Correlation Loss: Enforcing Correlation between Classification and Localization

Fehmi Kahraman[*,1], **Kemal Oksuz**[*,1], **Sinan Kalkan**[†,1,2], **Emre Akbas**[†,1,2]

[1]Dept. of Computer Engineering, Middle East Technical University (METU), Ankara, Turkey
[2]METU Center for Robotics and Artificial Intelligence (ROMER), Ankara, Turkey
{fehmi.kahraman_01, kemal.oksuz, skalkan, eakbas}@metu.edu.tr

Abstract

Object detectors are conventionally trained by a weighted sum of classification and localization losses. Recent studies (e.g., predicting IoU with an auxiliary head, Generalized Focal Loss, Rank & Sort Loss) have shown that forcing these two loss terms to interact with each other in non-conventional ways creates a useful inductive bias and improves performance. Inspired by these works, we focus on the correlation between classification and localization and make two main contributions: (i) We provide an analysis about the effects of correlation between classification and localization tasks in object detectors. We identify why correlation affects the performance of various NMS-based and NMS-free detectors, and we devise measures to evaluate the effect of correlation and use them to analyze common detectors. (ii) Motivated by our observations, e.g., that NMS-free detectors can also benefit from correlation, we propose Correlation Loss, a novel plug-in loss function that improves the performance of various object detectors by directly optimizing correlation coefficients: E.g., Correlation Loss on Sparse R-CNN, an NMS-free method, yields 1.6 AP gain on COCO and 1.8 AP gain on Cityscapes dataset. Our best model on Sparse R-CNN reaches 51.0 AP without test-time augmentation on COCO test-dev, reaching state-of-the-art. Code is available at: https://github.com/fehmikahraman/CorrLoss.

1 Introduction

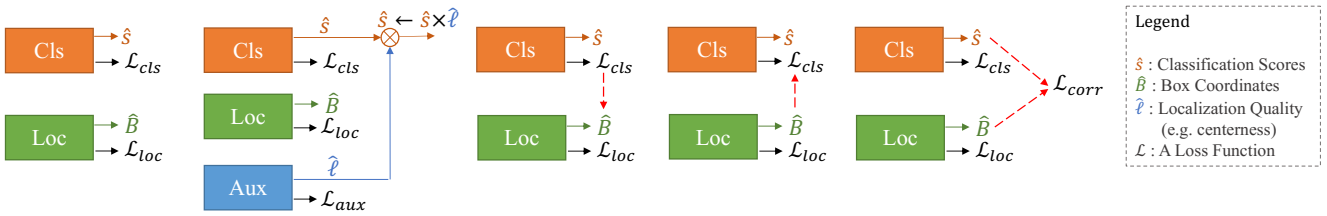
Most object detectors optimize a weighted sum of classification and localization losses during training. Results from recent work suggest that performance improves when these two loss functions are forced to interact with each other in non-conventional ways as illustrated in Fig. 1. For example, training an auxiliary (aux.) head to regress the localization qualities of the positive examples, e.g. centerness, IoU or mask-IoU, has proven useful (Jiang et al. 2018; Kim and Lee 2020; Tian et al. 2019; Zhang et al. 2020) (Fig. 1(b)). Other methods remove such auxiliary heads and aim directly to enforce correlation[1] in the classification or localization task during training; e.g., Average LRP Loss (Oksuz et al. 2020) weighs the examples in the localization task by ranking them with respect to (wrt.) their classification scores (Fig. 1(c)). Using localization quality as an additional supervision signal for classification has been more commonly adopted (Fig. 1(d)) (Li et al. 2020; Liu et al. 2021; Oksuz et al. 2021a; Zhang et al. 2021) in two main ways: (i) Score-based approaches aim to regress the localization qualities (Li et al. 2019, 2020; Zhang et al. 2021) in the classification score, and (ii) ranking-based approaches enforce the classifier to rank the confidence scores wrt. the localization qualities (Liu et al. 2021; Oksuz et al. 2021a).

