

# AutoAssign: Differentiable Label Assignment for Dense Object Detection

Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, Jian Sun  
Megvii Research

{zhubenjin, wangjianfeng, jiangzhengkai, zongfuhang, liusongtao, lizeming, sunjian}@megvii.com

## Abstract

In this paper, we propose an anchor-free object detector with a fully differentiable label assignment strategy, named AutoAssign. It automatically determines positive/negative samples by generating positive and negative weight maps to modify each location’s prediction dynamically. Specifically, we present a center weighting module to adjust the category-specific prior distributions and a confidence weighting module to adapt the specific assign strategy of each instance. The entire label assignment process is differentiable and requires no additional modification to transfer to different datasets and tasks. Extensive experiments on MS COCO show that our method steadily surpasses other best sampling strategies by  $\sim 1\%$  AP with various backbones. Moreover, our best model achieves 52.1% AP, outperforming all existing one-stage detectors. Besides, experiments on other datasets, e.g., PASCAL VOC, Objects365, and WiderFace, demonstrate the broad applicability of AutoAssign.

## 1. Introduction

Current state-of-the-art CNN based object detectors perform a common paradigm of dense prediction. Both two-stage (the RPN [17] part) and one-stage detectors [11, 20, 27, 26] predict objects with various scales, aspect ratios, and classes over every CNN feature locations in a regular, dense sampling manner.

This dense detection task raises an essential issue of sampling positives and negatives in the spatial locations, which we call *label assignment*. Moreover, as the modern CNN-based detectors commonly adopt multi-scale features (e.g., FPN [10]) to alleviate scale variance, label assignment requires not only selecting locations among spatial feature maps but also choosing the level of features with appropriate scale.

As shown in Fig. 1, existing detectors mainly sample the positive and negative locations by human prior: (1) Anchor-based detectors like RetinaNet [11] preset several anchors of diverse scales and aspect ratios on each location and re-

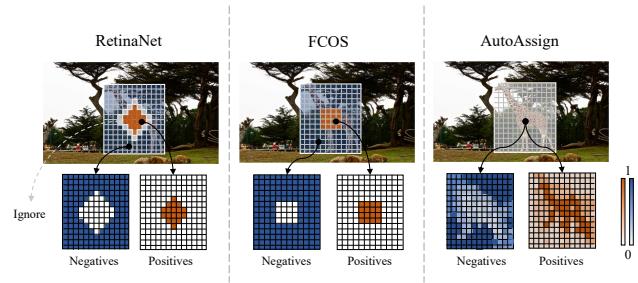


Figure 1. Label assignment results of RetinaNet, FCOS and AutoAssign. Only one scale is shown for better visualization

sort to the Intersection over Union (IoU) for sampling positives and negatives among spatial and scale-level feature maps. (2) Anchor-free detectors like FCOS [20] sample a fixed fraction of center area as spatial positive locations for each object, and select certain stages of FPN [10] by the pre-defined scale constraints. These detectors follow the prior distribution of the objects to design their assignment strategies, which are proved to be effective on challenging benchmarks, e.g., Pascal VOC [3, 4] and MS COCO [12].

However, as shown in Fig. 2, in the real world, appearances of objects vary a lot across categories and scenarios. The fixed center sampling strategy may pick locations outside objects as positives. Intuitively, sampling locations on objects is better than the plain background because these locations are prone to generate higher classification confidences. On the other hand, although CNN can learn offsets, the obstacle caused by feature shifting when backgrounds are sampled as positives may decrease the performance.

Thus the fixed strategies above may not always select the most appropriate locations among spatial and scale dimensions. Beyond the pure human-designed strategies, a few recent works introduce some partially data-dependent and dynamic strategies in label assignment. GuidedAnchoring [21] and MetaAnchor [23] dynamically change the prior of anchor shapes before sampling, while other methods adaptively modify the sampling strategy for each object in the spatial dimension [27, 26, 9] or the scale dimension [31]. These strategies only free a part of label assignment to be

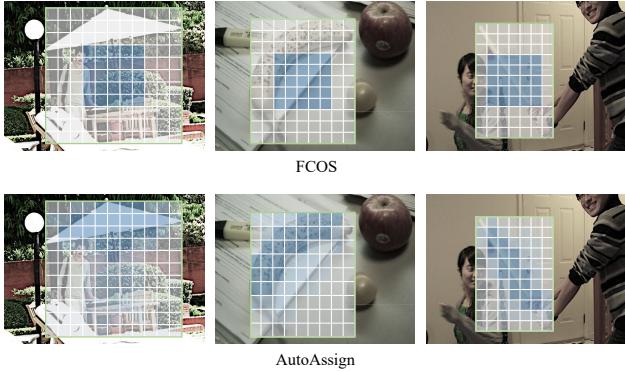


Figure 2. Comparison of label assignment between FCOS and AutoAssign. Label assignment results of FCOS are fixed, while AutoAssign makes label assignment results to be dynamic

data-driven, while the other parts stay constrained by human designs, preventing label assignment from being further optimized.

In this work, we propose a fully differentiable strategy for label assignment. As illustrated in Fig. 1, we first follow the anchor-free manner like FCOS [20] to directly predict objects on each locations without human-designed anchors. In order to retain enough locations for further optimizing, we initially treat all the locations inside a bounding box as both positive and negative candidates at all scale levels. Then we generate positive and negative weight maps to modify the prediction in the training loss. To accommodate the distribution from different categories and domains, we propose a category-wise weighting module named center weighting to learn the distribution of each category from data. To get adapted to the appearance and scale of each instance, we propose a confidence weighting module to modify the positive and negative confidences of the locations in both spatial and scale dimensions. Then we combine the two modules to generate positive and negative weight maps for all the locations. The entire process of weighting is differentiable and can be conveniently optimized by back-propagation.

