首先根据物体中心回归9个点，然后用9个点计算出box，使用box做监督信息。

与anchor base的区别其实不大，一个是用的anchor来回归，一个是用的中心点来回归，本质上都是使用了一个初始位置点。增益可能比较多的来自于，中心点相对于anchor更能表示物体？由于有更多的语义信息？

RepPoints: Point Set Representation for Object Detection

Ze Yang[1†*] Shaohui Liu[2,3†*] Han Hu[3] Liwei Wang[1] Stephen Lin[3]
[1]Peking University [2]Tsinghua University
[3]Microsoft Research Asia
yangze@pku.edu.cn, b1ueber2y@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 can be as effective as the state-of-the-art anchor-based detection methods, with 46.5 AP and 67.4 AP_{50} on the COCO test-dev detection benchmark, using ResNet-101 model. Code is available at https://github.com/microsoft/RepPoints.*

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 [30], human pose analysis [37], and visual reasoning [41]. 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, 11, 10, 33, 14, 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 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.

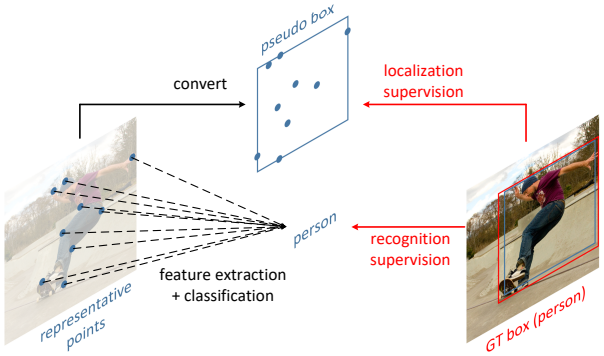

Figure 1. *RepPoints* is a new representation for object detection that consists of a set of points which indicate the spatial extent of an object and semantically significant local areas. This representation is learned via weak localization supervision from rectangular ground-truth boxes and implicit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

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

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

RepPoints, that provides more fine-grained localization and facilitates classification. Illustrated in Fig. 1, RepPoints is a set of points that learns 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 [38, 21, 48]. 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 [38, 21] or require ground truth object masks as additional supervision [48]. 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.

With RepPoints replacing all the conventional bounding box representations in a two-stage object detector, including anchors, proposals and final localization targets, we develop a clean and effective *anchor-free* object detector, which achieves 42.8 AP and 65.0 AP_{50} on the COCO benchmark [26] without multi-scale training and testing, and achieves 46.5 AP and 67.4 AP_{50} with multi-scale training and testing using ResNet-101 model. The proposed object detector not only surpasses all existing anchor-free detectors but also performing on-par with state-of-the-art anchor-based baselines. 之前的版本中有说：anchor free对分类更好，不知道这个结论是不是被推翻了，这边没说

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, 26, 20]. 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 [39, 5] and during the deep learning era [19, 34, 36, 15], are based on an input patch with a regular grid form. It is thus convenient to use the bounding box representation to facilitate feature extraction [11, 10, 33].

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 [24, 4, 13, 23, 2, 35, 27], and the bounding box representation appears in almost all stages: 1) as pre-defined [33, 25] or learnt [40, 45, 47] anchors that serve as hypotheses over the bounding box space; 2) as refined object proposals connecting successive recognition stages [33, 2]; and 3) as the final localization targets.

The RepPoints can be used to replace the bounding box representations in all stages of model object detectors, resulting in a more effective new object detector. Specifically, the anchors are replaced by center points, which is a special configuration of RepPoints. The bounding box proposals and final localization targets are replaced by the RepPoints proposals and final targets. Note the resulting object detector is anchor-free, due to the use of center points for initial object representation. It is thus even more convenient in use than the bounding box based methods.

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 [22] and a rotated bounding box to better handle rotational variations [16, 49].