Improving correlation seems to have a positive effect on performance of a variety of object detectors, as shown in Fig. 2. However, the effect of correlation on object detectors has not been thoroughly studied. We fill this gap in this paper and first identify that correlation affects the performance of object detectors at two levels: (i) *Image-level correlation*, the correlation between the classification scores and localization qualities (i.e., IoU for the rest of the paper) of the detections in a single image before post-processing, which is important to promote NMS performance, and (ii) *Class-level correlation*, the correlation over the entire dataset for each class after post-processing, which is related to the COCO-style Average Precision (AP). Moreover, we quantitatively define correlation at each level to enable analyses on how well an object detector captures correlation (e.g., β_{cls} in Fig. 2(a)). Then, we provide an analysis on both levels of correlation and draw important observations using common models. Finally, to better exploit correlation, we introduce a more direct mechanism to enforce correlation: *Correlation Loss*, a simple plug-in and detector-independent loss term (Fig. 1(e)), improving performance for a wide range of object detectors including NMS-free detectors, aligning with our analysis (Fig. 2(b)). Similar to the novel loss functions (Li et al. 2020; Oksuz et al. 2021a; Zhang et al. 2021), our Correlation Loss boosts the performance without an auxiliary head, but different from them, it is a simple plug-in technique that can easily be incorporated into any object detector, whether NMS-based or NMS-free.

Our main contributions are: **(1)** We identify how correlation affects NMS-based and NMS-free detectors, and design quantitative measures to analyze a detector wrt. correlation. **(2)** We analyze the effects of correlation at different levels on various object detectors. **(3)** We propose Correla-

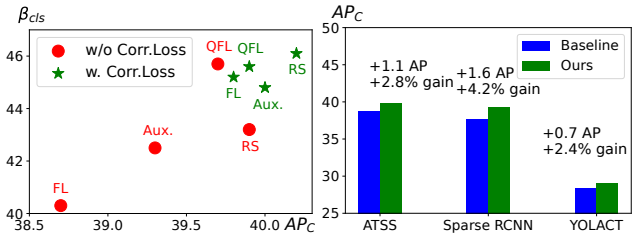
[*]These authors contributed equally.
[†]Equal contribution for senior authorship.

[1]In the rest of the paper, "correlation" will refer to the correlation between classification scores and IoUs.



(a) No correlation (b) Auxiliary Head (c) Novel Loc. Loss (d) Novel Cls. Loss (e) Correlation Loss (Ours)

Figure 1: Different ways of handling the classification and localization tasks from the perspective of correlation. **(a)** Conventional case of optimizing the two tasks independently (e.g., Chen et al. 2020; Sun et al. 2021b). **(b)** An additional auxiliary head predicts centerness (Zhang et al. 2020) or IoU (Jiang et al. 2018; Kim and Lee 2020), which introduces additional learnable parameters. **(c)** Novel loss functions replace the standard localization loss (Oksuz et al. 2020) or **(d)** novel classification loss (Li et al. 2020; Oksuz et al. 2021a) by more complicated ones to leverage correlation. **(e)** Our Correlation Loss explicitly optimizes a correlation coefficient. It is a simple, plug-in loss function which does not introduce additional parameters and has the flexibility to supervise classification or localisation head as well as both. Black and colored arrows respectively denote the loss functions (i.e., during training) & the network outputs (i.e., during inference).



(a) Detection vs. Correlation (b) Effect of our Corr. Loss

Figure 2: **(a)** Detection performance, measured by COCO-style AP ($\mathrm{AP_C}$) vs. correlation quality, measured by class-level correlation (β_{cls}, see Section 3.1 for details). The methods improving the correlation also improve $\mathrm{AP_C}$. Compare using aux. head, QFL, RS Loss with the baseline ATSS using Focal Loss (FL), all in red dots, to see the positive correlation between $\mathrm{AP_C}$ and β_{cls}. Our Correlation Loss explicitly optimizes a correlation coefficient and improves the performance ($\mathrm{AP_C}$) for all settings of ATSS owing to increasing β_{cls}, validating our hypothesis (compare green stars with red dots). **(b)** Our Correlation Loss is simple-to-use and improves NMS-based ATSS; NMS-free Sparse R-CNN; YOLACT, an instance segmentation method.

tion Loss as a plug-in loss function to optimize correlation explicitly. Thanks to its simplicity, our loss function can be easily incorporated into a diverse set of object detectors and improves the performance of e.g., Sparse R-CNN up to 1.6 AP and 2.0$\mathrm{AP_{75}}$, suggesting, for the first time, that NMS-free detectors can also benefit from correlation. Our best model yields 51.0 AP, reaching state-of-the art.