In summary, the contributions of this study are three-fold as follows:

1. A new differentiable label assignment strategy named AutoAssign is proposed for dense object detection, which automatically assigns the positives and negatives for each instance. Moreover, the learnable paradigm enables it to transfer to different datasets and tasks without any modification.
2. Two weighting modules, *i.e.*, center weighting and confidence weighting, are proposed to adaptively adjust the category-specific distribution as well as the instance-specific sampling strategy in both spatial and

scale dimensions.

3. Extensive experiments with competitive results are carried out on the challenging MS COCO [12] dataset and other datasets from different domains, such as PASCAL VOC [3, 4], Object365 [19] and WiderFace [22], demonstrating the effectiveness and broad applicability of AutoAssign.

## 2. Related Work

### 2.1. Fixed Label assignment

Classical object detectors sample positives and negatives with pre-determined strategies. Region Proposal Networks (RPN) in Faster R-CNN [17] preset several anchors with diverse scales and aspect ratios on each location. Then the assignment in both scale and spatial dimensions for each instance is guided by the anchor matching IoU. This anchor-based strategy quickly dominates modern detectors and extends to multi-scale output features such as YOLO [15, 16], SSD [13], and RetinaNet [11]. Recent academic attention has been geared toward anchor-free detectors without anchor settings. FCOS [20] and its precursors [7, 25, 14] drop the prior anchor setting and directly assign the spatial positions around bounding box center of each object as positives. In scale dimension, they pre-define the scale range of each FPN [10] stage to assign instances. Both the anchor-based and anchor-free strategies follow *center prior* of data distributions, which indicates that spatial locations near the bounding box center are more likely to contain objects.

There are also some other anchor-free detectors [8, 29, 2, 24, 28] based on a different mechanism, which treats bounding boxes as key-points and transforms regression task to classification on heat maps. The characteristics of these detectors are distinct from detectors based on bounding box regression. Therefore they are out of the scope of this paper.

### 2.2. Dynamic Label assignment

As the fixed assigning strategies may be suboptimal for the various object distributions, recent detectors propose adaptative mechanisms to improve label assignment. GuidedAnchoring [21] leverages semantic features to guide the anchor settings and dynamically change the shape of anchors to fit various distributions of objects, while MetaAnchor [23] randomly samples anchors of any shape during training to cover different kinds of object boxes. Besides the modification of anchor prior, some works directly change the sampling for each object. FSAF [31] dynamically assigns each instance to the most suitable FPN feature level with minimal training loss. SAPD [30] reweights the positive anchors and applies an extra meta-net to select the proper FPN stages. FreeAnchor [27] constructs a

Method	Prior	Instance		AP
		scale	spatial	
RetinaNet [11]	anchor	size & IoU	IoU	36.3
FreeAnchor [27]	anchor	size & IoU	top- $k$ weighting, IoU	38.7
ATSS [26]	anchor	size & IoU	top- $k$ , dynamic IoU	39.3
GuidedAnchoring [21]	dynamic anchor	size & IoU	IoU	37.1
FCOS* [20]	center	range	radius	38.7
FSAF [31]	anchor & center	loss	IoU & radius	37.2
AutoAssign (Ours)	dynamic center	weighting	weighting	<b>40.5</b>

Table 1. Comparison of label assignment between different typical detectors. Results in terms of *AP* (%) are reported on the MS COCO 2017 *val* set, using ResNet-50 [6] as backbone. \* denotes improved versions

bag of top- $k$  anchor candidates based on IoU for every object and uses a Mean-Max function to weight among selected anchors, and [9] designs another weighting function to eliminate noisy anchors. ATSS [26] proposes an adaptive training sample selection mechanism by the dynamic IoU threshold according to the statistical characteristics of instances.

To show the existing label assignment strategies from a more holistic perspective, we organize the critical components of some representative methods as prior-related and instance-related in Table 1. Clearly, apart from the heuristic-based methods like RetinaNet [11] and FCOS [20], all the existing dynamic strategies only make some components of label assignment to be data-driven. In contrast, the other components still follow the hand-crafted rules.

### 3. AutoAssign

#### 3.1. Overview

AutoAssign tackles label assignment in a fully data-driven manner. It is built from scratch and has no traditional components, e.g., anchors, IoU thresholds, top- $k$  or scale ranges. It directly uses network predictions to dynamically adjust the confidence of positive/negative at each location.

To optimize the entire label assignment process, we propose a fully differentiable strategy that dynamically adjusts the category-specific and instance-specific sampling strategies in both spatial and scale dimensions. The framework of our strategy is demonstrated in Fig. 3. We first follow the anchor-free manner like FCOS [20] to drop the pre-designed anchors and directly predict objects on each feature location. For each instance, we keep all the locations inside its bounding box among all scale levels as both positives and negatives. Then we generate positive and negative weight maps  $w^+$  and  $w^-$  to modify the prediction of positives and negatives in training precisely. Thus we transform the whole assignment step into two weight maps.

To accommodate to the distributions of different cat-

egories, we propose a category-wise and data-dependent weighting module named center weighting. It starts from the standard center prior and then learns the distribution of each category from data.

To get adapted to the appearance and scale of each instance, we further present an instance-wise weighting module called confidence weighting. It dynamically weights the positions in the spatial and scale dimensions based on the predicted confidences of each object.

Finally, we combine the two weighting modules to generate positive weights and negative weights for all the locations. Given an object  $n$ , we formulate the training loss  $\mathcal{L}_n(\theta)$  after applying our entire weighting mechanism as follows:

$$\mathcal{L}_n(\theta) = -\log\left(\sum_{i \in S_n} w_i^+ \mathcal{P}_i^+\right) - \sum_{i \in S_n} \log(w_i^- \mathcal{P}_i^-), \quad (1)$$

where  $S_n$  denotes all locations inside the bounding box at all the scale levels. For a location  $i \in S_n$ , its probabilities of being positive and negative are represented as  $\mathcal{P}_i^+$  and  $\mathcal{P}_i^-$  respectively, which are predicted by the network.  $w_i^+$  and  $w_i^-$  are weight maps generated by the proposed center weighting and confidence weighting.

