Confidence Propagation Cluster: Unleash Full Potential of Object Detectors

Yichun Shen* Wanli Jiang* Zhen Xu Rundong Li Junghyun Kwon
Siyi Li
NVIDIA
{ashen,williamj,zhenx,davidli,junghyunk,louli}@nvidia.com

Abstract

It's been a long history that most object detection methods obtain objects by using the non-maximum suppression (NMS) and its improved versions like Soft-NMS to remove redundant bounding boxes. We challenge those NMS-based methods from three aspects: 1) The bounding box with highest confidence value may not be the true positive having the biggest overlap with the ground-truth box. 2) Not only suppression is required for redundant boxes, but also confidence enhancement is needed for those true positives. 3) Sorting candidate boxes by confidence values is not necessary so that full parallelism is achievable.

In this paper, inspired by belief propagation (BP), we propose the Confidence Propagation Cluster (CP-Cluster) to replace NMS-based methods, which is fully parallelizable as well as better in accuracy. In CP-Cluster, we borrow the message passing mechanism from BP to penalize redundant boxes and enhance true positives simultaneously in an iterative way until convergence. We verified the effectiveness of CP-Cluster by applying it to various mainstream detectors such as FasterRCNN, SSD, FCOS, YOLOv3, YOLOv5, Centernet etc. Experiments on MS COCO show that our plug and play method, without retraining detectors, is able to steadily improve average mAP of all those state-of-the-art models with a clear margin from 0.3 to 1.9 respectively when compared with NMS-based methods.

1. Introduction

The occurrence of convolutional neural networks has brought in revolutionary improvements in various object detection tasks [10, 14, 24, 41]. Generally, two-stage/multi-stage detectors [4, 9, 15, 31, 51] can achieve higher accuracy, while one-stage detectors [1, 16, 23, 26, 30, 37, 38] take better accuracy-performance balance. Recently, other than achieving better state-of-the-art results and less inference cost, some research attentions are also paid to simplify training and inference pipelines. [21,36,46,52] got rid of the predefined anchors. [5,34,39,52,55] designed specific one-one label assignment strategies to train end-to-end detection models without need of post-processing methods. [8, 52] make use of only one output feature map.

Nowadays some NMS-free methods have gained reasonable accuracies, but they still suffer from more or less sacrifice in accuracies, performance, training time and flexibility of design choices. Especially, when real-time inference is not required, the ensembles of detection models equipped with NMS are used to achieve better results [33, 54]. Besides, in autonomous vehicles systems, researchers usually apply NMS to combine objects detected from multiple sensors. Therefore the majority of those mainstream detectors [23, 31, 36, 37] still employ NMS or Soft-NMS [2] to remove redundant bounding boxes in inference stage. Standard NMS greedily suppresses all neighboring bounding boxes around the box with highest confidence value. Following this, researchers proposed several methods to improve the standard NMS accuracy [2, 19, 25, 50]. Among them, Soft-NMS [2] was proved to achieve general improvements for various detectors, while others are either designed for specific detectors or require retraining with specific tricks. In addition, some methods are proposed to parallelize the NMS [3, 49], while those methods still rely on confidence sorting in their pipelines.

With those NMS-based methods, all candidate boxes are firstly sorted according to their detection scores, and then the bounding box with the highest score in each cluster is selected as a representative. Other objects with slightly lower scores are simply thrown away or assigned with a smaller confidence, as does not make full use of relations between candidate boxes.

In this paper, we aim to replace the NMS-based methods with a better clustering framework (CP-Cluster) for object detectors, as is able to achieve better accuracy and meanwhile fully parallelizable. As illustrated by Fig. 1, CP-Cluster firstly constructs a graph set from all candidate boxes based on their overlaps, then both positive messages and negative messages are propagated among boxes belong-

*Equal contributions.

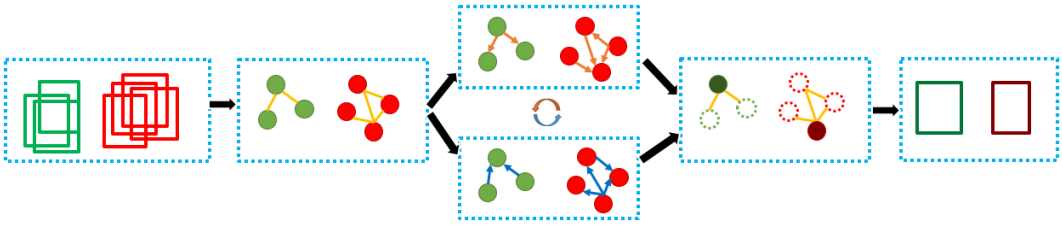


Figure 1. Overall pipeline of CP-Cluster. CP-Cluster converts all candidate boxes from an object detector into a set of graphs. Positive messages (blue arrows) and negative messages (orange arrows) are propagated within each graph iteratively, amplifying true positives and suppressing redundant boxes simultaneously.

ing to the same graph to tune each box's confidence value until convergence. In detail, to conquer the deficiencies of NMS-based methods, CP-Cluster is sophisticatedly designed to incorporate below strategies:

1) To make full use of relationships between candidate boxes, we propagate messages among them to tune their confidence values. Specifically, CP-Cluster generates positive messages to enhance true positive boxes and composes negative messages to penalize redundant boxes simultaneously.