2 Background and Related Work

Object Detection Pipeline. We group object detectors wrt. their usage of NMS (Fig. 3 presents overview & notation):

1. NMS-based Detectors. To detect all objects with different scales, locations and aspect ratios; most methods (He et al. 2017; Kong et al. 2020; Law and Deng 2018; Lin et al. 2020; Ren et al. 2017; Tian et al. 2019; Zhang et al. 2020) employ a large number of object hypotheses (e.g., anchors, points), which are labeled as positive (a.k.a. foreground) or negative (a.k.a. background) during training, based on whether/how they match GT boxes (Zhang et al. 2020, 2019). In this setting, there is no restriction for an object to be predicted by multiple object hypotheses, causing duplicates. Accordingly, during inference, NMS picks the detection with the largest confidence score among the detections that overlap more than a predetermined IoU threshold to avoid duplicate detections.

2. NMS-free Detectors. An emerging research direction is to remove the need for doing NMS, simplifying the detection pipeline (Carion et al. 2020; Dai et al. 2021; Roh et al. 2022; Sun et al. 2021b,a; Zhu et al. 2021). This is achieved by ensuring a one-to-one matching between the GTs and detections, which supervises the detector to avoid duplicates in the first place.

Methods Enforcing Correlation. One common way to ensure correlation is to use an additional auxiliary head, supervised by the localization quality of a detection such as centerness (Tian et al. 2019; Zhang et al. 2020), IoU (Jiang et al. 2018), mask IoU (Huang et al. 2019) or uncertainty (He et al. 2019), during training. During inference, the predictions of the auxiliary head are then combined with those of the classifier to improve detection performance. Recent methods show that the auxiliary head can be removed, and either (i) the regressor can prioritize the positive examples (Oksuz et al. 2020) or (ii) the classifier can be supervised to prioritize detections with confidence scores. The latter is ensured either by regressing the IoUs by the classifier (Li et al. 2020; Zhang et al. 2021) or by training the classifier to rank confidence scores (Liu et al. 2021; Oksuz et al. 2021a) wrt. IoUs. Unlike these methods, TOOD (Feng et al. 2021) takes correlation into account mainly while designing the model, particularly the detection head, i.e., not the loss function.

Correlation Coefficients. Correlation coefficients measure the strength and direction of the "relation" between two sets, $X = \{x_1, ..., x_N\}$ and $Y = \{y_1, ..., y_N\}$. Different re-