#### 3.2. Center Weighting

The prior distribution is a fundamental element for label assignment, especially in the early stage of training. In general, the distribution of objects is subject to the center prior. However, the objects from different categories, e.g., giraffe, and human, may have distinct distributions. Keeping sampling center positions cannot capture the different distributions of real-world instances. Preferably, for objects of different categories, adaptive center distributions are more desired.

Based on the center prior, we introduce a category-wise Gaussian-shape weighting function  $G$  with learnable parameters. “category-wise” means each category has its unique parameters  $(\mu, \sigma)$  while all objects of the same cate-

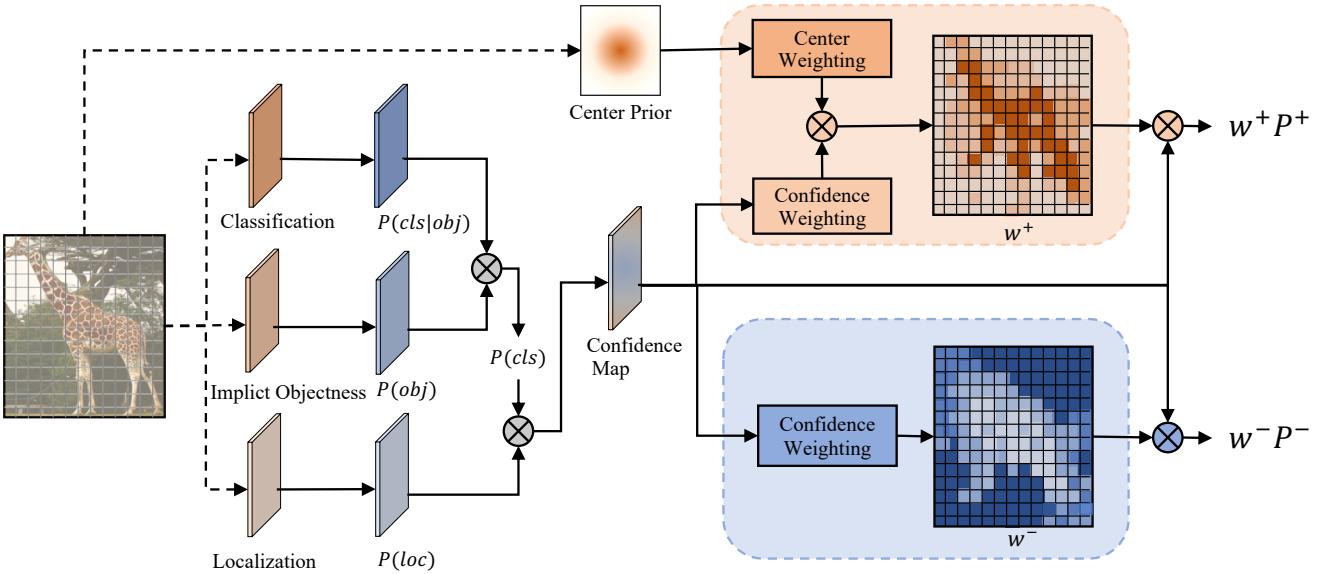


Figure 3. Illustration of weighting in AutoAssign. Specifically, given an object, we first keep all the locations inside its bounding box as both positives and negatives. Then the proposed center weighting module and confidence weighting module dynamically generate the positive and negative weight maps, which can adjust the confidences at each location of being positive and negative. AutoAssign has the same architecture as FCOS. Thus only one proposal will be predicted at each location. The *ImpObj* branch shares weights with localization branch

gory share the same parameters. Here we define  $G$  as:

$$G(\vec{d} \mid \vec{\mu}, \vec{\sigma}) = e^{-\frac{(\vec{d} - \vec{\mu})^2}{2\vec{\sigma}^2}}, \quad (2)$$

where  $\vec{d}$  denotes the offsets of a certain position inside an object to its box center along  $x$ - and  $y$ -axis, which means it can be negative.  $\vec{\mu}$  and  $\vec{\sigma}$  are learnable parameters of shape  $(K, 2)$ .  $K$  is the number of categories of a dataset. Each category has two parameters along  $x$ - and  $y$ -axis of the spatial dimension. As  $G$  contributes to the training loss, the parameters can be optimized by back-propagation.

Intuitively,  $\vec{\mu}$  controls center offset of each category from the box center. And  $\vec{\sigma}$  measures each position's importance based on category characteristics. Thus  $\vec{\sigma}$  determines how many positions will effectively contribute to positive loss with reasonable weights.

It also needs to be noticed that  $G$  is applied to all FPN stages. Since objects of the same category can have arbitrary sizes or aspect ratios, the most appropriate locations can be at any of the FPN stages. Furthermore, to compensate for the interference caused by the different downsampling rates of FPN, we normalize the distance  $\vec{d}$  by its FPN stage's downscale ratio.

### 3.3. Confidence Weighting

Existing dynamic strategies [31, 27, 9] are designed with the fact that networks can easily learn proper samples with high confidence while tending to predict low confidence for inferior samples. Specifically, the confidence indicator is

separately proved to be effective in the scale selection (loss indicator in [31]) and the spatial assignment (anchor confidence in [27, 9]). In confidence weighting, we propose a joint confidence indicator of both classification and localization to guide the weighting strategy in both spatial and scale dimensions.

#### 3.3.1 Classification confidence

Given a spatial location  $i$ , its confidence of classification can be naturally defined as  $\mathcal{P}_i(\text{cls}|\theta)$ , the probability of the object class that is directly predicted by the network.  $\theta$  denotes model parameters. However, in order to make sure all the proper locations are considered, we initially take all spatial locations inside boxes into account. As an object can hardly fill its bounding box completely, the initial positives set tends to contain a considerable part of background locations. If a location is, in fact, background, all class predictions in the location should be unreasonable. So taking too many inferior background locations as positives will damage detection performance.

