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 [1] 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

$$\mathcal{T}_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}), \tag{1}$$

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

$$\mathcal{R}_B(\mathcal{P}_B(I, \mathcal{B}_P)) = \mathcal{T}_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}), \tag{2}$$

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 [2], RoIPooling [8], or RoIAlign [4]. 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:

$$\mathcal{R}_B(\mathcal{P}_B(I, \mathcal{B}_P + \Delta\mathcal{B})) = \mathcal{F}(\mathcal{B}_P + \Delta\mathcal{B}, \mathcal{B}_{GT}). \tag{3}$$

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 [4] over RoIPooling [8] is partly due to this guaranteed translation sensitivity.

Analysis of Deformable RoI Pooling. For deformable RoI pooling [1], the system generates a pointwise deformation of samples on a regular grid [4] to produce a set of sample points S_P for each proposal. This can be formulated as

$$S_P = \mathcal{D}(I, \mathcal{B}_P), \tag{4}$$

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:

$$\mathcal{R}_S(\mathcal{P}_S(I, S_P)) = \mathcal{T}_P = \mathcal{F}(\mathcal{B}_P, \mathcal{B}_{GT}) \tag{5}$$

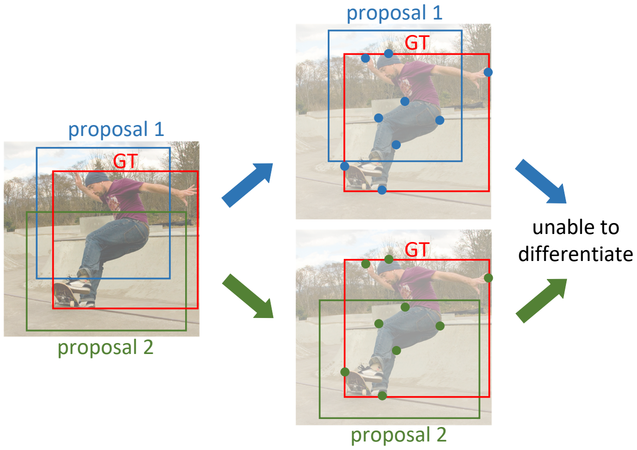

Figure 1. Illustration that deformable RoI pooling [1] is unable to serve as a geometric object representation, as discussed in Section 4 in the main paper. We consider two bounding box regressions based on different proposals. Assume that deformable RoI pooling [1] can learn a similar geometric object 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 [1] can indeed differentiate nearby object proposals, leading to a contradiction. Thus, it is concluded that deformable RoI pooling [1] cannot learn the geometric representation of objects.

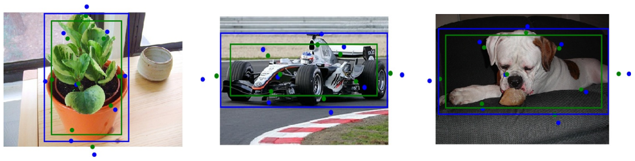

Figure 2. Visualization of the learned sample points of 3×3 deformable RoI pooling [1]. It is shown that the scale of sample points changes as the scale of the proposal changes, indicating that the sample points do not adapt to form a geometric *object* representation.

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 1 illustrates the contradiction that arises if deformable RoI pooling were a representation of object geometry. Moreover, Figure 2 illustrates that, for the learned

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 1. Benchmark results of RPDet on MS-COCO [7] validation set (`minival`). All the models here are trained with FPN [6] under the '2x' setting [3]. For the backbone notation, 'R-50' and 'R-101' denotes ResNet-50 and ResNet-101 [5] respectively. 'R-101-DCN' denotes ResNet-101 with all convolution layers substituted with deformable convolution layers [1]. 'X' denotes the ResNeXt-101 [9] backbone. "ms" indicates multi-scale.

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 [1] 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 1. Our PyTorch implementation is available at https://github.com/microsoft/RepPoints. All models were tested on MS-COCO [7] 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.

References

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 1, 2

[2] 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

[3] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018. 2

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1

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

[6] 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

[7] 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

[8] 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

[9] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 2