Rethinking the Misalignment Problem in Dense Object Detection

Yang Yang[1,2], Min Li[1,2]✉, Bo Meng[1,3], Zihao Huang[1,2],
Junxing Ren[1,2], and Degang Sun[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Beijing Institute of Technology, Beijing, China
`{yangyang1995, limin, renjunxing, sundegang}@iie.ac.cn`

Abstract. Object detection aims to localize and classify the objects in a given image, and these two tasks are sensitive to different object regions. Therefore, some locations predict high-quality bounding boxes but low classification scores, and some locations are quite the opposite. A misalignment exists between the two tasks, and their features are spatially entangled. In order to solve the misalignment problem, we propose a plug-in **S**pati**al**-disentangled and **T**ask-aligned operator (SALT). By predicting two task-aware point sets that are located in each task's sensitive regions, SALT can reassign features from those regions and align them to the corresponding anchor point. Therefore, features for the two tasks are spatially aligned and disentangled. To minimize the difference between the two regression stages, we propose a **S**elf-**d**istillation **r**egression (SDR) loss that can transfer knowledge from the refined regression results to the coarse regression results. On the basis of SALT and SDR loss, we propose SALT-Net, which explicitly exploits task-aligned point-set features for accurate detection results. Extensive experiments on the MS-COCO dataset show that our proposed methods can consistently boost different state-of-the-art dense detectors by ∼2 AP. Notably, SALT-Net with Res2Net-101-DCN backbone achieves 53.8 AP on the MS-COCO *test-dev*.

Keywords: Object detection · Misalignment problem · Spatial disentanglement

1 Introduction

The main goal of object detection contains two tasks, one is to give the accurate location of the object in an image (i.e., regression), and the other is to predict the category of the object (i.e., classification). During the inference step, the regression and classification results predicted from the same location are paired together as the detection result. Then the NMS algorithm is usually applied to remove redundant detection results by taking the classification scores as the ranking keywords. For the same instance, the detection result with a high classification score will be kept, while others are filtered out. However, the natures
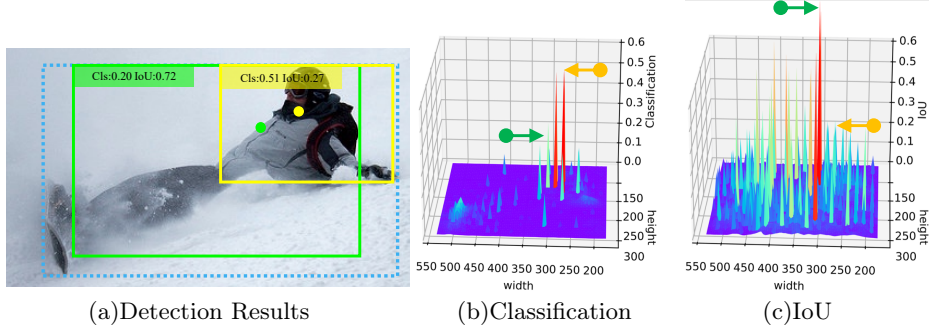
|(a)Detection Results|(b)Classification|(c)IoU|

Fig. 1: Illustration of the spatial misalignment of classification and regression. In (a), the blue box denotes the ground truth, and the other boxes are the detection results of ATSS [34]. The two points are the locations where the detection results are predicted. (b) and (c) are the distributions of classification and IoU scores over all image pixels, and "IoU" denotes the intersection over union between the predicted box and the ground truth.

of these two tasks are so distinct that they require features from different object locations [22]. As shown in Figure 1, the classification and regression quality (i.e., IoU) scores from the same location can be quite different. Classification focus on the salient part of the object (e.g., the head of the person), while regression is sensitive to the whole object, especially for its border part. *Therefore, the prediction distributions of the two tasks are misaligned.* The detection result with a high classification score can have low-quality regression prediction and vice versa.

We model the prediction qualities of the two tasks as two Discrete distributions. Therefore, the goal of solving the misalignment problem is *bridging the gap between these two distributions (i.e., minimizing the distance of their peak positions).*

CNN-based dense detectors utilize a coupled or decoupled head to conduct classification and regression. As illustrated in Figure 2 (a), the coupled head predicts the classification and regression results based on the shared features [20, 15, 19]. As a result, the coupled head structure introduces feature conflicts between the two tasks and makes them compromise each other. To solve this problem, the decoupled head structure [28] is proposed and has been widely adopted in recent years [22, 16, 23]. As shown in Figure 2 (b), the decoupled head utilizes two parrel sub-networks to perform regression and classification, respectively. This could alleviate the conflict problem by reducing the shared parameters. However, the point features (i.e., the two orange points) that predict the detection result still share the identical receptive field. In conclusion, *both the coupled and decoupled heads predict the classification and regression results from the spatially identical and entangled features.* Considering the difference in
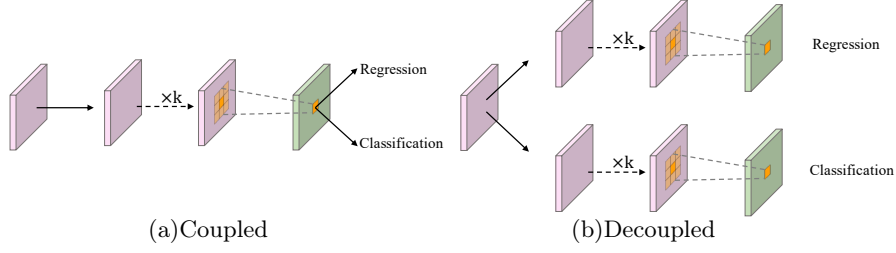
Fig. 2: Illustration of the entangled features in the coupled and decoupled heads.

their spatial sensitivity, the entangled features inevitably make a location prefer one task over the other one, thereby compounding the misalignment problem.