2) To further maximize the confidence margin between true positives and redundant boxes, the confidence message propagations are performed multiple times iteratively.

3) To achieve full parallelism, the message propagation is restricted within neighboring candidate boxes, so that each candidate box manages to update itself independently.

We summarize our contributions as below:

1) We propose a new fully parallelizable clustering framework (CP-Cluster) applicable for all object detectors which require post-processing, and this new clustering framework outperforms NMS-based methods in accuracies.

2) We apply CP-Cluster to various mainstream detectors without retraining them, including FasterRCNN [31], SSD [26], FCOS [36], yolov5 [37] etc. On MS COCO, experimental results show general improvement for all mainstream detectors by just setting CP-Cluster as post-processing step.

3) By applying CP-Cluster to CenterNet [52], we show that some of NMS-free detectors can also be explicitly improved by this clustering framework.

To our knowledge, after Soft-NMS [2], CP-Cluster is the only bounding box clustering method which manages to achieve general improvements on most of mainstream object detectors in a plug and play manner. Furthermore, it shows huge potential to be applied in real-time tasks due to its full parallelism.

2. Related Works

Two-stage object detection. Traditional object detection pipelines mostly employ the sliding window strategy, running a classifier on all ROIs. Early neural network based methods also follow this way, say the two-stage detectors [9, 12, 13, 31, 51]: Candidate ROIs are generated in the first stage, then are further classified in the second stage. Some subsequent works further improve the accuracy by importing multi-stage detection [4, 42], and [27] tries to build relationships between candidate ROIs with RNN. Generally, by employing hierarchical stages, those two-stage methods have the merits of high accuracy, but also suffer from high inference cost and complex training strategies.

One-stage object detection. One-stage detectors [1, 11, 16, 23, 26, 28–30, 36, 38] were proposed with the merits of simpler training pipeline and less inference cost. Some early one-stage detectors were not comparable with two-stage detectors in accuracy, but later works have hugely improved model quality by better training samples selections/assignment strategies [46,53], stronger neural network architectures [11, 30, 35, 40], more sophisticatedly designed loss functions [22, 23, 32, 48] and combination of all those techniques [1, 36–38, 47]. Latest methods like YOLO5 [37] have achieved both high accuracy as well as very low inference cost. One-stage and two-stage detectors are not always competing but can also co-work together as a stronger detector. For instance, most of those one-stage detectors can be integrated into a two-stage detection framework like FasterRCNN [31], working as the Region Proposal Network [51].

Simplified detectors. Recently, some research efforts are taken to further simplify the one-stage detectors. The first direction is to remove predefined anchor boxes during training, simplifying the positive and negative samples assignment strategy [5, 21, 34, 36, 52]. Secondly, some methods like CenterNet [52] and Yolof [8] only employ one output feature maps, but still achieve reasonable accuracies. Such simplification may benefit multi-task train-

ing, as it allows several tasks to share a same backbone. Thirdly, starting from keypoint-based detectors [20, 21, 52] and transformer-based detectors [5, 55], researchers start to investigate the possibility of end-to-end object detection without post-processing. Specifically, those methods rely on some carefully designed one-one assignment strategies, such as Hungary matching [5] and minimum cost assignments [34].

Non Maximum Suppression. Usually a one-one assignment strategy is necessary for an end-to-end detector. However, on the other hand, such strategy restricts detectors from further improving accuracy and reducing inference time cost. Hence, NMS still works as the most effective post-processing step for the majority of popular object detectors. Other than standard NMS, Soft-NMS [2] assigns lower confidence values for bounding boxes rather than removing them directly, which is more friendly to occlusion case. [25] makes use of the density to improve clustering quality specifically for pedestrian detection task. [17, 19] integrate specific tricks into the training progress to co-work with NMS. [18] converted NMS into a learnable neural network. [48] improved NMS by proposing a better overlap computation strategy. Also, there are some attentions paid on parallelizing the NMS [3, 49], while they still rely on confidence sorting so that they are not fully parallelizable.

Belief Propagation in computer vision. Graph-model based methods have a long history of being applied in computer vision tasks. Some stereo matching tasks [43, 45] make use of BP to smooth the disparity maps. For scene segmentation tasks, early versions of DeepLab [7] also employ BP as the post-processing step to generate fine-grained segmentation results. Recently, some face clustering methods are fully built upon graph theories to pinpoint face clusters [44]

Relations to previous methods. CP-Cluster differentiate from previous NMS-based methods on:

1. CP-Cluster is fully built upon graph models and confidence message propagations who no longer follow the framework of NMS.

2. CP-Cluster is the first bounding box clustering pipeline who tries to enhance true positives and penalize redundant boxes simultaneously.

3. CP-Cluster does not rely on sorting bounding boxes with confidence values so that full parallelism could be achieved.

Although implemented in different frameworks, CP-Cluster is also compatible with some tricks from previous NMS-based methods: 1) Box coordinates weighting such as [33, 50]. 2) Different overlaps calculation strategies such as CIOU [48].