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 [21, 48]. CornerNet [21] first predicts top-left and bottom-right corners and then employs a specialized grouping method [28] to obtain the bounding boxes of objects. However, the two opposing corner points still essentially model a rectangular bounding box. ExtremeNet [48] is proposed to locate the extreme points of objects in the x- and y-directions [29] 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 ob-

jects 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 [48] 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 [18] 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 and appendix. In this sense, we expand the usage of adaptive sample points in previous geometric modeling methods [18, 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:

$$\begin{aligned} \text{bbox anchors} &\xrightarrow{\text{bbox reg.}} \text{bbox proposals (S1)} \\ &\xrightarrow{\text{bbox reg.}} \text{bbox proposals (S2)} \\ &\ldots \\ &\xrightarrow{\text{bbox reg.}} \text{bbox object targets} \end{aligned} \tag{1}$$

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 [25].

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 [10] or RoI-Align [13]. For the two-stage framework [33], 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

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

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:

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

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

5×9
[32,64,128,256,512]
[0.5,1,2]*3

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:

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

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

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

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 predefined 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:

极值点表示框

- $\mathcal{T} = \mathcal{T}_1$**: 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.

9个点的子集

- $\mathcal{T} = \mathcal{T}_2$**: Partial min-max function.** Min-max operation over a subset of the sample points is performed over both axes to obtain the rectangular box $\mathcal{B}_p$.

9个点的均值表示中心点

- $\mathcal{T} = \mathcal{T}_3$**: Moment-based function.** The mean value and the standard deviation 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 λ_x and λ_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 4 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 representation evolves as follows:

$$\begin{aligned} \text{object centers} &\xrightarrow{\text{RP refine}} \text{RepPoints proposals (S1)} \\ &\xrightarrow{\text{RP refine}} \text{RepPoints proposals (S2)} \\ &\cdots \\ &\xrightarrow{\text{RP refine}} \text{RepPoints object targets} \end{aligned} \tag{6}$$

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 [31] and DenseBox [17] 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

选择物体中心作为初始点，用deformable 预测出9个点
将9个点转换为bbox，用bbox来计算loss

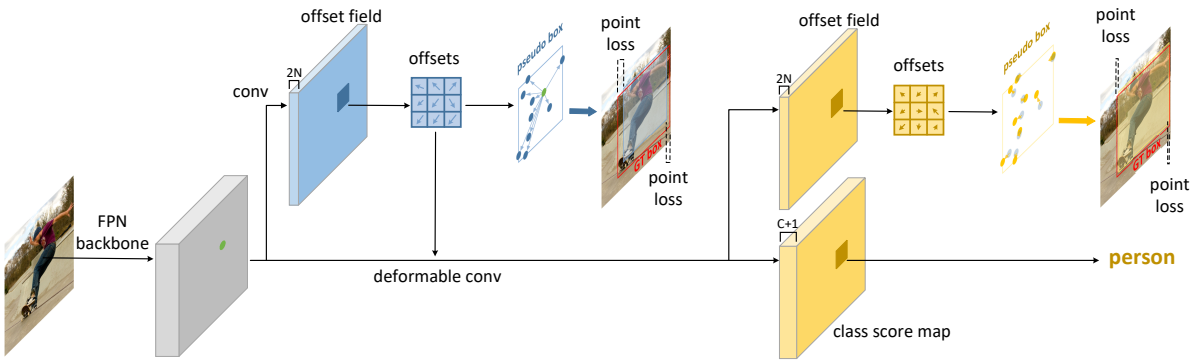

Figure 2. Overview of the proposed RPDet (RepPoints detector). While feature pyramidal networks (FPN) [24] are adopted as the backbone, we only draw the afterwards pipeline of one scale of FPN feature maps for clear illustration. Note all scales of FPN feature maps share the same afterwards network architecture and the same model weights.

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.

However, the center point based method also faces the problem of recognition target ambiguity, caused by two different objects locating at the same position in a feature map, which limits its prevalence in modern object detectors. In previous methods [31], this is mainly addressed by producing multiple targets at each position, which faces another issue of vesting ambiguity[1]. In RPDet, we show that this issue can be greatly alleviated by using the FPN structure [24] 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 [26] suffer from the issue of center points located at the same position when FPN is used.