To suppress false positives from the inferior locations, we introduce a novel *Implicit-Objectness* branch, which is shown in Fig. 3. It works just like the Objectness in RPN [17] and YOLO [14] as a binary classification task for foregrounds and backgrounds. But here we meet another issue of lacking explicit labels. RPN and YOLO adopt pre-defined assignment with consistent positive labels, while we need to find and emphasize proper positives dynamically.

We thus optimize the Objectness together with the classification branch, so it does not require explicit labels. That is why it is called *Implicit-Objectness* (*ImpObj*). For a location  $i \in S_n$ , after *ImpObj* is applied, the classification confidence  $\mathcal{P}_i(\text{cls}|\theta)$  can be defined as:

$$\mathcal{P}_i(\text{cls}|\theta) = \mathcal{P}_i(\text{cls}|\text{obj}, \theta)\mathcal{P}_i(\text{obj}|\theta), \quad (3)$$

where the proposed *ImpObj* acts as  $\mathcal{P}_i(\text{obj}|\theta)$ , which represents the probability of position  $i$  being foreground(object) or background.  $\mathcal{P}_i(\text{cls}|\text{obj}, \theta)$  is the probability of position  $i$  being a specific category given this position is known to be foreground or background. As demonstrated in Fig. 3,  $\mathcal{P}_i(\text{cls}|\text{obj}, \theta)$  is classification output. Its the same as classification branch in other detectors like RetinaNet [11].

To provide another view of understanding the novel *ImpObj*, we can take previous label assignment strategies as manually select foregrounds, that is setting  $\mathcal{P}_i(\text{obj}) = 1$  for positives and 0 for negatives. In this case, the classification confidence  $\mathcal{P}_i(\text{cls}|\theta) = \mathcal{P}_i(\text{cls}|\text{obj}, \theta)$ . While in AutoAssign,  $\mathcal{P}_i(\text{obj}|\theta)$  is dynamically decided by the network.

### 3.3.2 Joint confidence modeling

For generating unbiased estimation of each location towards positives/negatives, besides classification, we should also include the localization confidence. The typical outputs of localization are box offsets, which are hard to measure the regression confidence directly. We thus convert the localization loss  $\mathcal{L}_i^{\text{loc}}(\theta)$  into likelihood  $\mathcal{P}_i(\text{loc}|\theta)$ , then we combine classification and regression likelihood together to get the joint confidence  $\mathcal{P}_i(\theta)$ . It can be derived from loss  $\mathcal{L}_i(\theta)$  as follows. Without loss of generality, here we use Binary Cross-Entropy (BCE) loss for classification.

$$\mathcal{L}_i(\theta) = \mathcal{L}_i^{\text{cls}}(\theta) + \lambda\mathcal{L}_i^{\text{loc}}(\theta) \quad (4)$$

$$= -\log(\mathcal{P}_i(\text{cls}|\theta)) + \lambda\mathcal{L}_i^{\text{loc}}(\theta) \quad (5)$$

$$= -\log(\mathcal{P}_i(\text{cls}|\theta)e^{-\lambda\mathcal{L}_i^{\text{loc}}(\theta)}) \quad (6)$$

$$= -\log(\mathcal{P}_i(\text{cls}|\theta)\mathcal{P}_i(\text{loc}|\theta)) \quad (7)$$

$$= -\log(\mathcal{P}_i(\theta)), \quad (8)$$

where  $\lambda$  is the loss weight to balance between classification and localization.

### 3.3.3 Weighting function

Based on the joint confidence representation  $\mathcal{P}_i(\theta)$ , we propose our confidence weighting function  $C(\mathcal{P}_i)$  in an exponential form to emphasize the locations with high confidence containing objects as:

$$C(\mathcal{P}_i) = e^{\frac{\mathcal{P}_i(\theta)}{\tau}}, \quad (9)$$

where  $\tau$  is the temperature coefficient, it controls the contributions of high and low confidence locations towards positive losses.

## 3.4. Weight Maps

### 3.4.1 Positive weights

Intuitively, given an object  $i$ , for all locations inside its bounding box, we should focus on the proper locations with more accurate predictions. However, at the start of the training process, the network parameters are randomly initialized, making its predicted confidences unreasonable. Thus guiding information from prior is also critical. For location  $i \in S_n$ , we combine the category-specific prior  $G(\vec{d}_i)$  from center weighting module and the confidence weighting module  $C(\mathcal{P}_i)$  together to generate the positive weights  $w_i^+$  as:

$$w_i^+ = \frac{C(\mathcal{P}_i)G(\vec{d}_i)}{\sum_{j \in S_n} C(\mathcal{P}_i)G(\vec{d}_i)}, \quad (10)$$

### 3.4.2 Negative weights

As discussed above, a bounding box usually contains an amount of real-background locations, and we also need weighted negative losses to suppress these locations and eliminate false positives. Moreover, as the locations inside the boxes always tend to predict high confidence of positives, we prefer the localization confidence to generate the unbiased indicator of false positives. Paradoxically, the negative classification has no gradient for the regression task, which means the localization confidence should not be further optimized. Hence we use IoUs between each position's predicted proposal and all objects to generate our negative weights  $w_i^-$  as:

$$w_i^- = 1 - f\left(\frac{1}{1 - \text{iou}_i}\right), \quad (11)$$

in which  $\text{iou}_i$  denotes max IoU between proposal of location  $i \in S_n$  and all ground truth boxes. To be used as valid weights, we normalize  $1/(1 - \text{iou}_i)$  into range  $[0, 1]$  by its value range by function  $f$ . This transformation sharpens the weight distributions and ensure that the location with highest IoU receives zero negative loss. For all locations outside bounding boxes,  $w_i^-$  is set to 1 because they are backgrounds for sure.