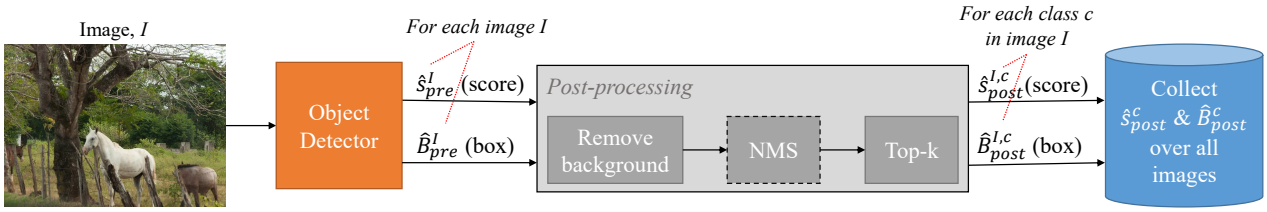


Figure 3: Object detection pipeline and notation. Given an input image, I, NMS-based detectors yield raw detections before post-processing, each of which has a predicted bounding box (BB) and an array of confidence scores over GT classes. We denote the confidence scores and the predicted BBs pertaining to the *positive detections*, i.e., the detections matching with GT objects during training, by $\hat{s}^I_{pre}$ and $\hat{B}^I_{pre}$, respectively. To obtain final detections, raw detections are post-processed in three steps: (i) Detections with low confidence scores, i.e., background, are removed, (ii) duplicates are eliminated by NMS, and (iii) top-k scoring detections are kept. As for these final detections, we denote the confidence scores and BBs of *true positive detections* for class c in a single image I by $\hat{s}^{I,c}_{post}$ and $\hat{B}^{I,c}_{post}$ respectively, and over the entire dataset by $\hat{s}^c_{post}$ and $\hat{B}^c_{post}$. As for NMS-free detectors; NMS, dashed gray box in post-processing, is excluded, hence post-processing is lighter.

lations are evaluated by different correlation coefficients: (i) *Pearson correlation coefficient*, denoted by $\alpha(\cdot,\cdot)$, measures the linear relationship between the sets, (ii) *Spearman correlation coefficient*, $\beta(\cdot,\cdot)$, corresponds to the ranking relationship and (iii) *Concordance correlation coefficient*, $\gamma(\cdot,\cdot)$, is more strict, measuring the similarity of the values and maximized when $x_i = y_i$ for all $i \in 1,...,N$. All correlation coefficients have a range of $[-1,+1]$ where positive/negative correlation corresponds to increasing/decreasing relation, while 0 implies no correlation between X and Y.

Comparative Summary. In this paper, we comprehensively identify and analyze the effect of explicitly correlating classification and localization in object detectors. Unlike other methods that also enforce correlation, some of which are tested only on a single architecture (Huang et al. 2019; Jiang et al. 2018; Tian et al. 2019), we propose a simple solution by directly optimizing the correlation coefficient, which is auxiliary-head free and easily applicable to *all* object detectors, whether NMS-based or NMS-free. Also, ours is the first to work on NMS-free detectors in this context.

3 Effects of Correlation on Object Detectors

This section presents why maximizing correlation is important for object detectors, introduces measures to evaluate object detectors wrt. correlation and provides an analysis on methods designed for improving correlation.

3.1 How Correlation Affects Object Detectors

Detectors are affected by correlation at two levels (Fig. 4):

Image-level Correlation. This level of correlation corresponds to the correlation between the classification scores and IoUs of the detections in a single image before post-processing, and accordingly, we measure it with the Spearman correlation coefficient[2], $\beta(\cdot,\cdot)$, averaged over images. Denoting the set of images to be evaluated by $\mathcal{I}$ and IoUs between the BBs of the positive detections ($\hat{B}^I_{pre}$, Fig. 3) and their associated GTs by IoU^I_{pre}, image-level correlation is measured as follows:

$$\beta_{img} = \frac{1}{|\mathcal{I}|}\sum_{I\in\mathcal{I}} \beta(\mathrm{IoU}^I_{pre}, \hat{s}^I_{pre}). \tag{1}$$

Maximizing image-level correlation is important for NMS-based detectors since NMS aims to suppress duplicates, i.e., to keep only a single detection for each GT when there is more than one. More particularly among overlapping detections (e.g., dark and light green detections in the detector output image in Fig. 4(a)), NMS picks the one with the larger score, and hence, if there is positive correlation between the confidence scores and IoUs of those overlapping detections, then the one with the best IoU (e.g., dark green detection in Fig. 4(a)) will survive and detection performance will increase.