3. Confidence Propagation Cluster

In this section, we discuss details of how CP-Cluster fuses candidate boxes step by step. We firstly describe how to convert the boxes clustering task to a graph model problem to maximize the confidence margin between true postitives and redundant boxes. Then we discuss the details of how positive messages and negative messages are composed with heuristics from box distributions to update each candidate box.

3.1. General Clustering Pipeline

Building MRFs for bounding boxes. To describe the neighboring relationships between predicted bounding boxes, we create connections between bounding boxes according to their IOUs and then generalize them into Markov Random Field (MRF) graphs. For an object detector model, $\mathcal{B} = \{b_1, b_2, b_3, ...\}$ is the raw bounding box set from model output before post-processing. For each box pair $(b_i, b_j \in \mathcal{B})$, we draw an undirected edge between them if their IOU is greater than θ, generating a set of MRFs $\mathcal{G} = \{g_1, g_2, ...\}$. For each graph $g_i \in \mathcal{G}$, we define $\mathcal{E}_{g_i}$ as its edge set and $\mathcal{V}_{g_i}$ as its node set. For a box $b_i \in \mathcal{V}_{g_n}$, its neighboring node set $\mathcal{N}_{b_i}$ accommodates all nodes connected to b_i in g_n.

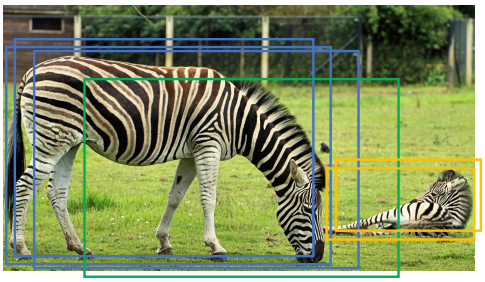
Fig. 2 is an example of how $\mathcal{G}$ is generated from $\mathcal{B}$ with $\theta = 0.6$, where $\mathcal{B} = \{A, B, C, D, E, F\}$ and $\mathcal{G} = \{g_1, g_2\}$. In detail, $\mathcal{V}_{g_1} = \{A, B, C, D\}$, $\mathcal{E}_{g_1} = \{(A, B), (B, C), (A, C), (C, D)\}$, $\mathcal{V}_{g_2} = \{E, F\}$, $\mathcal{E}_{g_2} = \{(E, F)\}$. Taking the box $A \in \mathcal{B}$ for example, its neighboring nodes $\mathcal{N}_A$ is $\{B, C\}$. From Fig. 2, it's noticeable that the number of graphs in $\mathcal{G}$ is same to the number of target boxes, while such equivalence doesn't hold true when two heavily occluded ground-truth boxes have overlap greater than θ.

Probabilistic objective. Given a bounding box $b_i \in \mathcal{B}$, we define $\hat{\mathbf{P}}(b_i) = \hat{\mathbf{P}}(b_i|\mathcal{N}_{b_i}, \overline{b_i})$ to be the confidence value of b_i from model output given its neighboring boxes and itself, thus the objective of the clustering process can be defined as:

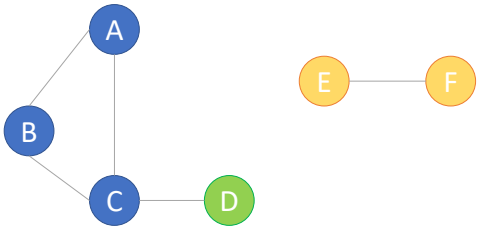
$$\hat{\mathbf{P}}(b_i) = \hat{\mathbf{P}}(b_i|\mathcal{N}_{b_i}, \overline{b_i}) = \begin{cases} 1.0 & b_i \in \mathcal{B}_p \\ 0.0 & b_i \in \mathcal{B}_n \end{cases} \quad (1)$$

where $\mathcal{B}_p$ stands for the set holding true positive candidate boxes having largest overlaps with ground-truth boxes, and $\mathcal{B}_n$ is the set of redundant bounding boxes. $\overline{b_i}$ is the observed confidence of b_i from object detectors. Equation (1) is targeted to maximize the confidence values of true positives, and meanwhile minimize the confidence values of redundant boxes. Compared with the objective of traditional NMS, CP-Cluster's objective is different in three aspects:

1. NMS-based clustering methods assume that the box of largest confidence value is always the best choice to be



(a) Example of raw detection results before clustering.



(b) Graph set generated by overlaps of bounding boxes.

Figure 2. Example of building MRF from bounding boxes by their IOUs($\theta = 0.6$)

selected, but in Equation (1) this assumption doesn't hold all the time.

2. We should not only suppress those redundant boxes, but also need to enhance the confidence value of those true positives.

3. Each candidate box is only impacted by its neighboring bounding boxes.

Clustering pipeline. In our task, unlike the typical case of belief propagation, neighboring bounding boxes not only smooth each other, but also compete with each other. Hence, we borrow the idea of iterative message passing from belief propagation but generate the messages by heuristics of bounding box distributions instead of traditional ways in BP such as sum-product or max-product. Specifically, we design the positive message $\mathbf{M_p}$ to reward those true positives, and negative messages $\mathbf{M_n}$ to penalize those redundant boxes. Both $\mathbf{M_p}$ and $\mathbf{M_n}$ only update confidence values of the bounding boxes by default.