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 regressing offsets over the center points. 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 4, extreme

[1]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.

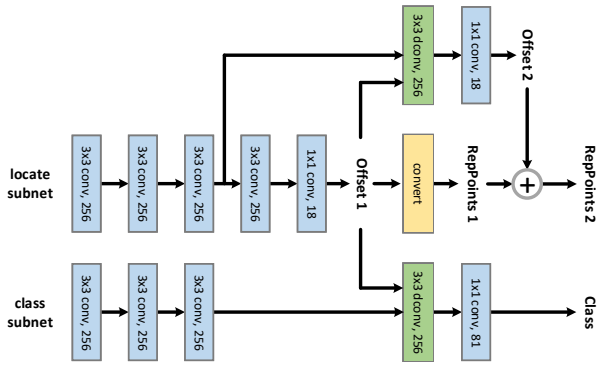

Figure 3. The head architecture of RPDet.

and key points are automatically learned. The second set of *RepPoints* 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 (please see the appendix for a proof).

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

Backbone and head architectures Our FPN backbone follows [25], which produces 5 feature pyramid levels from stage 3 (downsampling ratio of 8) to stage 7 (downsampling ratio of 128).

The head architecture is illustrated in Figure 3. There are two *non-shared* subnets, aiming at localization (RepPoints generation) and classification, respectively. The localization subnet first apply three 256-d 3×3 conv layers, fol-

lowed by two successive small networks to compute offsets for the two sets of *RepPoints*. The classification subnetwork also apply three 256-d 3×3 conv layers, followed by a 256-d 3×3 deformable conv layer with its input offset field shared with that of the first deformable conv layer in the localization subnetwork. The group normalization layer is applied after each of the first three 256-d 3×3 conv layers in both subnets.

Note although our approach utilizes two stages of localization, it is even more efficient than the one stage RetinaNet [25] detector (210.9 vs. 234.5 GFLOPS using ResNet-50). The additional localization stage introduces little overhead due to layer sharing. The anchor-free design reduces the burden of the final classification layer which leads to a slight reduction in computation.

Localization/class target assignment. There are two localization stages: generating the first set of RepPoints by refining from the object center point hypothesis (feature map bins); generating the second set of RepPoints by refining from the first RepPoints set. For both stages, only *positive* object hypothesis are assigned with localization (RepPoints) targets in training. For the first localization stage, a feature map bin is *positive* if 1) the pyramidal level of this feature map bin equals the log scale of a ground-truth object $s(B) = \lfloor log_2(\sqrt{w_B h_B}/4) \rfloor$; 2) the projection of this ground-truth object's center point locate within this feature map bin. For the second localization stage, the first RepPoints is *positive* if its induced pseudo box has sufficient overlap with a ground-truth object that their intersection-over-union is larger than 0.5.

Classification is conducted on the first set of RepPoints only. The classification assignment criterion follows [25] that IoU (between the induced pseudo box and ground-truth bounding box) larger than 0.5 indicates *positive*, smaller than 0.4 indicates background, and otherwise ignored. Focal loss is employed for classification training [25].

5. Experiments

5.1. Experimental Settings

We present experimental results of our proposed RPDet framework on the MS-COCO [26] 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 [15], if not otherwise specified. The state-of-the-art comparison is reported on `test-dev` in Table 7.

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 [12]. Random horizontal image flipping is adopted in training. In inference, NMS ($\sigma = 0.5$) is employed to post-process the results, following [25].

Representation	Backbone	AP	AP_{50}	AP_{75}
Bounding box	ResNet-50	36.2	57.3	39.8
RepPoints (ours)	ResNet-50	**38.3**	**60.0**	**41.1**
Bounding box	ResNet-101	38.4	59.9	42.4
RepPoints (ours)	ResNet-101	**40.4**	**62.0**	**43.6**

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_{50}	AP_{75}
	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
	✓	✓	**38.3**	**60.0**	**41.1**

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.

other repr.	init repr.	AP	AP_{50}	AP_{75}
bounding box	single anchor	36.2	57.3	39.8
	center point	**37.3**	**58.0**	**40.0**
RepPoints	single anchor	36.9	58.2	39.7
	center point	**38.3**	**60.0**	**41.1**