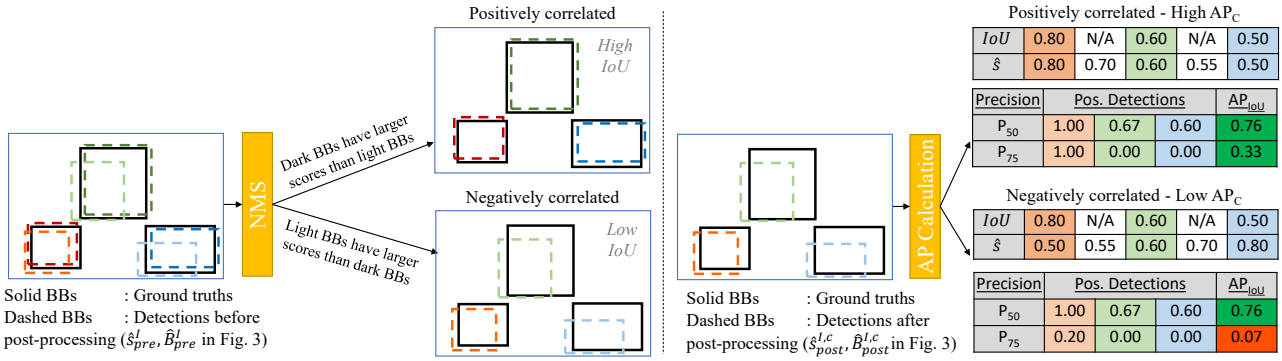
Class-level Correlation. This level of correlation indicates the correlation between the classification scores and IoUs of the detections obtained after post-processing for each class. Since class-level correlation is related to COCO-style AP, $\mathrm{AP_C}$, we average $\beta(\cdot,\cdot)$ over classes to be consistent with the computation of $\mathrm{AP_C}$:

$$\beta_{cls} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \beta(\mathrm{IoU}^c_{post}, \hat{s}^c_{post}), \tag{2}$$

where $\mathcal{C}$ is the set of classes in the dataset and IoU^c_{post} is the set IoUs of BBs of true positives for class c ($\hat{B}^c_{post}$, Fig. 3).

Class-level correlation affects the performance of all detectors since it is directly related to $\mathrm{AP_C}$, the performance measure itself. To be more specific, $\mathrm{AP_C}$ for a single class is defined as the average of APs computed over 10 different IoU thresholds, $\mathrm{IoU} \in \{0.50, 0.55, ..., 0.95\}$, validating the true positives. For a specific threshold IoU, the detections are first sorted with respect to the classification scores, and then precision and recall pairs are calculated on each detection. Using these pairs, a precision-recall (PR) curve is obtained, and the area under the PR curve corresponds to the single AP value, $\mathrm{AP_{IoU}}$. When the correlation between

[2]While analyzing object detectors in terms of correlation, we employ Spearman correlation coefficient, $\beta(\cdot,\cdot)$, to measure the relation between the ranks of the values (i.e., scores and IoUs) instead of the values themselves, and aim to isolate the correlation quality from the localization and classification performances.



(a) Image-level Correlation for better NMS

(b) Class-level Correlation for better AP

Figure 4: How correlation affects detection performance. **(a)** Image-level correlation: Given detections before post-processing, NMS benefits from image-level correlation, thereby yielding detections with better IoU. Compare IoUs of detections in "positively correlated" (i.e., when the dark-colored ones have larger score) and "negatively correlated" (i.e., when the light-colored ones have larger score) outputs after NMS. **(b)** Class-level correlation: Given detections after post-processing, APs with larger IoUs and COCO-style AP benefit from positive class-level correlation (compare $\mathrm{AP}_{\mathrm{IoU}}$ columns in "positively correlated" and "negatively correlated" outputs after AP Calculation to see lower AP_{75} for the "negatively correlated" output in the red cell). $\mathrm{P}_{\mathrm{IoU}}$: Precision computed on a detection using the threshold IoU, True positives are color-coded in tables and input, white cells: false positives, and hence their IoU is not available, N/A.

classification and localization is maximized among true positives, larger precision values are obtained on the same detections in larger IoU values (e.g. P_{75} of orange detection is 1.00 and 0.20 with positive and negative correlation respectively in Fig. 4(b)).