In this paper, we propose a plug-in operator to address the misalignment problem: the **S**patial-disentangled and **T**ask-aligned operator (SALT). The first stage of our network is the coarse regression predictions made by a simple Dirac delta decoder [34]. After that, SALT predicts two sets of spatial-disentangled points to represent each task's sensitive regions, respectively. Then we use bilinear interpolation to reassign features from those regions to the corresponding anchor point. In the second stage, SALT utilizes spatial-disentangled and task-aligned features to make refined predictions with a General distribution decoder [12]. Therefore, a single anchor point can obtain accurate regression and classification predictions simultaneously. Feature reassignment can bring the peak positions of the two Discrete distributions closer so that SALT can weaken the impact of the misalignment problem.

In order to minimize the difference between the first and second stage predictions, we also propose a novel **S**elf-**d**istillation **r**egression (SDR) loss, making the coarse predictions learn from the refined predictions. As a result, the final performance got improved without any extra inference cost.

1. We propose an operator that can generate spatial-disentangled and task-aligned features for regression and classification, respectively.
2. The proposed operator can be easily plugged into most dense object detectors and bring a considerable improvement of $\sim$2 AP.
3. Our proposed SDR loss can also boost the overall performance in an inference cost-free fashion.
4. Without bells and whistles, our best single-scale model (Res2Net-101-DCN) yields 51.5 AP on the COCO *test-dev* set, which is very competitive results among dense object detectors.

2 Related Work

Misalignment: Dense detectors, such as IoU-aware [27], FCOS [23] and PAA [10] apply an extra branch to predict the regression confidence and combine it with the classification confidence as the detection score. Different from previous methods, GFL [13] and VFNet [31] propose a joint representation format
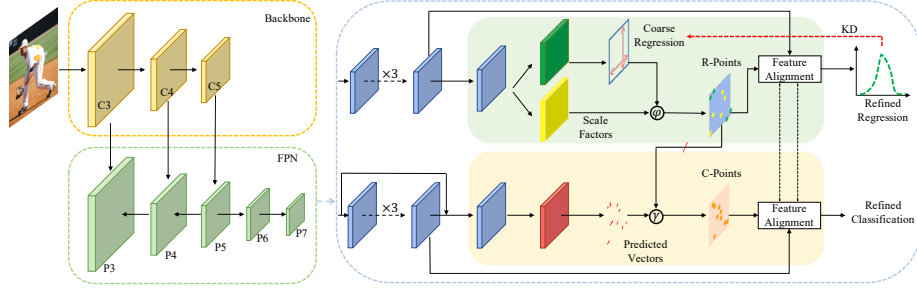
Fig. 3: Architecture of SALT-Net. Our proposed architecture consists of a backbone, an FPN (P3-P7), and two subnetworks for classification and regression, respectively. "φ" and "γ" denote Equations (2) and (5), respectively. "$/$" denotes the gradient flow detachment. "KD" denotes our proposed self-distillation approach. "R-points" and "C-Points" denote the regression-aware and classification-aware points, respectively.

by merging the regression confidence and classification result to eliminate the inconsistency between training and inference. TOOD [6] proposes a prediction alignment method that predicts the offset between each location and the best anchor and then readjusts the prediction results. Guided Anchoring [25], RefineDet [35], and SRN [3] learn an offset field for the preset anchor and then utilize a feature adaption module to extract features from the refined anchors. RepPoints [29] and VFNet [31] utilize the deformable convolution [4] to extract accurate point feature. However, all the aforementioned methods extract features for regression and classification from the same locations, without considering their spatial preference. That is, the features for these tasks are spatially entangled, which leads to inferior performance.