## 3.5. Loss function

By generating positive and negative weight maps, we achieve the purpose of dynamically assigning more appropriate spatial locations and automatically selecting the proper FPN stages for each instance. As the weight maps contribute to the training loss, AutoAssign tackles the label

assignment in a differentiable manner. The final loss function  $\mathcal{L}(\theta)$  of AutoAssign can be defined as follows:

$$\mathcal{L}(\theta) = - \sum_{n=1}^N \log \left( \sum_{i \in S_n} w_i^+ \mathcal{P}_i^+ \right) - \sum_{j \in S} \log \left( w_j^- \mathcal{P}_j^- \right), \quad (12)$$

where  $\mathcal{P}^- = 1 - \mathcal{P}(\text{cls}|\theta)$ , and  $n$  denotes the  $n$ -th ground truth. To ensure at least one location matches object  $n$ , we use weighted sum of all positive weights to get the final positive confidence.  $S$  denotes all the locations at all the scales. Thus for a location inside bounding boxes, both positive and negative loss will be calculated with different weights. This is a huge difference from all other label assignment strategies. To handle the imbalance problem among negative locations, Focal Loss is applied.

Although the magnitude of positive and negative weights might be different according to Eq. 10, 11, the positive loss and negative loss are calculated independently.

## 4. Experiments

Experiments are mainly evaluated on the challenging MS COCO 2017 [12] benchmark, which contains around 118k images in the *train* set, 5k in the *val* set and 20k in the *test-dev* set. We report analyses and ablation studies on the *val* set and compare our final results with other methods on the *test-dev* set.

### 4.1. Implementation Details

We use ResNet-50 [6] with FPN [10] as backbone for all experiments if not specifically pointed out. We initialize the backbone with weights pre-trained on ImageNet [1]. Followed the common practice, all models are trained for  $1\times$  schedule named in [5], i.e., 90k iterations with an initial learning rate of 0.01, which is then divided by 10 at 60k and 80k iterations, with the weight decay of 0.0001 and the momentum of 0.9. Horizontal image flipping is utilized in data augmentation. For all ablation studies we use an image scale of 800 pixels for training and testing, unless otherwise specified. We set  $\tau = 1/3$  in Eq. 9, and  $\lambda = 5.0$  in Eq. 4. We apply Focal Loss [11] with  $\alpha = 0.25$  and  $\gamma = 2.0$  for negative classification and GIOU loss [18] for box localization. Non-maximum suppression (NMS) with IoU threshold 0.6 is applied to merge the results.

### 4.2. Ablation Studies

#### 4.2.1 Center weighting and confidence weighting

To demonstrate the effectiveness of the two key components, we construct the positives weights  $w_i^+$  using center weighting and confidence weighting separately, while keeping the negatives weighting unchanged. As shown in Table 2, it can be seen that center weighting brings relatively significant performance gain, suggesting that the prior distribution is critical for guiding the training. Besides, confidence

weighting further improves the accuracy as it dynamically changes the strategy for each object in both spatial and scale dimensions.

Center	Conf	<i>AP</i>	<i>AP<sub>50</sub></i>	<i>AP<sub>75</sub></i>	<i>AP<sub>S</sub></i>	<i>AP<sub>M</sub></i>	<i>AP<sub>L</sub></i>
	✓	17.7	30.9	18.1	15.7	24.2	23.3
✓		21.5	35.8	22.6	16.6	28.9	36.0
✓		37.7	57.4	40.6	20.3	41.4	52.0
✓	✓	<b>40.5</b>	<b>59.8</b>	<b>43.9</b>	<b>23.1</b>	<b>44.7</b>	<b>52.9</b>

Table 2. Contributions of center weighting and confidence weighting. “Center” means center weighting, and “Conf” indicates confidence weighting. The 17.7 mAP baseline is the start point of AutoAssign, which can be seen as removing  $w^+$  and  $w^-$  from Eq. 12. Other detectors, like RetinaNet, can also be implemented by adding modules to this simple baseline

To better understand the working mechanism of the two proposed modules, we visualize the positive weight maps separately in each FPN stage from a well-trained detector. From Fig. 4, we can see that the center weighting is applied to all stages of FPN to achieve a coarse weighting based on the category-specific center prior. Then the confidence weighting performs spatial and scale assignments for each instance. Objects of different shapes and different sizes are assigned to its appropriate spatial locations and suitable scale stages automatically. Our final positive weight maps that combine the effect of center weighting and confidence weighting are shown in the third row.

#### 4.2.2 Analysis of center weighting

To analyze the design of the center weighting, we compare different prior distributions in Table 3. We denote the Gaussian-shape function  $G$  without learnable parameters as “fixed”, while “shared” means only one group of learnable  $\vec{\mu}$  and  $\vec{\sigma}$  is shared among all categories. Compared to the fixed prior, “shared” prior slightly drops the AP by 0.1%, while our category-wise prior increases the AP of 0.2% on MS COCO. As MS COCO contains 80 categories with a huge amount of data, its object distribution generally falls into a normal distribution. Thus the total improvement of category-wise prior is not significant. But when we look into some classes with unique distributions, *e.g.*, bear, surfboard and hotdog, the improvements are notable.

This can also be evidenced by the visualization of the learned priors for each category in Fig. 5. We mark white points as the center of bounding boxes and red points as the center of learned priors. We can see that in the categories of parking meter and hotdog, the learned centers  $\vec{\mu}$  shift down as these categories tend to have more essential clues at the bottom half. Moreover, the category-specific  $\vec{\sigma}$  is also changed for each category. For the categories of