Table 3. Comparison of using a single anchor and a center point as the initial object representation ("init repr."). "other repr." indicates representation method for object proposals and final targets.

method	backbone	# anchors per scale	AP
RetinaNet [25]	ResNet-50	3×3	35.7
FPN-RoIAlign [24]	ResNet-50	3×1	36.7
YOLO-like	ResNet-50	-	33.9
RPDet (ours)	ResNet-50	-	**38.3**
RetinaNet [25]	ResNet-101	3×3	37.8
FPN-RoIAlign [24]	ResNet-101	3×1	39.4
YOLO-like	ResNet-101	-	36.3
RPDet (ours)	ResNet-101	-	**40.4**

Table 4. Comparison of the proposed method (RPDet) with anchor-based methods (RetinaNet, FPN-RoIAlign) and an anchor-free method (YOLO-like). The YOLO-like method is adapted from the YOLOv1 method [31] by additionally introducing FPN [24], GN [42] and focal loss [25] for better accuracy.

Figure 4. Visualization of the learnt RepPoints and the induced bounding boxes on several examples from the COCO [26] minival set (using the pseudo box converting function $\mathcal{T}_1$). In general, the learned RepPoints are located on extreme or semantic keypoints of objects.

pseudo box converting function	AP	AP_{50}	AP_{75}
$\mathcal{T} = \mathcal{T}_1$: 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_{50}	AP_{75}
bounding box		36.2	57.3	39.8
	✓	36.9	58.0	41.0
RepPoints		38.3	60.0	41.1
	✓	**39.1**	**60.6**	**42.4**

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

5.2. Ablation Study

RepPoints vs. bounding box. To demonstrate the effectiveness of the proposed *RepPoints*, we compare the proposed RPDet to a baseline detector where RepPoints are all replaced by the regular bounding box representation.

Baseline detector based on bounding box representations. A single anchor with scale of 4 and aspect ratio of $1:1$ is adopted for the initial object representation, where the anchor box is *positive* if the IoU with a ground-truth object is larger than 0.5. The two sets of RepPoints are replaced by bounding box representation, where the geometric refinement is achieved by the standard bounding box regression method, and the feature extraction is replaced by the RoIAlign [13] method using 3×3 grid points[2]. All other settings are the same as in the proposed RPDet method.

Table 1 shows the comparison of two detectors. While the bounding box based method achieves 36.2 mAP, indicating a strong baseline detector, the change of object representation from bounding box to RepPoints brings a +2.1 mAP improvement using ResNet-50 [15] and a +2.0 mAP improvement using a ResNet-101 [15] backbone, demonstrating the advantage of the RepPoints representation over bounding boxes for object detection.

Supervision source for RepPoints learning. RPDet uses both an object localization loss and an object recognition loss (gradient from later recognition) 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 inluding the object 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 feedback cannot 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. For both detectors us-

[2]It can be also implemented by a deformable convolution operator with an unlearnable input offset field induced by the 3×3 grid points.

	Backbone	Anchor-Free	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN w. FPN [24]	ResNet-101		36.2	59.1	39.0	18.2	39.0	48.2
RefineDet [46]	ResNet-101		36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [25]	ResNet-101		39.1	59.1	42.3	21.8	42.7	50.2
Deep Regionlets [44]	ResNet-101		39.3	59.8	-	21.7	43.7	50.9
Mask R-CNN [13]	ResNeXt-101		39.8	62.3	43.4	22.1	43.2	51.2
FSAF [50]	ResNet-101		40.9	61.5	44.0	24.0	44.2	51.3
Cascade R-CNN [2]	ResNet-101		42.8	62.1	46.3	23.7	45.5	55.2
CornerNet [21]	Hourglass-104	✓	40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet [48]	Hourglass-104	✓	40.1	55.3	43.2	20.3	43.2	53.1
RPDet	ResNet-101	✓	41.0	62.9	44.3	23.6	44.1	51.7
RPDet	ResNet-101-DCN	✓	**42.8**	**65.0**	**46.3**	**24.9**	**46.2**	**54.7**
RPDet (ms train)	ResNet-101-DCN	✓	**45.0**	**66.1**	**49.0**	**26.6**	**48.6**	**57.5**
RPDet (ms train & ms test)	ResNet-101-DCN	✓	**46.5**	**67.4**	**50.9**	**30.3**	**49.7**	**57.1**