Self-distillation: Model distillation [7] usually refers to transferring knowledge from a pre-trained heavy teacher network to a compact student network. DML [36] provides a new paradigm that a pre-trained teacher is no longer needed and all the student counterparts are trained simultaneously in a cooperative peer-teaching manner. Following this paradigm, many self-distillation approaches [32, 30, 8, 14] are proposed for classification knowledge transfer learning. However, transferring regression knowledge of object detection has been proven to be difficult [9, 26], as different locations of an image have different contributions to the regression task. LGD [33] is the only self-distillation approach for general object detection, which proposes an intra-object knowledge mapper that generates a better feature pyramid and then performs distillation with feature imitation. This approach provides performance gains but also introduces too many auxiliary layers.
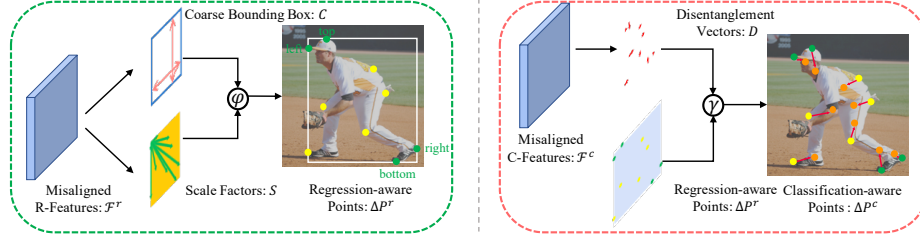
Fig. 4: (a) is the illustration of the regression-aware points. "φ" and the white box denotes Equation (2) and the coarse bounding box prediction, respectively. (b) is the illustration of the spatial disentanglement method. "γ" denotes Equation (5).

3   Proposed Approach

In this section, we first detail the proposed operator SALT. Then we introduce our self-distillation approach that enables the first-stage decoder to learn from the second-stage decoder. Finally, we introduce the loss function of SALT-Net.

3.1   SALT: Regression-aware Points

As shown in Figure 4 (a), given the misaligned regression features $\mathcal{F}^r$ from the last layer of the regression tower (i.e., the $3\times$ convolutions shown in Figure 3), SALT first predicts the coarse bounding box $\mathcal{C}$ with the Dirac delta decoder, as in [23, 34]. The coarse bounding box is represented by the top-left corner and its width and height (i.e., $(xmin, ymin, w, h)$).

Then SALT predicts the scale Factors $\mathcal{S}$ that measures the normalized distances between the top-left corner of the coarse bounding box and the regression-sensitive regions (i.e., regression-aware points $\Delta\mathcal{P}^r$). Scale factors $\mathcal{S}$ and the coarse bounding box $\mathcal{C}$ are obtained by only two convolution layers, i.e.:

$$\begin{cases} \mathcal{C} = \delta(conv_c(\mathcal{F}^r)) \\ \mathcal{S} = \sigma(conv_s(\mathcal{F}^r)) \end{cases} \tag{1}$$

where σ and δ are Sigmoid and ReLU, respectively. $\mathcal{C} \in \mathbb{R}^{H \times W \times 4}$, $\mathcal{S} \in \mathbb{R}^{H \times W \times (2N-4)}$ and N is the number of the regression-aware points. Then the location of i-th regression-aware point Δp_i can be obtained with Equation (2):

$$\begin{cases} x_i = x_{\min} + w * \mathcal{S}_{ix} \\ y_i = y_{\min} + h * \mathcal{S}_{iy} \end{cases} \tag{2}$$

where $(xmin, ymin)$ is the location of the top-left corner of the coarse bounding box $\mathcal{C}$, and $(\mathcal{S}_{ix}, \mathcal{S}_{iy})$ are scale factors that measure the normalized distance between the i-th point and top-left corner. Therefore, the location of the regression-aware points $\Delta\mathcal{P}^r$ can be represented by:

$$\Delta\mathcal{P}^r = \{\Delta p_i^r\}_{i=1}^N \tag{3}$$

Note that all coordinates are represented by taking the location that predicts the detection result as the coordinate origin. Therefore, the coordinates mentioned in this section are relative locations, not absolute coordinates.

The total number of the regression-aware points is N, the channels of scale factors $\mathcal{S}$ and the regression-aware points $\Delta\mathcal{P}^r$ are $2N-4$ and $2N$, respectively. The reason for this inconstancy is that we want to ensure that the sampled regression-aware points contain the four extreme points (i.e., left-most, right-most, top-most, bottom-most), which encode the location of the object. On this account, four points are sampled on the four bounds of the coarse bounding box (i.e., the green points in Figure 4 (a)), respectively. As the location of the bounding box has been predicted, four axial coordinates of the extreme points are preset and do not need to be learned (i.e., $xmin, ymin, xmin+w, ymin+h$).

3.2   SALT: Classification-aware Points

Regression and classification are sensitive to different areas of the object. For this reason, extracting features from the regression-interested-locations hinders the detection performance. Therefore, SALT contains a spatial disentangle module to guide the classification branch to generate a set of classification-aware points. As shown in Figure 4 (b), the regression-aware points act as the shape hypothesis of the object to be classified. In other words, we take the regression-aware points as a point-set anchor for predicting the classification-aware points.