In Algorithm 1, the graph model construction step (line 2) is similar to overlap matrix calculation step in traditional NMS. F_{gp} is the function to generate positive messages by $\mathcal{G}$ (Sec. 3.2), and F_{gn} generates negative messages (Sec. 3.3). Line 8 indicates that θ will be increased by λ in each iteration where λ is always positive, leading to incremental IOU threshold during the iterative message passing process. The motivation behind this incremental overlap threshold is: higher overlap two bounding boxes have,

Algorithm 1 Confidence Propagation Clustering

Require: $\mathcal{B}, \theta, F_{gp}, F_{gn}$
1: **for** $iteration = 1, 2, \ldots, N$ **do**
2: Calculate $\mathcal{G}$ with θ
3: **for all** b_i in $\mathcal{B}$ **do**
4: $\mathbf{M_p(i)} \leftarrow F_{gp}(\mathcal{G})$ ▷ Positive msg in Sec. 3.2
5: $\mathbf{M_n(i)} \leftarrow F_{gn}(\mathcal{G})$ ▷ Negative msg in Sec. 3.3
6: $\hat{\mathbf{P}}(\mathbf{b_i}) \leftarrow \hat{\mathbf{P}}(\mathbf{b_i}) + \mathbf{M_p(i)} - \mathbf{M_n(i)}$
7: **end for**
8: $\theta \leftarrow \theta + \lambda$
9: **end for**

more reasonable one of them should be suppressed more than once. Furthermore, Algorithm 1 is fully parallelizable because the confidence value updating step for each box is completely independent.

Fig. 3 is an example of comparison between standard NMS and CP-Cluster. Images in the first row and second row are generated by same Yolov5 model but clustered by standard NMS and CP-Cluster separately and visualized with a constant confidence threshold ($conf > 0.4$). Compared with output boxes from NMS, CP-Cluster not only obtains more objects but also generates higher confidence values for those true positive boxes.

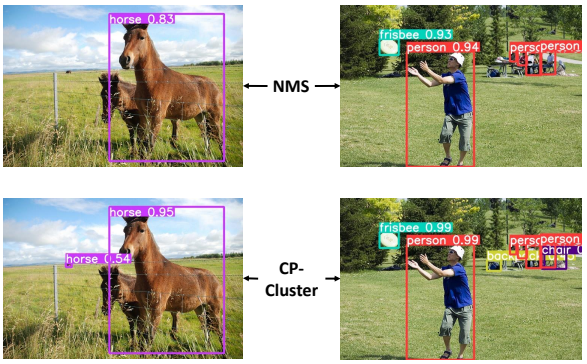


Figure 3. Example of how CP-Cluster enhances true positives and removes redundant boxes in the same time.

3.2. Positive Messages Generation

One key target of Equation (1) is to increase the rank of true positive candidate boxes. For a specific box b_i, positive messages are generated from its neighboring nodes $\mathcal{N}_{b_i}$ to increase $\hat{\mathbf{P}}(b_i)$.

Weaker friends aggregation (WFA). In contrast to high confidence candidate boxes, low confidence boxes are one of the least engaged in traditional post-processing pipelines. As more firewood produce stronger flame, we consider that those low confidence boxes are sometimes evidences to prove their stronger neighbors to be true positives. With

a bounding box $b_i \in \mathcal{V}_{g_n}$, his weaker friend set $\mathcal{W}_{b_i}$ is a subset of its neighbors $\mathcal{N}_{b_i}$, where $IOU(b_j, b_i) > \theta_n$ and $\hat{\mathbf{P}}(b_j) < \hat{\mathbf{P}}(b_i)$ for each $b_j \in \mathcal{W}_{b_i}$. Usually θ_n is greater than the overlap threshold θ in Algorithm 1, saying that only close enough neighbors can be treated as b_i's friends. Specifically, we found that the enhancement of a bounding box is mostly affected by below two factors:

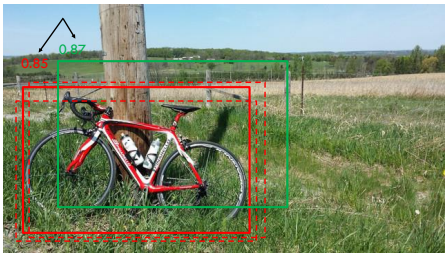
1. The number of its weaker friends, where such friends indicate stronger enhancement motivation.

2. The confidence value of its weaker friends. As more friends with high confidence values are evidences to prove that the box itself is true positive. After trying many options, the maximum confidence value of its weaker friends is proved to work best in Equation (2).