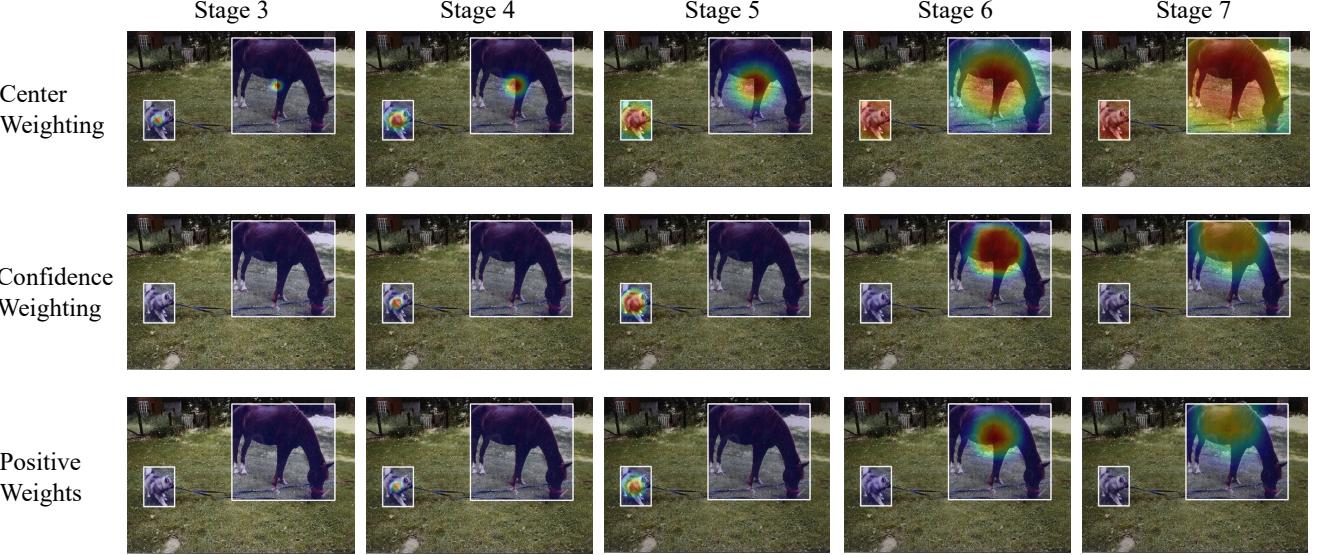


Figure 4. Visualization of center weighting, confidence weighting, and positive weights. From the middle row, objects of different sizes can be automatically assigned to their appropriate FPN stages.

Center	<i>AP</i>	moto	prk-mtr	bear	surfboard	hotdog
none	21.5	15.2	14.9	66.3	7.9	11.5
fixed	40.3	42.2	41.9	71.9	32.4	33.5
shared	40.2	41.8	40.7	69.2	33.1	32.7
<b>category</b>	<b>40.5</b>	<b>42.9</b>	<b>43.3</b>	<b>73.6</b>	<b>34.8</b>	<b>35.8</b>

Table 3. Results of different center weighting choices over the whole categories of MS COCO and the subset. “moto” means motorcycle, “prk-mtr” means parking meter

motorcycle and surfboard, the prior becomes ellipses to accommodate the shape characteristics of these categories.

#### 4.2.3 Analysis of confidence weighting

Confidence	<i>AP</i>	<i>AP<sub>50</sub></i>	<i>AP<sub>75</sub></i>	<i>AP<sub>S</sub></i>	<i>AP<sub>M</sub></i>	<i>AP<sub>L</sub></i>
$\mathcal{P}(\text{cls})$ -only	38.7	<b>59.9</b>	41.6	22.9	42.0	49.5
$\mathcal{P}(\text{loc})$ -only	39.7	58.4	43.1	22.4	43.6	51.6
no-obj	39.4	58.7	42.5	22.4	43.5	50.7
explicit-obj	39.5	58.8	42.3	21.6	43.4	52.2
AutoAssign	<b>40.5</b>	59.8	<b>43.9</b>	<b>23.1</b>	<b>44.7</b>	<b>52.9</b>

Table 4. Comparison of different choices for confidence weighting.  $\mathcal{P}(\text{cls})$ -only means only use  $\mathcal{P}(\text{cls})$  for confidence weighting. “no-obj” means do not use  $\text{ImpObj}$  for  $\mathcal{P}(\text{cls})$ . “explicit-obj” means give the object-ness branch individual supervision, rather than sharing with classification

We evaluate the effectiveness of classification confidence

Method	<i>ImpObj</i>	<i>AP</i>	<i>AP<sub>50</sub></i>	<i>AP<sub>75</sub></i>
RetinaNet [11]	✓	35.9	55.9	38.2
		36.1	56.3	38.8
FCOS* [20]	✓	38.8	57.6	42.2
		39.0	58.2	42.3
FreeAnchor [27]	✓	38.3	57.1	41.2
		38.5	57.6	41.5
ATSS [26]	✓	39.3	57.8	42.6
		39.7	58.4	42.9
AutoAssign	✓	39.4	58.7	42.5
		<b>40.5</b>	<b>59.8</b>	<b>43.9</b>

Table 5. Contribution of *ImpObj* to typical detectors. The number in brackets indicates the contribution of *ImpObj* to corresponding detectors. For example, *ImpObj* brings 0.2% improvements to RetinaNet. Bold fonts indicate the best performance.

$\mathcal{P}(\text{cls})$ , localization confidence  $\mathcal{P}(\text{loc})$ , and *ImpObj* separately in Table 4. In the first two rows, we respectively use classification confidence  $\mathcal{P}(\text{cls})$  and localization confidence  $\mathcal{P}(\text{loc})$  alone in the confidence weighting. Then in the next two rows, we evaluate the effect of *Implicit-Objectness*. *Explicit-Objectness* means that we explicitly supervise the Objectness with consistent positive labels for all the locations inside the boxes. From the whole table we can see that the combination of classification and localization confidence brings the major improvement and the design of *Implicit-Objectness* is also critical.

We also notice *Implicit-Objectness* is a universal design



Figure 5. Visualization of learned center weighting weights of different categories. All of the objects are visualized at the same scale. Center weighting results of motorcycle and surfboard show the prior distribution become ellipses (controlled by  $\sigma$ ) to accommodate the shape characteristics of these categories. Center offsets (controlled by  $\mu$ ) are actually larger than 10 pixels in the raw image, which means it could shift one or more grids on output feature maps.