Similar to the scale factors, this module also consists of only one convolution layer. As shown in Figure 4 (b), given the feature map $\mathcal{F}^c$ from the last layer of the classification tower, the disentanglement vectors $\mathcal{D}$ are obtained by:

$$\mathcal{D} = \delta(conv_d(\mathcal{F}^c)) \tag{4}$$

With the regression-aware points $\Delta\mathcal{P}^r$ taken as the point-set anchor, we propose two functions to generate the classification-aware points, as illustrated by Equation (5) and (6). We choose Equation (5) as the final prediction strategy. Details and analysis can be found in Sec. 4.2.

$$\Delta\mathcal{P}^c = e^{\mathcal{D}} \cdot \Delta\mathcal{P}^r \tag{5}$$

$$\Delta\mathcal{P}^c = \mathcal{D} + \Delta\mathcal{P}^r \tag{6}$$

To make sure the learning process of classification and regression are independent of each other. The gradient flow of the regression-aware points $\Delta\mathcal{P}^r$ is detached from the classification branch. $\Delta\mathcal{P}^r$ only serves as the prior knowledge in this module. Therefore, the supervision of the classification task does not affect the learning of regression-aware points.

3.3   SALT: Feature Alignment

The regression-aware and classification-aware points are located in each task's sensitive regions, and they are spatially misaligned. Therefore, we aggregate
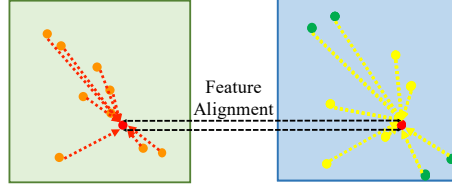
Fig. 5: Illustration of the feature alignment method. The red point denotes the location that predicts the detection result.

features from those regions to the same anchor point (shown in Figure 5). Given the learned point set $\Delta \mathcal{P} = \{\Delta p_i\}_{i=1}^N$, we use the bilinear interpolation to make $\Delta \mathcal{P}$ differentiable. Let $\{p_i\}_{i=1}^N$ be the sampling window of a regular convolution grid, where N is the number of grid points. The new irregular sampling locations can be represented by Equation (7), and the bilinear interpolation is formulated as Equation (8),

$$\hat{\mathcal{P}} = \{p + p_i + \Delta p_i \mid i = 1, \ldots, N\} \tag{7}$$

$$\hat{\mathcal{F}}(p) = \sum_{\hat{p}} G(\hat{p}, p) \cdot \mathcal{F}(\hat{p}) \tag{8}$$

where $\mathcal{F}(\cdot)$ and $\hat{\mathcal{F}}(\cdot)$ are the input and output feature maps, and $G(\cdot, \cdot)$ is the bilinear interpolation kernel. $\hat{p} \in \hat{\mathcal{P}}$, and p is the location that predicts the detection result.

The aligned task features are extracted from the locations of the task-aware points, and then they are used for classification and regression refinement. Different from the first stage, the second regression stage utilizes the General distribution decoder [13] that outputs the discrete representation of the bounding box.

3.4   Self-distillation

As Figure 6 shows, the bottom boundary of the handcrafted annotation is inaccurate and ambiguous, which can misguide and hurt the training process. However, the network's localization prediction results sometimes provide better and clearer regression targets that are easier for the network to learn. For this reason, we propose a self-distillation regression loss (SDR) that could transfer regression knowledge from the refined predictions to the coarse ones.

As Equation 9 shows, R_1, R_2, and y are the output of the stage-one, stage-two decoders, and the classification score. IoU_1 and IoU_2 denote the Intersection over Union between the ground truth label and the corresponding regression results, and $GIoU$ is the Generalized Intersection over Union as in [21]. As the stage-two decoder is fed with better features, we take its outputs as the regression upper bound of the stage-one decoder. That is, we utilize the integral results from the
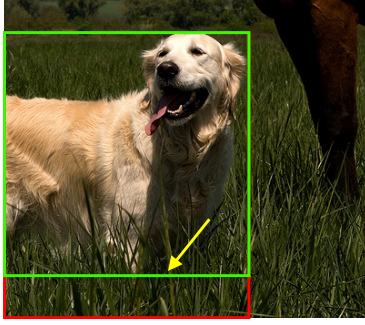
Fig. 6: The red and green boxes are the ground truth and the bounding box prediction made by our stage-two decoder, respectively.

Discrete distribution decoder as the soft target for the Dirac delta decoder. Here, $y \cdot IoU_2$ denotes the confidence score of the refined regression result, and its gradient is detached. Thus, SDR loss pays more attention to the high-confidence prediction results. Notably, SDR loss only penalizes the network when the predictions of the stage-two decoder are better than the stage-one decoder (i.e., $IoU_2 > IoU_1$). Our proposed SDR loss enables the coarse predictions to learn from the refined results and bridges the gap between them. Better stage-one predictions lead to better stage-two predictions and promote the training process into positive circulation.

$$\mathcal{L}_{SDR}(R_1, R_2, y) = \begin{cases} y \cdot IoU_2 \cdot (1 - GIoU(R_1, R_2)), & \text{if } IoU_2 > IoU_1 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