3.2 Analyses of Object Detectors wrt. Correlation

Dataset and Implementation Details. Unless otherwise specified; we (i) employ the widely-used COCO dataset (Lin et al. 2014) by training the models on *trainval35K* (115K images), testing on *minival* (5k images), comparing with SOTA on *test-dev* (20k images), (ii) build upon the mmdetection framework (Chen et al. 2019), (iii) rely on AP-based measures and also use Optimal LRP (oLRP) (Oksuz et al. 2021b), β_{img} (Eq. 1) and β_{cls} (Eq. 2) to provide more insights, (iv) keep the standard configuration of the models, (v) use a ResNet-50 backbone with FPN (Lin et al. 2017), (vi) train models on 4 GPUs (A100 or V100 type GPUs) with 4 images on each GPU (16 batch size).

Analysis Setup. We conduct experiments to analyze the effects of the image-level (β_{img} – Table 1) and class-level (β_{cls} – Table 2) correlations. For both analyses, we compare three sets of methods, all of which are incorporated into the common ATSS baseline (Zhang et al. 2020) (see Sec. 2 for a discussion of these methods): (i) AP Loss and Focal Loss as methods not enforcing correlation, (ii) using an auxiliary head to enforce correlation, and (iii) Quality Focal Loss (QFL), aLRP Loss and Rank & Sort Loss as recent loss functions enforcing correlation. In our class-level analysis, we also employ NMS-free methods to demonstrate the effects of correlation on that approach.

We compare the methods based on (i) their AP-based performance, (ii) our proposed measures for correlation (Eqs. 1 and 2), and finally (iii) lower/upper bounds, $\mathrm{AP}_{\mathrm{C}}^{+1}/\mathrm{AP}_{\mathrm{C}}^{-1}$, obtained by modifying the ranking of the confidence scores pertaining to the GT classes of the positive detections to minimize/maximize Eq. 1 in Table 1 and Eq. 2 in Table 2. More particularly, in Table 1, given $\hat{s}^I_{pre}$ and $\hat{B}^I_{pre}$ (Fig. 3), we collect the GT class probabilities of positive detections and change their ranking in $\hat{s}^I_{pre}$ within an image following the ranking order of IoUs (computed using $\hat{B}^I_{pre}$), and in Table 2, we do the same operation class-wise for true positives given $\hat{s}^c_{post}$ and $\hat{B}^c_{post}$ (Fig. 3). To decouple other types of errors as much as possible; in Table 1, we *do not modify* the scores of the negative detections, the predicted BBs and the scores of non-GT classes of the positive detections, and in Table 2, we *do not modify* the scores of the false positives and the predicted BBs of the true positives. Note that achieving the upper bound in (iii) for image-level correlation also corresponds to perfectly minimizing RS Loss.

Observations. We observe in Tables 1 and 2 that:

(1) Our proposed measures in Eqs. 1 and 2 can measure the improvements in correlation consistently. In Tables 1 and 2, (i) aLRP Loss and RS Loss are proposed to improve AP Loss and (ii) aux. head and QFL are proposed to improve Focal Loss. In both tables, the proposed methods are shown to improve their baselines in terms of β_{img} and β_{cls}, suggesting that our measures can consistently evaluate image-level and class-level correlations respectively.

(2) NMS-free detectors can also potentially benefit from correlation. All detectors, including NMS-free ones, can exploit class-level correlation (compare AP_{C} and AP_{C}^{+1} to see ~ 10 points gap in Table 2). Still, existing methods do not enforce this correlation on NMS-free detectors.

(3) Existing methods enforcing correlation have still a large room for improvement. Considering that $\beta_{img} \in [27.2\%, 33.8\%]$ (Table 1) and $\beta_{cls} \in [37.5\%, 47.0\%]$ (Table 2), there is still room for improvement wrt. correlation.