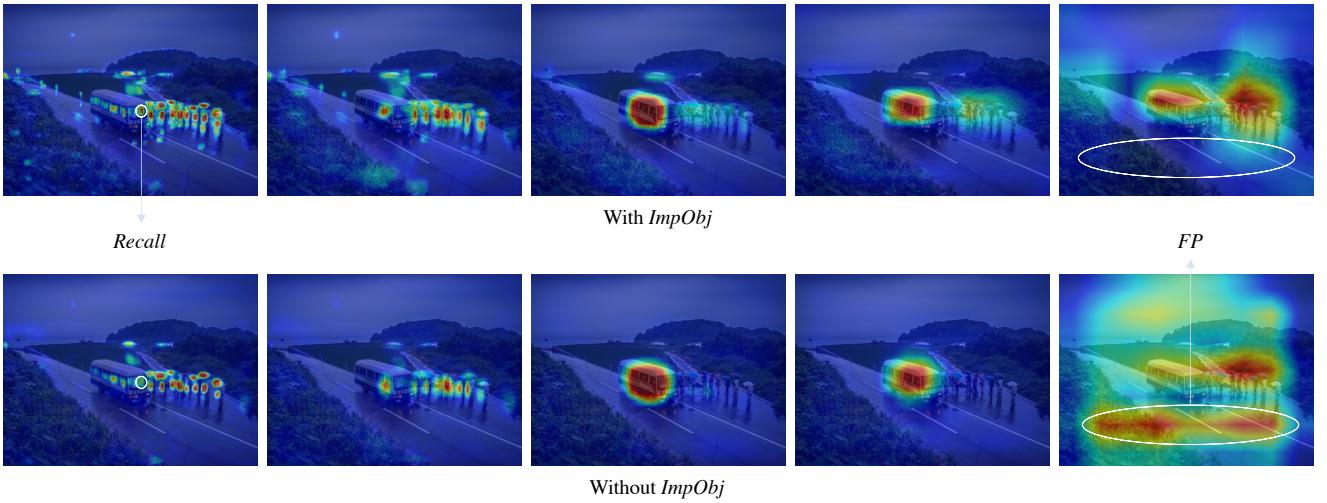


Figure 6. Confidence visualization of detectors trained with/without *ImpObj*. With the help of *ImpObj*, more objects can be detected, and false positives from background locations can be suppressed, thus recall and precision can be improved

which can be applied to other detectors. As shown in Table 5, due to the constrained number of positive locations (e.g. top k anchors in FreeAnchor), the contribution of *ImpObj* to those detectors are minor. This indicates that *ImpObj* is more suitable for fully dynamic assignment.

To better understand the behavior of *ImpObj*, we compare the final classification score used for NMS in Fig. 6. From visualization and evaluation results, improvements come from both recall and precision. The proposed *implicit-objectness* can filter out the noise and achieve better separation from background.

Then we investigate the learning process of confidence weighting in Fig. 7. We only visualize two representative FPN stages for simplicity. In the beginning, confidence weighting is weak for all objects because the probabilities among all locations are similarly low. With the training progress progressing, confidences become more salient and gradually converge to their appropriate FPN stages for objects of different sizes, demonstrating the effectiveness of our learnable strategy.

### 4.3. Comparison with State-of-the-art

We compare AutoAssign with other state-of-the-art detectors on MS COCO *test-dev* set. We adopt  $2\times$  schedule following the previous works [20, 27, 26]. Results are shown in Table 6. Our AutoAssign with ResNet-101 backbone achieves 44.5% AP, outperforming all state-of-the-art one-stage detectors with the same backbone as we known. By changing the backbone and training setting the same to other methods, our method consistently outperforms other counter-parts. With a wider range of multi-scale training and multi-scale testing strategy, our best model achieves 52.1% AP, which outperforms all existing one-stage detectors.

### 4.4. Generalization

To demonstrate the generalization ability, we evaluate our AutoAssign with several typical label assign strategies on different detection tasks, including general object detection (PASCAL VOC [3, 4], Objects365 [19]) and face detection (WiderFace [22]). In these experiments we keep the assignment strategy of each method (like anchor scale and

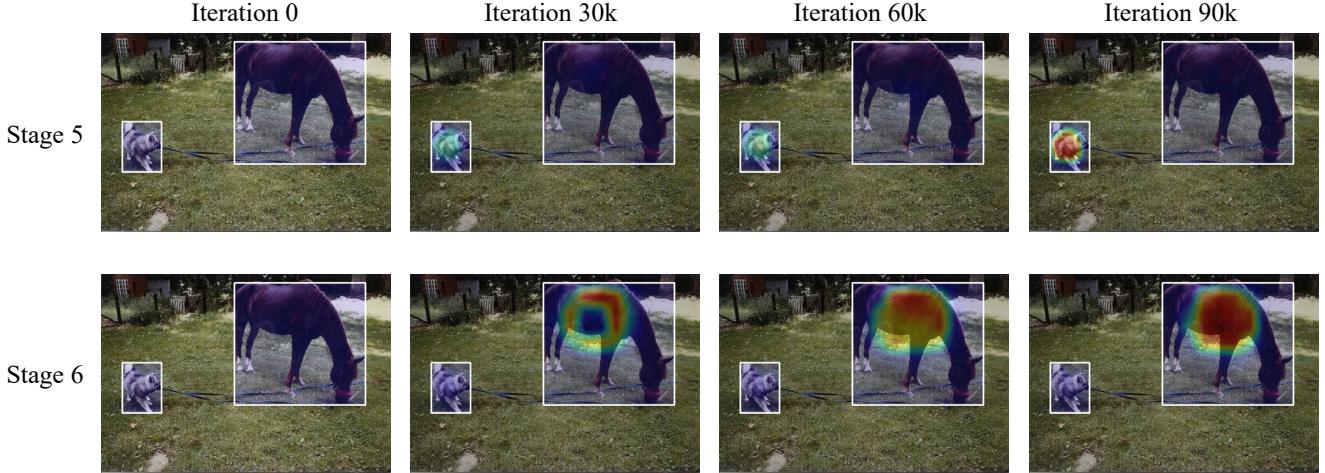


Figure 7. Illustration of confidence weighting evolution from iteration 0 to 90k. We select the most activated stages for different objects

