting that the center point representation can be viewed as a special RepPoints configuration, where only a single fixed sample point is used, thus maintaining a coherent representation throughout the proposed detection framework.

**Utilization of RepPoints.** As shown in Figure 2, RepPoints serve as the basic object representation throughout our detection system. Starting from the center points, the first set of *RepPoints* is obtained via a  $3 \times 3$  regular convolution [51]. The learning of these *RepPoints* is driven by two objectives: 1) the top-left and bottom-right points distance loss between the induced *pseudo box* and the ground-truth bounding box; 2) the object recognition loss of the subsequent stage. As illustrated in Figure 3, extreme and key points are automatically learned. The second set of *Rep-*

<sup>1</sup>If the center points of multiple ground truth objects are located at a same feature map position, only one randomly chosen ground truth object is assigned to be the target of this position.

*Points* represents the final object localization, which is refined from the first set of *RepPoints* by Eq. (5). Driven by the points distance loss alone, this second set of *RepPoints* aims to learn finer object localization.

*Relation to deformable RoI pooling [4].* As mentioned in Section 2, deformable RoI pooling plays a different role in object detection compared to the proposed *RepPoints*. Basically, *RepPoints* is a geometric representation of objects, reflecting more accurate semantic localization, while deformable RoI pooling is geared towards learning stronger appearance features of objects.

In fact, deformable RoI pooling cannot learn sample points representing accurate localization of objects. This can be seen from the following contradiction. If the deformable RoI pooling method indeed learns an accurate geometric representation of objects, the deformable RoI pooling layer would produce the same appearance features for two nearby proposals of the same object. In this case, the object detectors would fail, due to the inability to differentiate these two proposals. However, the deformable RoI pooling method has been shown to well differentiate two nearby proposals [4]. This shows that the deformable RoI pooling cannot learn accurate localization of objects. Please see the appendix for a more detailed discussion.

We also note that deformable RoI pooling can be complementary to RepPoints, as indicated in Table 6.

**Other details.** Our FPN structures follow [28] to produce 5 feature pyramid levels from stage 3 (downsampling ratio of 8) to stage 7 (downsampling ratio of 128). Specifically, for a box  $B$  with width  $w_B$  and height  $h_B$ , it corresponds to pyramid level  $lvl(B)$ , where  $lvl(B) = \lfloor \log_2(\sqrt{w_B h_B}) \rfloor$ . For each ground-truth box  $B_t$ , we project its center point to each feature level  $l$  in the pyramid ( $l = 3, 4, 5, 6, 7$ ), where the projected bin is denoted as  $q_l$ . Then, we set  $q_{lvl(B_t)}$  to be a positive bin and  $q_{lvl(B_t)-1}$  and  $q_{lvl(B_t)+1}$  as ignored bins<sup>2</sup>. All other unset bins are assigned negative labels.

## 5. Experiments

### 5.1. Implementation Details

We present experimental results of our proposed RPDet framework on the MS-COCO [29] detection benchmark, which contains 118k images for training, 5k images for validation (minival) and 20k images for testing (test-dev). All the ablation studies are conducted on minival with ResNet-50 [16], if not otherwise specified. The state-of-the-art comparison is reported on test-dev in Table 7. The code will be made available.

<sup>2</sup>If  $lvl(B_t) - 1$  ( $lvl(B_t) + 1$ ) is not in  $\{3, 4, 5, 6, 7\}$ , i.e.  $lvl(B_t)$  equals to 3 (7), we just assign one ignored label to  $lvl(B_t) + 1$  ( $lvl(B_t) - 1$ ).

Representation	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>
Bounding box	ResNet-50	36.2	57.3	39.8
RepPoints (ours)	ResNet-50	<b>38.3</b>	<b>60.0</b>	<b>41.1</b>
Bounding box	ResNet-101	38.4	59.9	42.4
RepPoints (ours)	ResNet-101	<b>40.4</b>	<b>62.0</b>	<b>43.6</b>

Table 1. Comparison of the RepPoints and bounding box representations in object detection. The network structures are the same except for processing the given object representation.