3.5 Loss Function

The proposed SALT-Net is optimized in an end-to-end fashion, and both the coarse and the refined detection stages utilize ATSS [34] as the positive and negative targets assignment strategy. The training loss of SALT-Net is defined as follows:

$$\begin{aligned} \mathcal{L} = {} & \frac{1}{N_{\text{pos}}} \sum_z \lambda_0 \mathcal{L}_{\mathcal{Q}} \\ & + \frac{1}{N_{\text{pos}}} \sum_z \mathbf{1}_{\{c_z^* > 0\}} \left(\lambda_1 \mathcal{L}_{\mathcal{R}_1} + \lambda_2 \mathcal{L}_{\mathcal{R}_2} + \lambda_3 \mathcal{L}_{\mathcal{D}} + \lambda_4 \mathcal{L}_{\mathcal{SDR}} \right) \end{aligned} \quad (10)$$

where $\mathcal{L}_{\mathcal{Q}}$ is the Quality Focal loss [13] for the classification task. $\mathcal{L}_{\mathcal{R}_1}$ and $\mathcal{L}_{\mathcal{R}_2}$ are both $GIoU$ loss [21], one for the coarse bounding box prediction and the other for the refined regression result. $\mathcal{L}_{\mathcal{D}}$ is the Distribution Focal Loss [13] for optimizing the general distribution representation of the bounding box, and $\mathcal{L}_{\mathcal{SDR}}$ is the proposed self-distillation loss. $\lambda_0 \sim \lambda_4$ are the hyperparameters used to balance different losses, and they are set as $1, 1, 2, 0.5$, and 1, respectively.

Table 1: Ablation study of SALT on the COCO val2017 split. S_1 and S_2 denote the stage-one and stage-two regression results. "R-Points" and "C-Points" denote the regression-aware and Classification-aware points, respectively. "P-anchor" denotes utilizing the regression-aware points as the point-set anchor for generating the Classification-aware points. "skip" denotes the skip connection of the classification tower, as shown in Figure 3.

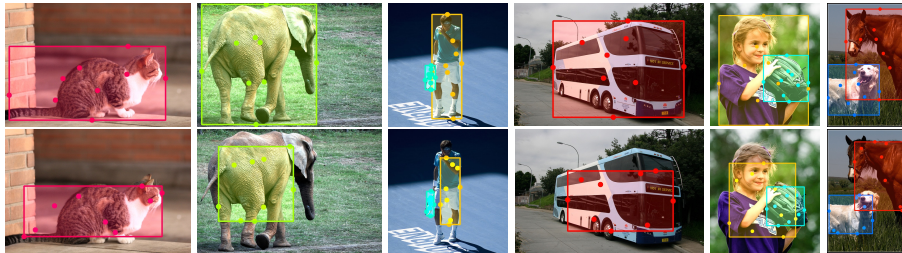| Method	R-Points	C-Points	P-anchor	skip	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
baseline										
S_1[34]					39.9	58.5	43.0	22.4	43.9	52.7
S_2[12]					40.9	58.3	44.4	23.9	44.7	53.5
S_2	✓				41.3	58.7	44.9	23.3	45.0	54.2
S_2	✓	✓			41.6	59.1	45.6	23.7	45.4	54.7
S_2	✓	✓	✓		42.1	59.6	45.6	24.8	45.4	55.5
S_2	✓	✓	✓	✓	**42.5**	**60.1**	**46.2**	**25.1**	**45.9**	**56.4**
S_1	✓	✓	✓	✓	41.3	58.6	44.9	23.2	45.1	54.2

Fig. 7: Visualization of the regression-aware (upper row) and classification-aware (lower row) points. Different task-aware points are located on the different areas of the object, and their sensitive regions (i.e., the bounding boxes) are distinct.

N_{pos} denotes the number of selected positive samples, and z denotes all the locations on the pyramid feature maps. $\mathbf{1}_{\{c_z^* > 0\}}$ is the indicator function, being 1 if $c_z^* > 0$ and 0 otherwise.

4   Experiments

Figure 3 presents the network of our proposed SALT-Net. We take state-of-the-art dense detectors ATSS [34] and GFLv2 [12] as our baseline, and they serve as the stage-one and stage-two decoders, respectively. Our SALT-Net is evaluated on the challenging MS-COCO benchmark [17]. Following the common practice, we use the COCO train2017 split (115K images) as the training set and the COCO val2017 split (5K images) for the ablation study. To compare with state-of-the-art detectors, we report the COCO AP on the *test-dev* split (20K images) by uploading the detection result to the MS-COCO server.

Table 2: Spatial disentanglement strategies. "exp" and "+" denotes utilizing Equation (5) and (6) to generate the classification-aware points, respectively.

Method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
w/ +	42.3	60.1	46.1	24.7	**46.0**	56.1
w/ exp	**42.5**	**60.1**	**46.2**	**25.1**	45.9	**56.4**

Table 3: Performance of implementing our proposed approach in popular dense detectors.

Method	AP	AP$_{50}$	AP$_{75}$
FCOS	38.6	57.2	41.7
SALT-FCOS	**40.9** (+2.3)	**59.3**	**44.2**
RepPoints w/ GridF	37.4	58.9	39.7
SALT-RepPoints	**39.0** (+1.6)	**60.5**	**41.6**

4.1   Performance of SALT's component parts

To validate the effectiveness of different component parts of our proposed operator SALT, we gradually add the proposed modules to the baseline. As shown in Table 1, the second and third rows are the baseline performances of the stage-one and stage-two decoders, respectively. Note that the stage-one decoder utilizes joint representation of IoU and classification scores instead of its original centernesss branch, as in [13]. The baseline performances of the two stages are 39.9 AP and 40.9 AP, respectively.