Table 7. Comparison of the proposed RPDet with the previous state-of-the-art detectors on COCO [26] `test-dev`. The proposed RPDet can achieve 46.5 AP, significantly surpasses all other detectors. Also note the AP_{50} of RPDet is relatively high, which is believed as a better metric in many applications [32]. "ms" indicates multi-scale.

ing bounding boxes and RepPoints, the center point based method surpass the anchor based method by +1.1 mAP and +1.4 mAP, respectively, likely because of its better coverage of ground-truth objects.

We also compare the proposed anchor-free detector based on RepPoints to RetinaNet [25] (a popular one-stage anchor-based method), FPN [24] with RoIAlign (a popular two-stage anchor-based method) [13], and a YOLO-like detector which is adapted from the anchor-free method of YOLOv1 [31], in Table 4. The proposed method outperforms both RetinaNet [25] and the FPN [24] 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 +0.8 mAP gain, implying that the roles of deformable RoI pooling and the proposed RepPoints are complementary.

5.3. RepPoints Visualization

In Figure 4, we visualize the learned *RepPoints* and the corresponding detection results on several examples from the COCO [26] `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 with 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 benchmark [26], which is on-par with 4-stage anchor-based Cascade R-CNN [2] and outperforms all existing anchor-free approaches. By adopting multi-scale training and testing (see Appendix for details), RPDet can achieve 46.5 AP, significantly surpassing all other detectors. Also note the AP_{50} of RPDet is relatively high, which is believed as a better metric in many applications [32].

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] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, pages 168–181. Springer, 2010. 2, 3

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2, 3, 8

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 2, 3, 5, 7, 8, 11, 12

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE Computer Society, 2005. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 6

[7] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2147–2154, 2014. 1

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

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 2, 3

[10] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 2, 3

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1, 2, 11

[12] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 6, 12

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 3, 7, 8, 11

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *PAMI*, 37(9):1904–1916, 2015. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 6, 7, 12

[16] Chang Huang, Haizhou Ai, Yuan Li, and Shihong Lao. High-performance rotation invariant multiview face detection. *PAMI*, 29(4):671–686, 2007. 2

[17] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015. 4

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 3

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2

[20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom 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

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 2, 8

[22] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume 1, pages 878–885. IEEE, 2005. 2

[23] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. 2

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *ICCV*, pages 2117–2125, 2017. 2, 5, 6, 8, 12

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2, 3, 5, 6, 8

[26] 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 *ECCV*, pages 740–755. Springer, 2014. 2, 5, 6, 7, 8, 12

[27] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 2

[28] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, pages 2277–2287, 2017. 2

[29] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, pages 4930–4939, 2017. 2

[30] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015. 1

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 4, 5, 6, 8

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 8

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 3, 11

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[35] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *CVPR*, pages 3578–3587, 2018. 2

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2

[37] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 1

[38] Lachlan Tychsen-Smith and Lars Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *ICCV*, pages 428–436, 2017. 2

[39] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR*, 1:511–518, 2001. 2

[40] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *CVPR*, 2019. 2

[41] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual deanimation. In *NIPS*, pages 153–164, 2017. 1

[42] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, pages 3–19, 2018. 6

[43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 12

[44] Hongyu Xu, Xutao Lv, Xiaoyu Wang, Zhou Ren, Navaneeth Bodla, and Rama Chellappa. Deep regionlets for object detection. In *ECCV*, pages 798–814, 2018. 8

[45] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *NeurIPS*, pages 318–328, 2018. 2

[46] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018. 8

[47] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Anchor box optimization for object detection. *arXiv preprint arXiv:1812.00469*, 2018. 2