Representation	Supervision		AP	AP <sub>50</sub>	AP <sub>75</sub>
	loc.	rec.			
bounding box	✓		36.2	57.3	39.8
	✓	✓	36.2	57.5	39.8
RepPoints		✓	33.8	54.3	35.8
	✓		37.6	59.4	40.4
	✓	✓	<b>38.3</b>	<b>60.0</b>	<b>41.1</b>

Table 2. Ablation of the supervision sources, for both bounding box and RepPoints based object detection. “loc.” indicates the object localization loss. “rec.” indicates the object recognition loss from the next detection stage.

Method	AP	AP <sub>50</sub>	AP <sub>75</sub>
Single anchor	36.9	58.2	39.7
Center point	<b>38.3</b>	<b>60.0</b>	<b>41.1</b>

Table 3. Comparison of using a single anchor and a center point as the initial object representation. For the single anchor based method, the objectness assignment of an initial instance is determined by the IoU (intersection-of-union) with the ground-truth bounding box (larger than 0.5 or being the maximal IoU to a ground-truth bounding box). For the center point based method, the objectness assignment of an initial instance (each position on a feature map) is determined by whether there exists a ground-truth bounding box with the center located at this position. All other settings are exactly the same.

Our detector is trained with synchronized stochastic gradient descent (SGD) over 4 GPUs with a total of 8 images per minibatch (2 images per GPU). The ImageNet [6] pre-trained model was used for initialization. Our learning rate schedule follows the ‘1x’ setting for RetinaNet [28] in [13]. We use GN [48] and focal loss [28] to facilitate training. For our model design, we replace the last  $3 \times 3$  convolution layer in [28] with an offset-constrained deformable convolution layer, followed by a 1024-d  $1 \times 1$  convolution layer. For data augmentation, we use only horizontal image flipping. In inference, NMS ( $\sigma = 0.5$ ) is employed to post-process the results, following [28].

### 5.2. Ablation Study

**RepPoints vs. bounding box.** To demonstrate the effectiveness of the proposed *RepPoints*, we compare the pro-



Figure 3. Visualization of the learned RepPoints and the corresponding detection results on several examples from the COCO [29] minival set (using pseudo box converting function of  $\mathcal{T}_1$ ). In general, the learned RepPoints are located on extreme or semantic keypoints of the objects.

method	backbone	# anchors per scale	AP
RetinaNet [28]	ResNet-50	$3 \times 3$	35.7
FPN-RoIAlign [27]	ResNet-50	$3 \times 1$	36.7
YOLO-like	ResNet-50	-	33.9
RPDet (ours)	ResNet-50	-	<b>38.3</b>
RetinaNet [28]	ResNet-101	$3 \times 3$	37.8
FPN-RoIAlign [27]	ResNet-101	$3 \times 1$	39.4
YOLO-like	ResNet-101	-	36.3
RPDet (ours)	ResNet-101	-	<b>40.4</b>

Table 4. Comparison of the proposed method (RPDet) with an anchor-based method (RetinaNet, FPN-RoIAlign) and an anchor-free method (YOLO-like). The YOLO-like method is adapted from the YOLOv1 method [35] by additionally introducing FPN [27], GN [48] and focal loss [28] into the method for better accuracy.

posed RPDet to a version where RepPoints are all replaced by the regular bounding box representation. Specifically, the bounding box representation based version uses a single anchor as the initial object representation (the width and height of this anchor is  $4 \times$  of the feature map stride). The intersection-over-union (IoU) criterion is used to assign classification labels to each bounding box. The bounding box regression method in Section 3.1 is used to obtain the first stage’s bounding box proposals and the final bounding box targets. The  $3 \times 3$  RoIAlign [14] operator is adopted to extract object features of the second stage. All other settings are the same as in the proposed RPDet method.

Table 1 shows that the change of object representation from bounding box to RepPoints can bring a +2.1 mAP improvement using ResNet-50 [16] and a +2.0 mAP improvement using a ResNet-101 [16] backbone, showing the advantage of the RepPoints representation over bounding boxes for object detection.

pseudo box converting function	AP	AP <sub>50</sub>	AP <sub>75</sub>
$\mathcal{T} = \mathcal{T}_1$ : min-max	38.2	59.7	40.7
$\mathcal{T} = \mathcal{T}_2$ : partial min-max	38.1	59.6	40.5
$\mathcal{T} = \mathcal{T}_3$ : moment-based	38.3	60.0	41.1