	Performance				Modify ranking of scores					
Method	AP_C	AP_{50}	AP_{75}	β_{img}	AP_C^{-1}	AP_{50}^{-1}	AP_{75}^{-1}	AP_C^{+1}	AP_{50}^{+1}	AP_{75}^{+1}
Not Enforcing Correlation										
ATSS w. AP Loss (Chen et al. 2020)	38.1	58.2	41.0	27.2	24.9	53.2	19.2	57.0	72.4	62.2
ATSS w. Focal Loss (Lin et al. 2020)	38.7	57.6	41.5	27.3	25.6	51.8	21.1	55.8	70.6	60.5
Using Aux. Head										
ATSS w. ctr. head (Zhang et al. 2020)	39.3	57.5	42.6	28.7	16.8	32.4	15.3	49.8	64.8	54.2
Using Novel Loss										
ATSS w. aLRP Loss (Oksuz et al. 2020)	37.7	57.4	39.9	33.8	22.7	48.8	17.5	54.2	70.4	58.7
ATSS w. QFL (Li et al. 2020)	39.7	58.1	42.7	33.2	25.7	51.1	21.9	55.8	70.9	60.6
ATSS w. RS Loss (Oksuz et al. 2021a)	39.9	58.9	42.6	30.9	26.2	53.9	21.3	57.1	71.8	62.1

Table 1: Evaluation of NMS-based detectors in terms of image-level correlation. See Eq. 1 for β_{img}. AP_{IoU}^{+1} and AP_{IoU}^{-1} refer to the upper & lower bound APs (see analysis setup for details). The values are in %. Our β_{img} captures correlation consistently, e.g. that (i) Focal Loss is improved by ctr. head and QFL and (ii) AP Loss is improved by aLRP Loss and RS Loss wrt. β_{img}. Also, there is still room for improvement for object detectors wrt. β_{img} with a range between 27.2% and 33.8%.

	Performance				Modify ranking of scores					
Method	AP_C	AP_{50}	AP_{75}	β_{cls}	AP_C^{-1}	AP_{50}^{-1}	AP_{75}^{-1}	AP_C^{+1}	AP_{50}^{+1}	AP_{75}^{+1}
Not Enforcing Correlation										
- NMS-free Detectors										
Sparse R-CNN (Sun et al. 2021b)	37.7	55.8	40.5	37.5	30.1	55.8	28.9	48.6	55.8	52.7
DETR (Carion et al. 2020)	40.1	60.6	42.0	47.0	32.9	60.6	30.6	51.9	60.6	55.8
- NMS-based Detectors										
ATSS w. AP Loss (Chen et al. 2020)	38.1	58.2	41.0	39.4	30.0	58.2	26.6	48.5	58.2	54.0
ATSS w. Focal Loss (Lin et al. 2020)	38.7	57.6	41.5	40.3	30.2	57.6	27.3	48.7	57.6	53.6
Using Aux. Head										
ATSS w. ctr. head (Zhang et al. 2020)	39.3	57.4	42.5	42.5	30.2	57.4	27.6	48.7	57.4	53.5
Using Novel Loss										
ATSS w. aLRP Loss (Oksuz et al. 2020)	37.7	57.4	39.9	42.0	29.1	57.4	25.0	47.8	57.4	52.7
ATSS w. QFL (Li et al. 2020)	39.7	58.1	42.7	45.7	30.6	58.1	27.7	49.1	58.1	53.9
ATSS w. RS Loss (Oksuz et al. 2021a)	39.9	58.9	42.6	43.2	31.1	58.9	28.1	49.8	58.9	54.8

Table 2: Evaluation of detectors wrt. class-level correlation. See Eq. 2 for β_{cls}. AP_{IoU}^{+1} & AP_{IoU}^{-1} denote upper & lower bound APs (analysis setup for details). Values are in %. NMS-free detectors can also benefit from class-level correlation (compare AP_C^{+1} with AP_C for Sparse R-CNN), and as in β_{img} (c.f. Table 1 and its caption), β_{cls} measures the correlation consistently. $AP_{50}^{+1} = AP_{50}^{-1}$ = AP_{50} since only modifying TPs validated from IoU=0.50 does not effect AP_{50} (see Fig. 4(b) for an example).

(4) While significantly important, improving correlation may not always imply performance improvement. For example, aLRP Loss in Table 1 has the largest correlation but the lowest AP_C. Such a situation can arise, for example, when a method does not have good localization performance. In the extreme case, assume a detector yields perfect β_{img}, image-level ranking correlation, but the IoUs of all positive examples are less than 0.50 implying no TP at all. Hence, boosting the correlation, while simultaneously preserving a good performance in each branch, is critical.