As presented in the fourth row, the first experiment investigates the effect of implementing the regression-aware points. Therefore, SALT only predicts the scale factors $\mathcal{S}$ for generating the regression-aware points. Both subnetworks utilize aligned features from the locations of the regression-aware points for the refined detection results. The AP is improved to 41.3, which indicates that the aligned features do improve the detection accuracy, even though features for the two tasks are still spatially entangled.

As shown in the fifth row, to test the effect of spatial disentanglement, SALT predicts the disentanglement vectors $\mathcal{D}$ for generating the classification-aware points. Note that these points are learned without the regression-aware points acting as the point-set anchor (i.e., $\Delta \mathcal{P}^c = \mathcal{D}$), yet the AP is still boosted to 41.6. These classification-aware points are located in different regions from the regression-aware points, and higher accuracy is obtained (41.6 vs. 41.3). Therefore, spatial disentanglement does raise the detection performance by eliminating their spatial feature conflicts.

The sixth row shows the performance when taking the regression-aware points as the point-set anchor for generating the classification-aware points. It can be observed that a notable performance gain is achieved (i.e., 0.5 AP improvement). That thereby proves the effectiveness of utilizing the regression-aware points as the shape hypothesis and the importance of task disentanglement. Figure 7 is

Table 4: The effect of SDR loss

Method	SDR	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
S_1		41.3	58.6	44.9	23.2	45.1	54.2
S_2		42.5	60.1	46.2	25.1	45.9	56.4
S_1	✓	**42.1**(+0.8)	60.5	45.9	24.8(+1.6)	45.7	54.9
S_2	✓	**42.8**(+0.3)	60.6	46.7	25.1	46.4	56.0

the visualization of task-aware points and their sensitive regions. This figure indicates that classification and regression are sensitive to different locations of the object, which also gives the interpretability of spatial disentanglement.

As shown in the seventh row, the long-range skip connection (i.e., the residual connection on the classification tower) can also bring a considerable performance boost and gain 0.4 AP. Note that the overall performance has been improved by 1.6 AP and 2.9 AP$_L$ compared with the strong baseline. Finally, the last row indicates that the coarse regression results with the refined classification results can also improve the baseline performance by 1.4 AP.

4.2   The Selection of Spatial Disentanglement Strategies

We propose two disentanglement functions to generate the classification-aware points, as illustrated in Equation (5) and (6). In Equation (5), the disentanglement vector set $\mathcal{D}$ is taken as the exponent, whereas $\mathcal{D}$ is directly aggregated with $\Delta\mathcal{P}^r$ in Equation (6). As illustrated in Table 2, the "exp" strategy performs better than that of "+." The reason is that predicting log-space transforms (i.e., $\mathcal{D} = \ln \frac{\Delta\mathcal{P}^c}{\Delta\mathcal{P}^r}$), instead of directly predicting the distance (i.e., $\mathcal{D} = \Delta\mathcal{P}^c - \Delta\mathcal{P}^r$), prevents unstable gradients during training. Therefore, it is easier to be learned.

4.3   Generality of SALT

Our proposed SALT can act as a plug-in operator for dense detectors. Therefore, we plug SALT into popular detectors [23] and [29], to validate its generality. As shown in Table 3, the performance gain is 2.3 AP on FCOS, which is a considerable improvement. Compared with RepPoints, our SALT-RepPoints performs better than it and gains 1.6 AP. One can see that SALT can significantly improve the accuracy of different detectors, which demonstrates its generality.

4.4   Self-distillation Regression Loss

The baseline for this ablation study is the best model of Table 1. Here, both stages utilize the refined classification scores as the NMS ranking keywords. As Table 4 shows, after applying the SDR loss to the SALT-Net, the performance of both stages got improved. The performance gain of the stage-one decoder is an absolute 0.8 AP score. Notably, the performance on small objects has been

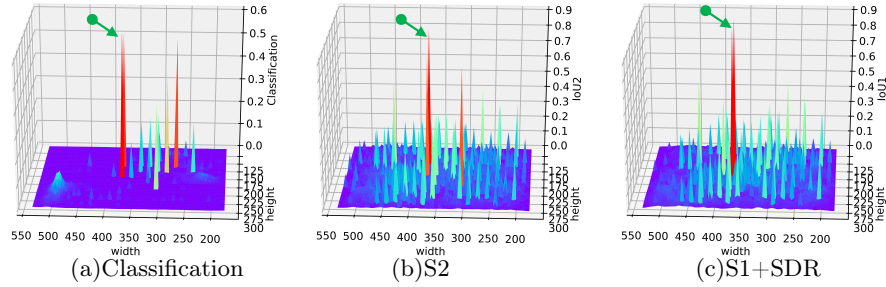(a)Classification   (b)S2   (c)S1+SDR

Fig. 8: Prediction distributions (Figure 1) after applying SALT and SDR loss. The locations of their distribution peaks are identical.

improved by 1.6 AP, which is a relatively large margin compared with the strong baseline. Furthermore, the improvement of the stage-one decoder also brings positive feedback to the stage-two decoder and leads to the highest performance of our SALT-Net (i.e., 42.8 AP).