Table 5. Comparison of different transformation functions from RepPoints to pseudo box,  $\mathcal{T}$ .

representation method	w. dpool	AP	AP <sub>50</sub>	AP <sub>75</sub>
bounding box		36.2	57.3	39.8
	✓	36.9	58.0	41.0
RepPoints		38.3	60.0	41.1
	✓	<b>39.1</b>	<b>60.6</b>	<b>42.4</b>

Table 6. The effect of applying the deformable RoIPooling layer [4] on the proposals of the first stages (see Eq. (1) and Eq. (6)). The deformable RoIPooling layer can boost both the methods using bounding boxes and RepPoints, respectively.

**Supervision source for RepPoints learning.** RPDet uses both an object localization loss and an object recognition loss to drive the learning of the first set of RepPoints, which represents the object proposals of the first stage. Table 2 ablates the use of these two supervision sources in the learning of the object representations. As mentioned before, describing the geometric localization of objects is an important duty of a representation method. Without the object localization loss, it is hard for a representation method to accomplish this duty, as it results in significant performance degradation of the object detectors. For RepPoints, we observe a huge drop of 4.5 mAP by removing the object localization supervision, showing the importance of describing the geometric localization for an object representation method.

Table 2 also demonstrates the benefit of including the ob-

	Backbone	Anchor-Free	<i>AP</i>	<i>AP<sub>50</sub></i>	<i>AP<sub>75</sub></i>	<i>AP<sub>S</sub></i>	<i>AP<sub>M</sub></i>	<i>AP<sub>L</sub></i>
YOLOv2 [36]	DarkNet-19		21.6	44.0	19.2	5.0	22.4	35.5
SSD [31]	ResNet-101		31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [37]	DarkNet-53		33.0	57.9	34.4	18.3	35.4	41.9
DSSD [10]	ResNet-101		33.2	53.3	35.2	13.0	35.4	51.1
Faster R-CNN w. FPN [27]	ResNet-101		36.2	59.1	39.0	18.2	39.0	48.2
RefineDet [52]	ResNet-101		36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [28]	ResNet-101		39.1	59.1	42.3	21.8	42.7	50.2
Deep Regionlets [49]	ResNet-101		39.3	59.8	-	21.7	43.7	50.9
Mask R-CNN [14]	ResNeXt-101		39.8	62.3	43.4	22.1	43.2	51.2
FSAF [56]	ResNet-101		40.9	61.5	44.0	24.0	44.2	51.3
LH R-CNN [26]	ResNet-101		41.5	-	-	25.2	45.3	53.1
Cascade R-CNN [2]	ResNet-101		42.8	62.1	46.3	23.7	45.5	55.2
CornerNet [24]	Hourglass-104	✓	40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet [54]	Hourglass-104	✓	40.1	55.3	43.2	20.3	43.2	53.1
<b>RPDet</b>	ResNet-101	✓	41.0	62.9	44.3	23.6	44.1	51.7
<b>RPDet</b>	ResNet-101-DCN	✓	<b>42.8</b>	<b>65.0</b>	<b>46.3</b>	<b>24.9</b>	<b>46.2</b>	<b>54.7</b>

Table 7. Comparison the proposed RPDet to the state-of-the-art detectors on COCO [29] test-dev. Without multi-scale training and testing, our proposed framework achieves 42.8 AP with ResNet-101-DCN backbone [16, 4], which is on-par with 4-stage anchor-based Cascade R-CNN [2] method and outperforms all existing anchor free detectors. Moreover, RPDet obtains an *AP<sub>50</sub>* of 65.0, surpassing all baselines by a significant margin.

ject recognition loss in learning RepPoints (+0.7 mAP). The use of the object recognition loss can drive the RepPoints to locate themselves at semantically meaningful positions on an object, which leads to fine-grained localization and improves object feature extraction for the following recognition stage. Note that the object recognition loss benefit object detection with the bounding box representation (see the first block in Table 2), further demonstrating the advantage of RepPoints in flexible object representation.

**Anchor-free vs. anchor-based.** We first compare the center point based method (a special RepPoints configuration) and the prevalent anchor based method in representing initial object hypotheses, in Table 3. The center point based method surpasses the anchor based method by +1.4 mAP, likely because of its better coverage of ground-truth objects.