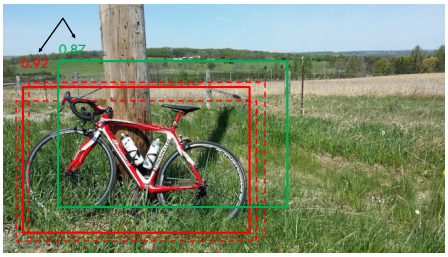
Therefore, we give the definition of positive message generation for a box b_i (Line 4 of Algorithm 1) as below:

$$\mathbf{M_p}(\mathbf{i}) \leftarrow \frac{Q}{Q+1} * (1 - \hat{\mathbf{P}}(b_i)) * \max_{\hat{b} \in \mathcal{W}_{b_i}} \hat{\mathbf{P}}(\hat{b}) \qquad (2)$$

where Q is the number of b_i's weaker friends, and $(1 - \hat{\mathbf{P}}(b_i))$ is the normalization term to ensure that the maximum value of $\hat{\mathbf{P}}(b_i)$ won't be greater than 1.0 after applying the positive message.



(a) Before WFA.



(b) After WFA.

Figure 4. Confidence value of the bounding box (red solid box) with more weaker friends is enhanced after WFA.

Fig. 4 is an example on how WFA tune confidence values of bounding boxes. The red solid box is enhanced during the progress of WFA because it has many weaker friends around (red dashed boxes), while the green solid box is not impacted by positive message update due to lack of weaker friends.

SNMS-WFA. To verify the effectiveness of our positive message generation step separately, we integrated the weaker friends aggregation step into standard Soft-NMS, leading to SNMS-WFA. Specifically, we perform Equation (2) to amplify those important boxes before their weaker friends are suppressed. In Sec. 4 we will also discuss experimental results of SNMS-WFA and compare it with CP-Cluster.

3.3. Negative Messages Generation

Other than enhancing true positive boxes, suppressing redundant boxes is another objective as indicated by Equation (1).

Given a bounding box $b_i \in \mathcal{V}_{g_n}$, his stronger neighbors $\mathcal{S}_{b_i}$ is a subset of $\mathcal{N}_{b_i}$, where $IOU(b_j, b_i) > \theta$ and $\hat{\mathbf{P}}(b_j) > \hat{\mathbf{P}}(b_i)$ for each $b_j \in \mathcal{S}_{b_i}$. In each iteration of Algorithm 1, if a bounding box's stronger neighbor set is not empty, it will be suppressed by one of its stronger neighbor $b_j \in \mathcal{S}_{b_i}$. As to which bounding box is selected to suppress b_i, we design the negative impact factor $\mathcal{T}_{(b_j,b_i)}$ from $b_j \in \mathcal{S}_{b_i}$ to b_i as below:

$$\mathcal{T}_{(b_j,b_i)} \leftarrow \alpha * \hat{\mathbf{P}}(b_j)/\hat{\mathbf{P}}(b_i) + (1 - \alpha) * IOU(b_j, b_i)/\theta \qquad (3)$$

In Equation (3), when we set $\alpha = 1.0$, the box with largest confidence value in $\mathcal{S}_{b_i}$ is selected. On the contrary, the nearest stronger neighbor with maximum $IOU(b_i, b_j)$ is selected from $\mathcal{S}_{b_i}$ if $\alpha = 0.0$.

Another problem between b_i and $\mathcal{S}_{b_i}$ is worthwhile to be discussed: How many times is a certain box $b_j \in \mathcal{N}_{b_i}$ allowed to suppress b_i? For flexibility, we define the suppression counting matrix $\mathbf{SUP}_{j,i}$ to count the times b_j has suppressed b_i, and ζ is the maximum suppression time. We will discuss more details about how to configure ζ in Sec. 4.1.

Based on above discussion, the negative message (Line 5 of Algorithm 1) for a box b_i is generated by below equation:

$$\mathbf{M_n}(\mathbf{i}) \leftarrow \hat{\mathbf{P}}(b_i) * IOU(b_i, \underset{b_j \in \mathcal{N}_{b_i}, \mathbf{SUP}_{j,i} <= \zeta}{\arg\max} \mathcal{T}_{(b_j,b_i)}) \qquad (4)$$

Where $\mathbf{SUP}_{j,i}$ is used to restrict the times for b_i to be suppressed by b_j, and the box with maximum negative impact factor will be picked up to penalize b_i.

3.4. More Details Behind Confidence Propagation

Message Flow Directions. As is shown in Sec. 3.2, positive messages are passed from weaker boxes to stronger boxes. On the contrary, negative messages flow from stronger boxes to weaker boxes as discussed in Sec. 3.3.

Parallelism We have already briefly discussed the parallelism of Algorithm 1 in Sec. 3.1. Specifically, as each candidate box is only impacted by his neighbors within one

iteration, $\mathcal{K}$ threads can be created to handle each box in parallel, where $\mathcal{K}$ is the number of candidate boxes. Actually, we can further improve the parallelism by combining the graph generation step and message propagation, and $\mathcal{K}$*$\mathcal{K}$ threads can be created to handle the message passing between two boxes.

4. Experiments

Dataset. We conduct experiments on COCO 2017 dataset [24]. Evaluation results are reported on the COCO val and test-dev dataset.

Experiments. We didn't train new models but directly downloaded models from model zoo for those mainstream detectors. Then we replace the NMS-based post-process step with CP-Cluster and run the evaluation on COCO val and test-dev dataset.