4.5   Evaluations for Task-alignment of SALT-Net

Figure 8 (a) and (b) are the distributions of the refined detection results when implementing SALT, whereas Figure 1 (b) and (c) are the original coarse prediction distributions made by the stage-one decoder. The green arrows point to the distribution peaks, and one can see that they are spatially aligned (i.e., at the same location). Therefore, the detection result with the highest classification score also has the best regression result, and the misalignment gap is bridged. Figure 8 (c) is the IoU distribution of the stage-one decoder after applying the SDR loss. Its quality distributions become very close to the stage-two decoder (i.e., Figure 8 (b)), which proves the effectiveness of the regression knowledge transfer. In Figure 9, the qualitative results show that SALT can align the regression and detection tasks and thereby suppress some low IoU but high classification score results.

4.6   Comparisons with State-of-the-arts

The multi-scale training strategy (i.e., input images are resized from [400, 1333] to [960, 1333]) and the 2 × schedule [1] are adopted as they are commonly used strategies in state-of-the-art methods. GFLV2 only applies DCN on the last two stages of the backbone, whereas the common practices [24, 31] usually apply it on the last three stages. Therefore, for a fair comparison, the results of the proposed method with both settings are reported. As Table 5 shows, our model achieves a 46.1 AP with ResNet-50, which outperforms other state-of-the-art methods with heavier backbones (e.g., FCOS with X-101-32x8d-DCN). With test-time augmentations and R2-101-DCN as the backbone, our best model achieves a 53.8 AP, which is a very competitive result among dense object detectors.

Table 5: SALT-Net vs. State-of-the-art Detectors. All test results are reported on the COCO *test-dev* set. "DCN^2, DCN"-applying Deformable Convolutional Network [38] on the last two and three stages of the backbone, respectively. "DCN^p"-applying DCN on both the backbone and the FPN. "*" indicates applying SDR loss and "†" indicates test-time augmentations, including horizontal flip and multi-scale testing.

Method	Backbone	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
multi-stage								
GuidedAnchor[25]	R-50	12	39.8	59.2	43.5	21.8	42.6	50.7
DCNV2 [38]	X-101-32x8d-DCN	24	44.5	65.8	48.4	27.0	48.5	58.9
BorderDet[18]	X-101-64x4d-DCN	24	48.0	67.1	52.1	29.4	50.7	60.5
RepPointsV2[2]	X-101-64x4d-DCN	24	49.4	68.9	53.4	30.3	52.1	62.3
TSD†[22]	SE154-DCN	24	51.2	71.9	56.0	33.8	54.8	64.2
VFNet[31]	X-101-32x8d-DCN	24	50.0	68.5	54.4	30.4	53.2	62.9
LSNet[5]	R2-101-DCN^p	24	51.1	70.3	55.2	31.2	54.3	65.1
one-stage								
CornerNet [11]	HG-104	200	40.5	59.1	42.3	21.8	42.7	50.2
SAPD[37]	X-101-32x8d-DCN	24	46.6	66.6	50.0	27.3	49.7	60.7
ATSS[34]	X-101-32x8d-DCN	24	47.7	66.5	51.9	29.7	50.8	59.4
GFL[13]	X-101-32x8d-DCN	24	48.2	67.4	52.6	29.2	51.7	60.2
FCOS-imprv [24]	X-101-32x8d-DCN	24	44.1	63.7	47.9	27.4	46.8	53.7
PAA [10]	X-101-64x4d-DCN	24	49.0	67.8	53.3	30.2	52.8	62.2
GFLV2 [12]	R-50	24	44.3	62.3	48.5	26.8	47.7	54.1
GFLV2 [12]	X-101-32x8d-DCN^2	24	49.0	67.6	53.5	29.7	52.4	61.4
TOOD [6]	X-101-64x4d-DCN	24	51.1	69.4	55.5	31.9	54.1	63.7
SALT-Net*	R-50	24	46.1	64.0	50.3	28.0	49.5	57.2
SALT-Net	X-101-32x8d-DCN^2	24	49.8	68.5	54.2	30.6	53.2	62.6
SALT-Net	X-101-32x8d-DCN	24	50.2	68.8	54.9	31.2	53.4	63.1
SALT-Net	R2-101-DCN^2	24	51.1	69.7	55.7	32.3	54.5	64.0
SALT-Net*	R2-101-DCN	24	51.5	70.0	56.2	32.1	55.1	64.8
SALT-Net*†	R2-101-DCN	24	**53.8**	**71.1**	**59.9**	**36.3**	**56.9**	**65.1**