We also compare the proposed anchor-free detector based on RepPoints to RetinaNet [28] (a popular one-stage anchor-based method), FPN [27] with RoIAlign (a popular two-stage anchor-based method) [14], and a YOLO-like detector which is adapted from the anchor-free method of YOLOv1 [35], in Table 4. The proposed method outperforms both RetinaNet [28] and the FPN [27] methods, which utilize multiple anchors per scale and sophisticated anchor configurations (FPN). The proposed method also significantly surpasses another anchor-free method (the YOLO-like detector) by +4.4 mAP and +4.1 mAP, respectively, probably due to the flexible RepPoints representation and its effective refinement.

**Converting RepPoints to pseudo box.** Table 5 shows that different instantiations of the transformation functions  $\mathcal{T}$  presented in Section 3.2 work comparably well.

**RepPoints act complementary to deformable RoI pooling [4].** Table 6 shows the effect of applying the deformable RoI pooling layer [4] to both bounding box proposals and RepPoints proposals. While applying the deformable RoI pooling layer to bounding box proposals brings a +0.7 mAP gain, applying it to RepPoints proposals also brings a +0.7 mAP gain, implying that the roles of deformable RoI pooling and the proposed RepPoints are complementary.

### 5.3. RepPoints Visualization

In Figure 3, we visualize the learned *RepPoints* and the corresponding detection results on several examples from the COCO [29] minival set. It can be observed that *RepPoints* tend to be located at extreme points or key semantic points of objects. These point distributed over objects are automatically learned without explicit supervision. The visualized results also indicate that the proposed RPDet, implemented here with the min-max transformation function, can effectively detect tiny objects.

### 5.4. State-of-the-art Comparison

We compare RPDet to state-of-the-art detectors on test-dev. Table 7 shows the results. Without any bells and whistles, RPDet achieves 42.8 AP on COCO bench-

mark [29], which is on-par with 4-stage anchor-based Cascade R-CNN [2] and outperforms all existing anchor-free approaches.

Moreover, without multi-scale training and testing, our clean anchor-free design achieves  $65.0 AP_{50}$ <sup>3</sup>, which surpasses all baselines by a significant margin ( $\sim 3.0$ ). This improvement mostly comes from its performance on small objects ( $AP_S$ ). These observations are consistent with the findings in YOLOv3 [37] on the advantage of using center point based label assignment.

## 6. Conclusion

In this paper, we propose *RepPoints*, a representation for object detection that models fine-grained localization information and identifies local areas significant for object classification. Based on *RepPoints*, we develop an object detector called RPDet that achieves competitive object detection performance without the need of anchors. Learning richer and more natural object representations like *RepPoints* is a direction that holds much promise for object detection.

## References

- [1] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, pages 168–181. Springer, 2010. 3
- [2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2, 3, 8, 9
- [3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1
- [4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 2, 3, 6, 7, 8, 11, 12
- [5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE Computer Society, 2005. 2
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 6
- [7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2147–2154, 2014. 1
- [8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2
- [9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 3
- [10] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 8
- [11] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 2, 3
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1, 2, 11
- [13] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. <https://github.com/facebookresearch/detectron>, 2018. 6
- [14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 3, 7, 8, 11
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *PAMI*, 37(9):1904–1916, 2015. 1
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 6, 7, 8
- [17] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *PAMI*, 29(4):671–686, 2007. 3
- [18] L. Huang, Y. Yang, Y. Deng, and Y. Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015. 5
- [19] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 3
- [20] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, pages 784–799, 2018. 2
- [21] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi. Consistent optimization for single-shot object detection. *arXiv preprint arXiv:1901.06563*, 2019. 2
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2
- [23] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 2
- [24] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 2, 3, 8
- [25] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume 1, pages 878–885. IEEE, 2005. 3
- [26] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. 2, 8
- [27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *ICCV*, pages 2117–2125, 2017. 2, 5, 7, 8
- [28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2, 3, 6, 7, 8
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 5, 6, 7, 8, 9

<sup>3</sup> $AP_{50}$  is believed to be a better metric in [37].