Baselines. We take standard NMS and Soft-NMS as baselines to compare with our CP-Cluster. We also performed exhaustive experiments on other plug-and-play NMS versions like weighted-NMS [50] and Clustered-NMS [49], but usually they cannot compete with Soft-NMS or even have negative impacts on some detectors. For other NMS-based methods like [17, 18, 25], they either require retraining models with extra architecture modifications, or are targeted for special tasks. To save space, we only report baseline metrics for standard NMS and Soft-NMS. In addition, we also report experimental results on WFA-SNMS to prove the effectiveness of our positive message generation strategy separately.

4.1. Ablation Studies

All experiments in this section are performed with Yolov5s model downloaded from Yolov5 model zoo. Fig. 5 shows how mAP, AP50, AP75 are impacted by different hyperparameters separately.

Number of iterations. CP-Cluster provides an iterative way to enhance true positive boxes and meanwhile suppress redundant boxes. As illustrated by red columns in Fig. 5a, usually 2 iterations have already been good enough to run the clustering process into convergence.

Negative Impact Factor. In the negative message generation step, negative impact factor is designed to pick up the most appropriate strong neighbor to penalize a box b_i if necessary. The strong neighbor selection criterion is controlled by the parameter α. After trying different options, we found the best result is usually achieved when we apply different α in each iteration. In detail, we pick up the box with largest confidence value ($\alpha = 1.0$) in the first iteration, while in the second iteration we select the box of biggest overlap with b_i ($\alpha = 0.0$).

Incremental IOU threshold. From Algorithm 1, the parameter λ is used to increment the overlap threshold in each iteration. Intuitively, higher is λ, less boxes will be penalized in the second iteration. From green columns in Fig. 5, a smaller λ leads to better AP50 but worse AP75. In below experiments, we set $\lambda = 0.2$ to achieve the most balanced improvements on all buckets.

Thresholds to select weaker friends. In the positive message generation step, the parameter θ_n decides how many boxes are incorporated in the weaker friend set of b_i. Specifically, larger θ_n means less friends of b_i. As shown by blue columns in Fig. 5, the best accuracy can usually be achieved when θ_n is around 0.8.

Maximum suppression time. In equation (4), ζ is used to decide the maximum times a box b_i can be suppressed by b_j. From yellow columns in Fig. 5, $\zeta = 2$ is beneficial to AP50, while we can get slightly better AP75 when $\zeta = 1$. As we found that $\zeta = 2$ leads to more stable improvements in most cases, we adopt this setting in our following experiments.

4.2. Experiments in MMDetection

MMDetection [6] is a toolbox with a collection of popular object detector implementations. We implemented our CP-Cluster in mmcv, which is a tool library used by MMDetection.

Since CP-Cluster doesn't require retraining models, we download those popular models from MMDetection model zoo and get them evaluated along with CP-Cluster. Experimental results are reported on both COCO val and test-dev dataset in Tab. 1.

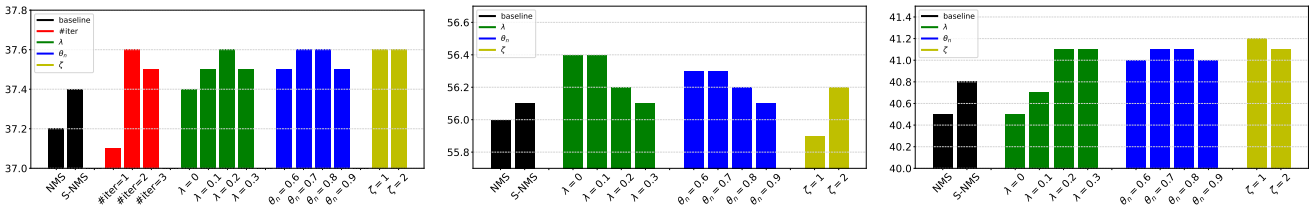
From Tab. 1, with CP-Cluster, the average mAP of all those popular models are improved by $0.3 - 0.7$ compared with standard NMS. And compared with Soft-NMS, CP-Cluster still achieved $0.2 - 0.6$ improvements on average mAP.

4.3. Experiments With Yolov5

Recently Yolov5 [37] is getting popular due to its extreme balance in accuracy and time cost.

In our experiments, we download the pretrained checkpoints (v6 on 1/10/2022) and pair them with our CP-Cluster. For default NMS, we reproduce the evaluation result on COCO test-dev with suggested IOU threshold $\theta = 0.65$. While for CP-Cluster, we employ a slightly smaller $\theta = 0.6$.

Experimental results are reported on COCO test-dev dataset in Tab. 2, which shows that CP-Cluster manages to achieve $0.3 - 0.4$ improvements on average mAP compared with standard NMS. To save table size, we don't report evaluation results for Soft-NMS and SNMS-WFA. In fact, Soft-NMS fails to make explicitly positive impact on most of Yolov5 models, while SNMS-WFA can achieve similar improvements compared with CP-Cluster.



(a) mAP. (b) AP50. (c) AP75.