[48] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2, 3, 8

[49] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *CVPR*, pages 519–528, 2017. 2

[50] Chenchen Zhu, Yihui He, and Marios 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

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

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

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

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 [11], RoIPooling [33], or RoIAlign [13]. 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:

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

Thus, the pooled feature $\mathcal{P}_B(I, B_P)$ should be sensitive to the box proposal B_P. Specifically, for any pair of proposals $\mathcal{B}_1 \neq \mathcal{B}_2$, we should have $\mathcal{P}_B(I, B_1) \neq \mathcal{P}_B(I, B_2)$. Most existing feature extractors $\mathcal{P}_B$ satisfy this property. Note that the improvement of RoIAlign [13] over RoIPooling [33] 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 [13] to produce a set of sample points S_P for each proposal. This can be formulated as

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

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:

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

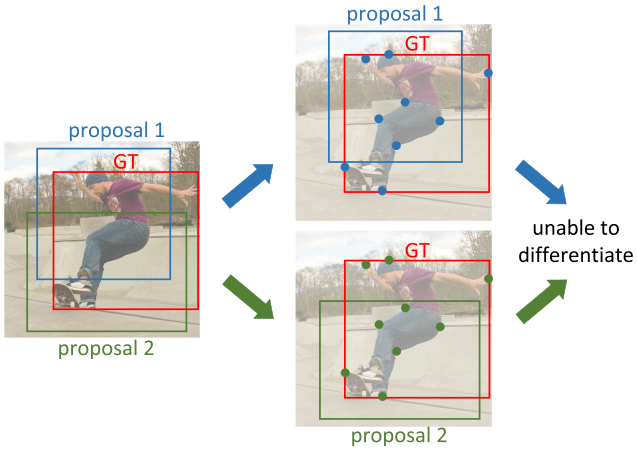

Figure 5. Illustration that deformable RoI pooling [4] is unable to serve as a geometric object representation, as discussed in Section 4 in the main paper. We consider two bounding box regressions based on different proposals. Assume that deformable RoI pooling [4] can learn a similar geometric object 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 6. Visualization of the learned sample points of 3×3 deformable RoI pooling [4]. It is shown that the scale of 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 5 illustrates the contradiction that arises if deformable RoI pooling were a representation of object ge-

method	backbone	ms train	ms test	AP
RPDet	R-50			38.6
	R-50	✓		40.8
	R-50	✓	✓	42.2
	R-101			40.3
	R-101	✓		42.3
	R-101	✓	✓	44.1
	R-101-DCN			43.0
	R-101-DCN	✓		44.8
	R-101-DCN	✓	✓	46.4
	X-101-DCN			44.5
	X-101-DCN	✓		45.6
	X-101-DCN	✓	✓	46.8

Table 8. Benchmark results of RPDet on MS-COCO [26] validation set (`minival`). All the models here are trained with FPN [24] under the '2x' setting [12]. For the backbone notation, 'R-50' and 'R-101' denotes ResNet-50 and ResNet-101 [15] respectively. 'R-101-DCN' denotes ResNet-101 with all convolution layers substituted with deformable convolution layers [4]. 'X' denotes the ResNeXt-101 [43] backbone. "ms" indicates multi-scale.

ometry. Moreover, Figure 6 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 4 in the main paper). While object localization supervision is not applied on the sample points of deformable RoI pooling, we show in Table 2 in the main paper 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 in the main paper), further indicating their different functionality.

A2. More Benchmark Results for RPDet

We present more benchmark results of our proposed detector RPDet in Table 8. Our PyTorch implementation is available at https://github.com/microsoft/RepPoints. All models were tested on MS-COCO [26] validation set (`minival`).

Multi-scale training and test settings. In multi-scale training, for each mini-batch, the shorter side is randomly selected from a range of $[480, 960]$. In multi-scale testing, we first resize each image to a shorter side of $\{400, 600, 800, 1000, 1200, 1400\}$. Then the detection results (before NMS) from all scales are merged, followed by a NMS step to produce the final detection results.