- [30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 2
- [31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 8
- [32] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, pages 2277–2287, 2017. 3
- [33] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, pages 4930–4939, 2017. 3
- [34] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015. 1
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 5, 7, 8
- [36] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 8
- [37] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 8, 9
- [38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 3, 11
- [39] H. Rezatofighi, N. Tsai, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 2
- [40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2
- [41] B. Singh and L. S. Davis. An analysis of scale invariance in object detection snip. In *CVPR*, pages 3578–3587, 2018. 2
- [42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2
- [43] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 1
- [44] L. Tychsen-Smith and L. Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *ICCV*, pages 428–436, 2017. 2
- [45] P. Viola, M. Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:511–518, 2001. 2
- [46] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. In *CVPR*, 2019. 2
- [47] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum. Learning to see physics via visual de-animation. In *NIPS*, pages 153–164, 2017. 1
- [48] Y. Wu and K. He. Group normalization. In *ECCV*, pages 3–19, 2018. 6, 7
- [49] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa. Deep regionlets for object detection. In *ECCV*, pages 798–814, 2018. 8
- [50] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun. Metaanchor: Learning to detect objects with customized anchors. In *NeurIPS*, pages 318–328, 2018. 2
- [51] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 5
- [52] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018. 8
- [53] Y. Zhong, J. Wang, J. Peng, and L. Zhang. Anchor box optimization for object detection. *arXiv preprint arXiv:1812.00469*, 2018. 2
- [54] X. Zhou, J. Zhuo, and P. Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2, 3, 8
- [55] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao. Oriented response networks. In *CVPR*, pages 519–528, 2017. 3
- [56] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019. 8

## Appendix

### A1. Relationship between RepPoints and Deformable RoI pooling

In this section, we explain the differences between our method and deformable RoI pooling [4] in greater detail. We first describe the translation sensitivity of the regression step in the object detection pipeline. Then, we discuss how deformable RoI pooling [4] works and why it does not provide a geometric representation of objects, unlike the proposed RepPoints representation.

**Translation Sensitivity** We explain the translation sensitivity of the regression step in the context of bounding boxes. Denote a rectangular bounding box proposal before regression as  $\mathcal{B}_P$  and the ground-truth bounding box as  $\mathcal{B}_{GT}$ . The target for bounding box regression can then be expressed as

$$T_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}), \quad (7)$$

where  $\mathcal{F}$  is a function for transforming  $\mathcal{B}_P$  to  $\mathcal{B}_{GT}$ . This transformation is conventionally learned as a regression function  $\mathcal{R}_B$ :

$$\mathcal{R}_B(\mathcal{P}_B(I, \mathcal{B}_P)) = T_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}), \quad (8)$$

where  $I$  is the input image and  $\mathcal{P}_B$  is a pooling function defined over the rectangular proposal, e.g., direct cropping of the image [12], RoIPooling [38], or RoIAlign [14]. This formulation aims to predict the relative displacement to the ground truth box based on features within the area of  $\mathcal{B}_P$ . Shifts in  $\mathcal{B}_P$  should change the target accordingly:

$$\mathcal{R}_B(\mathcal{P}_B(I, \mathcal{B}_P + \Delta\mathcal{B})) = \mathcal{F}(\mathcal{B}_P + \Delta\mathcal{B}, \mathcal{B}_{GT}). \quad (9)$$

Thus, the pooled feature  $\mathcal{P}_B(I, \mathcal{B}_P)$  should be sensitive to the box proposal  $\mathcal{B}_P$ . Specifically, for any pair of proposals  $\mathcal{B}_1 \neq \mathcal{B}_2$ , we should have  $\mathcal{P}_B(I, \mathcal{B}_1) \neq \mathcal{P}_B(I, \mathcal{B}_2)$ . Most existing feature extractors  $\mathcal{P}_B$  satisfy this property. Note that the improvement of RoIAlign [14] over RoIPooling [38] is partly due to this guaranteed translation sensitivity.