Figure 5. Accuracies with different hyperparameters on Yolov5s.

MAP (val/test-dev)	nms	soft-nms	snms-wfa	cp-cluster
ssd512	29.5/29.6	29.8/29.9	30.0/30.0	**30.1/30.1**
frcnn-r50fpn	38.4/38.7	39.0/39.2	39.1/39.3	**39.2/39.4**
fcos-x101	42.7/42.8	42.7/42.8	42.8/42.9	**43.0/43.2**
retina-r50fpn	37.4/37.7	37.5/37.9	37.7/38.2	**38.1/38.4**
yolov3	33.5/33.5	33.8/33.8	33.6/33.7	**34.1/34.1**
yolof	37.5/37.8	37.6/37.8	38.0/38.4	**38.1/38.4**
autoassign-fpn50	40.4/40.6	40.5/40.7	40.6/40.8	**41.0/41.2**

Table 1. CP-Cluster with various popular models in MMDetection on COCO val/test-dev.

4.4. Experiments With Keypoint-based Detectors

Keypoint-based object detectors [20, 21, 52] are among the earliest attempts to remove the NMS post-process step. Specifically, they replaced the NMS with a simple maxpooling operation to pick up peak points in predicted heatmaps. As discussed in [52], NMS methods show positive impacts for some Centernet models but lead to negative results for others.

In our experiments, we download the pretrained models directly from official Centernet repo [52]. For those non-maxpooling based experiments, the maxpooling step is replaced by Soft-NMS and CP-Cluster respectively with IOU threshold $\theta = 0.5$. Experimental results on COCO test-dev are reported in Tab. 3, where "dla34_flip_scale" means the model with "dla34" arch, augmented by rescaling and flipping.

Compared with default maxpooling post-processing step, all Centernet models are improved with a margin $0.6 - 1.9$ on average mAP when paired with CP-Cluster, including those models with multi-scale and flip augmentations. Furthermore, Soft-NMS method can also improve the accuracy of Centernet when they replaced maxpooling in those experiments on single models, while it has negative impacts in multi-scale fusion experiments. The stable improvements provided by CP-Cluster on multi-scale tests show its potential as a better cluster to handle bounding boxes from multiple models.

4.5. Experiments for Instance Segmentation

Instance segmentation methods are usually built upon object detectors to gain accurate instance area for detected objects. Still with MMDetection, we apply CP-Cluster to various MaskRCNN models from model zoo, and experimental results on COCO test-dev are shown in Tab. 4. Compared with standard NMS, CP-Cluster shows considerable improvements on both BOX-AP as well as MASK-AP. Although Soft-NMS and CP-Cluster achieve similar accuracy on the X101 model, CP-Cluster outperforms Soft-NMS on all other more lightweight MaskRCNN models.

4.6. Runtime Measurements

We measure the runtime cost for both CPU and GPU versions of CP-Cluster along with Yolov5 framework. CP-Cluster is compared with CPU Soft-NMS in mmcv and GPU NMS in torchvision. Note that CP-Cluster does not rely on sorting bounding boxes by their confidence values. However, to make the APIs consistent with torchvision, an extra box sorting step is appended at the end of our CP-Cluster to make sure that true positive boxes are returned in descending order by their confidence values. When measuring runtime of CP-Cluster on GPU, we exclude the step of box sorting. The measurements are run on a workstation with a 9th-Gen Core-i7 CPU and a Titan-V GPU.

As shown in Tab. 5, our GPU implementation of CP-Cluster($Iter = 2$) is comparable to the NMS implementation in torchvision. Actually, we are still working on further optimizing the GPU implementation as it will benefit from more sophisticatedly designed CUDA tricks.

Model	Method	AP	AP50	AP75	APS	APM	APL	AR100
s_640	nms	37.1	55.7	40.2	20.1	41.5	45.2	55.1
	cp-cluster	**37.4**	**56.0**	**40.8**	**20.3**	**41.9**	**45.5**	**57.2**
m_640	nms	45.5	64.0	49.7	26.6	50.0	56.6	62.2
	cp-cluster	**45.8**	**64.2**	**50.3**	**26.9**	**50.3**	**56.9**	**64.3**
l_640	nms	49.0	67.3	53.4	29.9	53.4	61.3	64.6
	cp-cluster	**49.3**	**67.4**	**53.9**	**30.1**	**53.7**	**61.5**	**67.1**
x_640	nms	50.7	68.8	55.1	31.9	54.9	63.4	66.6
	cp-cluster	**51.1**	**68.9**	**55.7**	**32.3**	**55.2**	**63.5**	**68.7**
s6_1280	nms	44.3	**62.7**	48.8	27.0	48.3	53.6	62.3
	cp-cluster	**44.6**	**62.7**	**49.4**	**27.3**	**48.5**	**54.1**	**64.4**
m6_1280	nms	51.2	**69.2**	56.2	33.5	55.1	62.1	68.1
	cp-cluster	**51.5**	**69.2**	**56.7**	**33.7**	**55.4**	**62.5**	**70.2**
l6_1280	nms	53.8	**71.6**	58.9	36.3	57.8	64.9	70.3
	cp-cluster	**54.1**	**71.6**	**59.4**	**36.6**	**58.1**	**65.3**	**72.4**
x6_1280	nms	55.1	**72.8**	60.4	37.8	58.9	66.5	71.5
	cp-cluster	**55.5**	**72.8**	**60.9**	**38.1**	**59.3**	**66.8**	**73.4**

Table 2. CP-Cluster with 8 yolov5 models on COCO test-dev.