Method	Iteration	$AP$	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
<b>ResNet-101</b>							
RetinaNet [11]	135k	39.1	59.1	42.3	21.8	42.7	50.2
FCOS [11]	180k	41.5	60.7	45.0	24.4	44.8	51.6
FreeAnchor [27]	180k	43.1	62.2	46.4	24.5	46.1	54.8
SAPD [30]	180k	43.5	63.6	46.5	24.9	46.8	54.6
ATSS [26]	180k	43.6	62.1	47.4	<b>26.1</b>	47.0	53.6
AutoAssign (Ours)	180k	<b>44.5</b>	<b>64.3</b>	<b>48.4</b>	25.9	<b>47.4</b>	<b>55.0</b>
<b>ResNeXt-64x4d-101</b>							
FCOS* [20]	180k	44.7	64.1	48.4	27.6	47.5	55.6
FreeAnchor [27]	180k	44.9	64.3	48.5	26.8	48.3	55.9
SAPD [30]	180k	45.4	65.6	48.9	27.3	48.7	<b>56.8</b>
ATSS [26]	180k	45.6	64.6	49.7	<b>28.5</b>	48.9	55.6
AutoAssign (Ours)	180k	<b>46.5</b>	<b>66.5</b>	<b>50.7</b>	28.3	<b>49.7</b>	56.6
<b>ResNeXt-64x4d-101-DCN</b>							
SAPD [30]	180k	47.4	67.4	51.1	28.1	50.3	61.5
ATSS [26]	180k	47.7	66.5	51.9	29.7	50.8	59.4
AutoAssign (Ours)	180k	48.3	67.4	52.7	29.2	51.0	60.3
AutoAssign (Ours) <sup>†</sup>	180k	49.5	68.7	54.0	29.9	52.6	62.0
AutoAssign (Ours) <sup>†‡</sup>	180k	<b>52.1</b>	<b>69.6</b>	<b>58.0</b>	<b>33.9</b>	<b>54.0</b>	<b>64.0</b>

Table 6. Performance comparison with state-of-the-art one-stage detectors on MS COCO 2017 *test-dev* set. All results listed adopt multi-scale training. <sup>†</sup> indicates multi-scale training with wider range [480, 960] used in [27]. <sup>‡</sup> indicates multi-scale testing. \* indicates improved versions

aspect ratios) unchanged and only adjust the training settings following the common paradigm of each dataset.

Results are shown in Table 7. We can see that the performance of other methods with fixed or partly fixed assigning strategies are unstable on different datasets. Although they achieve excellent performance on certain datasets, their accuracies on the other dataset may be worse. This proves that the label assignment strategy of these methods has a low robustness and need to be adjusted cautiously. In contrast,

AutoAssign with differential label assignment can adapt to different datasets and achieve superior performance without any adjustment.

## 5. Conclusions

In this paper, we propose a differentiable label assignment strategy named AutoAssign. It tackles label assignment in a fully data-driven manner by automatically determine the positives/negatives in both spatial and scale dimen-

Method	PASCAL VOC			Objects365			WiderFace		
	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>
RetinaNet [11]	55.4	81.0	60.1	18.4	28.4	19.6	46.7	83.7	47.1
FCOS* [20]	55.4	80.5	61.1	20.3	29.9	21.9	48.1	87.1	48.4
FreeAnchor [27]	56.8	81.1	62.1	21.4	31.5	22.8	46.3	81.6	47.5
ATSS [26]	56.6	80.7	62.6	20.7	30.0	22.4	48.9	87.1	49.7
AutoAssign (Ours)	<b>57.9</b>	<b>81.6</b>	<b>64.1</b>	<b>21.6</b>	<b>31.7</b>	<b>23.2</b>	<b>49.5</b>	<b>88.2</b>	<b>49.9</b>

Table 7. Performance comparison with typical detectors on PASCAL VOC, Objects365 and WiderFace. \* indicates improved versions

sions. AutoAssign is built from scratch and has no traditional components, e.g., anchors, making it fully learnable. It achieves consistent improvement to all the existing sampling strategies by  $\sim 1\%$  AP with various backbones on MS COCO. Besides, extensive experiments on other datasets demonstrate that AutoAssign can conveniently transfer to other datasets and tasks without additional modification.

This new label assignment strategy also comes with new challenges, for example, current weighting mechanism is not simple enough and can be further simplified, which will be left for future work.

## References

- [1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [2] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *The IEEE International Conference on Computer Vision*, 2019.
- [3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [5] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *The IEEE International Conference on Computer Vision*, 2019.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [7] 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.
- [8] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision*, 2018.
- [9] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. *arXiv preprint arXiv:1912.05086*, 2019.
- [10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision*, 2017.
- [12] 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 *The European Conference on Computer Vision*, 2014.
- [13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *The European Conference on Computer Vision*, 2016.
- [14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [15] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.
- [18] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [19] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *The IEEE International Conference on Computer Vision*, 2019.
- [20] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *The IEEE International Conference on Computer Vision*, 2019.
- [21] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

- [22] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [23] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *Advances in Neural Information Processing Systems*, 2018.
- [24] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *The IEEE International Conference on Computer Vision*, 2019.
- [25] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *The ACM International Conference on Multimedia*, 2016.
- [26] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. *arXiv preprint arXiv:1912.02424*, 2019.
- [27] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *Advances in Neural Information Processing Systems*, 2019.
- [28] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
- [29] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [30] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. *arXiv preprint arXiv:1911.12448*, 2019.
- [31] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.