**Analysis of Deformable RoI Pooling.** For deformable RoI pooling [4], the system generates a pointwise deformation of samples on a regular grid [14] to produce a set of sample points  $S_P$  for each proposal. This can be formulated as

$$S_P = \mathcal{D}(I, \mathcal{B}_P), \quad (10)$$

where  $\mathcal{D}$  is the function for generating the sample points. Then, bounding box regression aims to learn a regression function  $\mathcal{R}_S$  which utilizes the sampled features via  $S_P$  to predict the target  $T_P$  as follows:

$$\mathcal{R}_S(\mathcal{P}_S(I, S_P)) = T_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}) \quad (11)$$

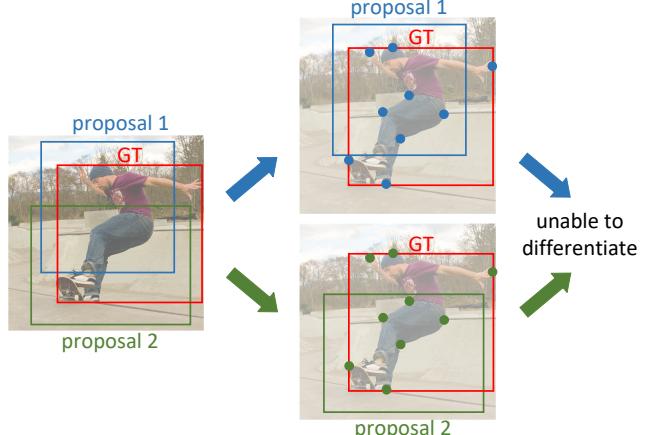


Figure 4. Illustration that deformable RoI pooling [4] is unable to serve as a geometric object representation, as discussed in Section 4. We consider two bounding box regressions based on different proposals. Assume that deformable RoI pooling [4] can learn a similar geometric object representation where the two sets of sample points lie at similar locations over the object of interest. For that to happen, the sampled features would need to be similar, such that the two proposals cannot be differentiated. However, deformable RoI pooling [4] can indeed differentiate nearby object proposals, leading to a contradiction. Thus, it is concluded that deformable RoI pooling [4] cannot learn the geometric representation of objects.

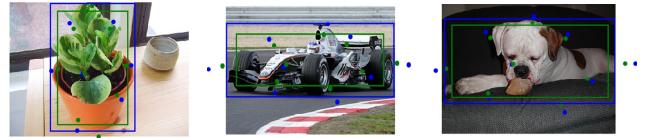


Figure 5. Visualization of the learned sample points of  $3 \times 3$  deformable RoI pooling [4]. It is shown that the scale of the sample points changes as the scale of the proposal changes, indicating that the sample points do not adapt to form a geometric *object* representation.

where  $\mathcal{P}_S$  is the pooling function with respect to the sample points  $S_P$ .

From the translation sensitivity property, we have  $\mathcal{P}_S(I, \mathcal{D}(I, \mathcal{B}_1)) \neq \mathcal{P}_S(I, \mathcal{D}(I, \mathcal{B}_2)), \forall \mathcal{B}_1 \neq \mathcal{B}_2$ . Because the pooled feature  $\mathcal{P}_S(I, \mathcal{D}(I, \mathcal{B}))$  is determined by the locations of sample points  $\mathcal{D}(I, \mathcal{B})$ , we have  $\mathcal{D}(I, \mathcal{B}_1) \neq \mathcal{D}(I, \mathcal{B}_2), \forall \mathcal{B}_1 \neq \mathcal{B}_2$ . This means that for two different proposals  $\mathcal{B}_1$  and  $\mathcal{B}_2$  of the same object, the sample points of these two proposals by deformable RoI pooling should be different. Hence, the sample points of different proposals cannot correspond to the geometry of the same object. They represent a property of the proposals rather than the geometry of the object.

Figure 4 illustrates the contradiction that arises if deformable RoI pooling were a representation of object geometry. Moreover, Figure 5 illustrates that, for the learned

sample points of two proposals for the same object by deformable RoI pooling, the sample points represent a property of the proposals instead of the geometry of the object.

**RepPoints** In contrast to deformable RoI pooling where the pooled features represent the original bounding box proposals, the features extracted from RepPoints localize the object. As it is not restricted by translation sensitivity requirements, RepPoints can learn a geometric representation of *objects* when localization supervision on the corresponding pseudo box is provided (see Figure 3). While object localization supervision is not applied on the sample points of deformable RoI pooling, we show in Table 2 that such supervision is crucial for RepPoints.

It is worth noting that deformable RoI pooling [4] is shown to be complementary to the RepPoints representation (see Table 6), further indicating their different functionality.