Model	Method	AP	AP50	AP75	APS	APM	APL	AR100
dla34	maxpool	37.3	55.1	40.7	18.6	41.1	49.2	55.8
	soft-nms	38.1	57.0	41.1	18.7	40.8	50.7	56.8
	cp-cluster	**39.2**	**57.9**	**43.0**	**20.4**	**42.4**	**51.3**	**58.0**
dla34_ flip_scale	maxpool	41.7	60.6	45.1	21.7	44.0	56.0	60.4
	soft-nms	40.6	58.7	43.8	21.2	43.1	54.8	57.4
	cp-cluster	**43.3**	**61.8**	**47.6**	**24.3**	**45.9**	**56.4**	**62.7**
hg104	maxpool	40.2	59.1	43.8	22.5	43.4	50.8	56.0
	soft-nms	40.6	58.7	44.5	23.1	43.9	**51.0**	57.4
	cp-cluster	**41.1**	**59.9**	**45.0**	**24.4**	**44.6**	**51.0**	**58.4**
hg104_ flip_scale	maxpool	45.2	64.1	49.3	26.7	47.2	57.9	63.2
	soft-nms	44.3	62.8	48.3	26.2	46.5	57.0	60.8
	cp-cluster	**46.6**	**65.0**	**51.5**	**28.9**	**49.0**	**58.3**	**65.1**

Table 3. CP-Cluster for Centernet on COCO test-dev.

	NMS		Soft-NMS		CP-Cluster	
	Box AP	Mask AP	Box AP	Mask AP	Box AP	Mask AP
MaskRCNN_R50_3X	41.5	37.7	42.0	37.8	**42.2**	**38.1**
MaskRCNN_R101_3X	43.1	38.8	43.6	39.0	**43.7**	**39.2**
MaskRCNN_X101_3X	44.6	40.0	**45.2**	**40.2**	**45.2**	**40.2**

Table 4. CP-Cluster for MaskRCNN on COCO test-dev.

Runtime(ms)	NMS	Soft-NMS	CP(Iter=1,2,3)		
CPU(mmcv)	N/A	11.1	32	52	63
GPU	1.4	N/A	1.0	1.3	1.5

Table 5. Runtime Comparison of CP-Cluster.

5. Conclusion

In this work, we have presented a new graph model based bounding box clustering framework (**CP-Cluster**), which is fully parallelizable. This framework can work as a general post-processing step for all object detectors, replacing traditional NMS-based methods. Compared with NMS and Soft-NMS, CP-Cluster is able to achieve better accuracy on MS COCO dataset when applied to the same model.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1, 2

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 1, 2, 3

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019. 1, 3

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1, 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 2, 3

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 3

[8] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13039–13048, 2021. 1, 2

[9] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 1, 2

[10] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 1

[11] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 2

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 1

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 1, 2

[17] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2888–2897, 2019. 3, 6

[18] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6469–6477, 2017. 3, 6

[19] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–799, 2018. 1, 3

[20] Shiyi Lan, Zhou Ren, Yi Wu, Larry S Davis, and Gang Hua. Saccadenet: A fast and accurate object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10397–10406, 2020. 3, 7

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 1, 2, 3, 7

[22] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020. 2

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2

[24] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 6

[25] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6459–6468, 2019. 1, 3, 6

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[27] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards

balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019. 2

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[29] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[30] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 2

[33] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, pages 1–6, 2021. 1, 3

[34] Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9934–9944. PMLR, 2021. 1, 2, 3

[35] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1, 2

[37] Ultralytics. Yolov5. 2021. 1, 2, 6

[38] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021. 1, 2

[39] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15849–15858, 2021. 1

[40] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[41] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018. 1

[42] Jia Xiang and Gengming Zhu. Joint face detection and facial expression recognition with mtcnn. In *2017 4th international conference on information science and control engineering (ICISCE)*, pages 424–427. IEEE, 2017. 2

[43] Xueqin Xiang, Mingmin Zhang, Guangxia Li, Yuyong He, and Zhigeng Pan. Real-time stereo matching based on fast belief propagation. *Machine vision and applications*, 23(6):1219–1227, 2012. 3

[44] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2298–2306, 2019. 3

[45] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1458–1465. IEEE, 2010. 3

[46] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020. 1, 2

[47] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018. 2

[48] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000, 2020. 2, 3

[49] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 2021. 1, 3, 6

[50] Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. Cad: Scale invariant framework for real-time object detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 760–768, 2017. 1, 3, 6

[51] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 1, 2

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 3, 7

[53] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020. 2

[54] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. Tph-yolov5: Improved yolov5 based on transformer prediction

head for object detection on drone-captured scenarios. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2778–2788, 2021. 1

[55] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 3