5   Conclusion

In this work, we presented SALT, a simple yet effective plug-in operator that can solve the misalignment problem between regression and classification. Our new-fashioned framework can disentangle classification and regression from the spatial dimension by extracting features from each task's sensitive locations and aligning them to the same anchor point. We also proposed SDR loss to transfer the regression knowledge from the stage-two decoder to the stage-one decoder. The refined detection results also received positive feedback by improving the coarse regression results, and the final performance improved in an inference cost-free fashion. Extensive experiments showed that SALT could considerably raise the performance of various dense detectors, and SALT-Net showed promising results among the state-of-the-art dense detectors.
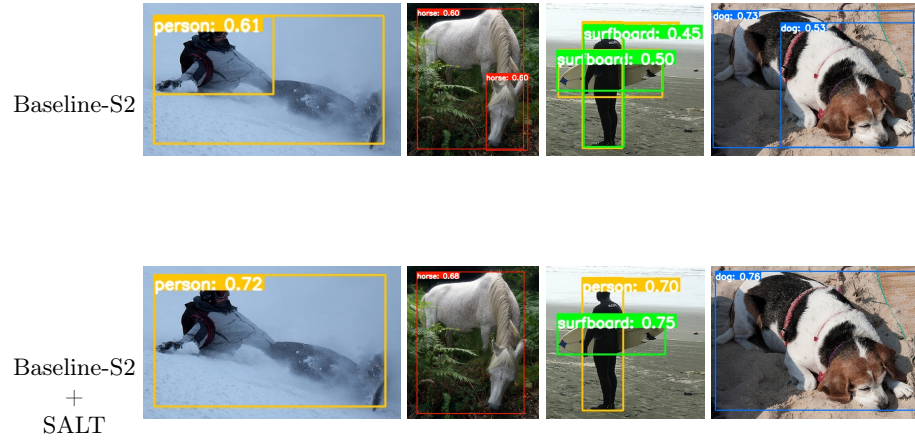
Fig. 9: Comparisons between the stage-two baseline decoder and our SALT-Net.

References

1. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al.: Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
2. Chen, Y., Zhang, Z., Cao, Y., Wang, L., Lin, S., Hu, H.: Reppoints v2: Verification meets regression for object detection. Advances in Neural Information Processing Systems **33** (2020)
3. Chi, C., Zhang, S., Xing, J., Lei, Z., Li, S.Z., Zou, X.: Selective refinement network for high performance face detection. In: AAAI. vol. 33, pp. 8231–8238 (2019)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017)
5. Duan, K., Xie, L., Qi, H., Bai, S., Huang, Q., Tian, Q.: Location-sensitive visual recognition with cross-iou loss. arXiv preprint arXiv:2104.04899 (2021)
6. Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W.: Tood: Task-aligned one-stage object detection. In: ICCV. pp. 3490–3499. IEEE Computer Society (2021)
7. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)
8. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: CVPR. pp. 10664–10673 (2021)
9. Kang, Z., Zhang, P., Zhang, X., Sun, J., Zheng, N.: Instance-conditional knowledge distillation for object detection. In: NeurIPS (2021)
10. Kim, K., Lee, H.S.: Probabilistic anchor assignment with iou prediction for object detection. In: ECCV. pp. 355–371. Springer (2020)
11. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV. pp. 734–750 (2018)

12. Li, X., Wang, W., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. In: CVPR. pp. 11632–11641 (2021)
13. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In: NeurIPS (2020)
14. Li, Z., Li, X., Yang, L., Yang, J., Pan, Z.: Student helping teacher: Teacher evolution via self-knowledge distillation. arXiv preprint arXiv:2110.00329 (2021)
15. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
18. Qiu, H., Ma, Y., Li, Z., Liu, S., Sun, J.: Borderdet: Border feature for dense object detection. In: ECCV. pp. 549–564. Springer (2020)
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR. pp. 658–666 (2019)
22. Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. In: CVPR. pp. 11563–11572 (2020)
23. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: ICCV. pp. 9627–9636 (2019)
24. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: A simple and strong anchor-free object detector. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
25. Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D.: Region proposal by guided anchoring. In: CVPR. pp. 2965–2974 (2019)
26. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: CVPR. pp. 4933–4942 (2019)
27. Wu, S., Li, X., Wang, X.: Iou-aware single-stage object detector for accurate localization. Image and Vision Computing **97**, 103911 (2020)
28. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization for object detection. In: CVPR. pp. 10186–10195 (2020)
29. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: ICCV. pp. 9657–9666 (2019)
30. Yao, A., Sun, D.: Knowledge transfer via dense cross-layer mutual-distillation. In: ECCV. pp. 294–311. Springer (2020)
31. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: CVPR. pp. 8514–8523 (2021)
32. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: ICCV. pp. 3713–3722 (2019)
33. Zhang, P., Kang, Z., Yang, T., Zhang, X., Zheng, N., Sun, J.: Lgd: Label-guided self-distillation for object detection. arXiv preprint arXiv:2109.11496 (2021)

34. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR. pp. 9759–9768 (2020)
35. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: CVPR. pp. 4203–4212 (2018)
36. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: CVPR. pp. 4320–4328 (2018)
37. Zhu, C., Chen, F., Shen, Z., Savvides, M.: Soft anchor-point object detection. In: ECCV. pp. 91–107. Springer (2020)
38. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR. pp. 9308–9316 (2019)