4 Correlation Loss: A Novel Loss Function for Object Detection

Correlation (Corr.) Loss is a simple plug-in loss function to improve correlation of classification and localization tasks. Correlation Loss is unique in that it can be easily incorporated into any object detector, whether NMS-based or NMS-free (see Observation (2) - Sec. 3.2), and improves performance without affecting the model size, inference time and with negligible effect on training time (Sec. 5.4). Furthermore, from a fundamental perspective, Corr. Loss can supervise both classification and localisation heads for a better correlation while existing methods generally focus on a single head such as classification (Fig. 1).

Definition. Given an object detector with loss function $\mathcal{L}_{OD}$, our Correlation Loss ($\mathcal{L}_{corr}$) is simply added using a weighting hyper-parameter λ_{corr}:

$$\mathcal{L}_{OD} + \lambda_{corr}\mathcal{L}_{corr}. \tag{3}$$

$\mathcal{L}_{corr}$ is the Correlation Loss defined as:

$$\mathcal{L}_{corr} = 1 - \rho(\hat{\text{IoU}}, \hat{\text{s}}), \tag{4}$$

where $\rho(\cdot,\cdot)$ is a correlation coefficient; $\hat{\text{s}}$ and $\hat{\text{IoU}}$ are the confidence scores of the GT class and IoUs of the predicted BBs pertaining to the positive examples in the batch.

Practical Usage. To avoid promoting trivial cases with high correlation but low performance (Observation (4) - Sec. 3.2), similar to QFL (Li et al. 2020) and RS Loss (Oksuz et al. 2021a), we only use the gradients of $\mathcal{L}_{corr}$ wrt. classification score, i.e., we backpropagate the gradients through only the classifier. We mainly adopt two dif-

Method	$AP_C \uparrow$	$AP_{50} \uparrow$	$AP_{75} \uparrow$	oLRP $\downarrow$
– NMS-based Methods –				
Retina Net	36.5	55.4	39.1	70.7
w. Conc.Corr	37.0	**55.7**	39.7	70.2
w. Spear.Corr	**37.5**	55.4	**40.5**	**69.7**
Fovea Box	36.4	**56.5**	38.6	70.2
w. Conc.Corr	**37.1**	56.4	**39.6**	**69.7**
w. Spear.Corr	37.0	55.6	39.3	70.0
ATSS	38.7	57.6	41.5	69.0
w. Conc.Corr	**39.8**	**57.9**	**43.2**	**68.2**
w. Spear.Corr	39.3	56.6	42.5	68.7
PAA	39.9	57.3	43.4	68.6
w. Conc.Corr	**40.7**	**58.8**	**44.3**	**67.7**
w. Spear.Corr	40.4	58.0	43.7	67.8
– NMS-free Methods –				
Sparse R-CNN	37.7	55.8	40.5	69.5
w. Conc.Corr	38.9	**57.2**	41.8	**68.1**
w. Spear.Corr	**39.3**	56.7	**42.5**	68.3

Table 3: Comparison on detectors not considering correlation. Accordingly, we remove aux. heads from ATSS (Zhang et al. 2020) and PAA (Kim and Lee 2020) for fair comparison (see Table 6 for comparison with aux. heads and novel loss functions). We use ResNet-50 and train the models for 12 epochs. Simply incorporating our Corr. Loss provides (i) $\sim 1AP_C$ improvement for NMS-based detectors consistently and (ii) $\sim 1.5AP_C$ on the NMS-free Sparse R-CNN.

ferent correlation coefficients for $\rho(\cdot,\cdot)$ and obtain two versions of Correlation Loss: (i) *Concordance Loss*, defined as the Correlation Loss when Concordance correlation coefficient is optimized ($\rho(\cdot,\cdot) = \gamma(\cdot,\cdot)$), which aims to match the confidence scores with IoUs. (ii) *Spearman Loss* as Correlation Loss when Spearman correlation coefficient is optimized ($\rho(\cdot,\cdot) = \beta(\cdot,\cdot)$), thereby enforcing the ranking of the classification scores considering IoUs. To tackle the non-differentiability of ranking operation while computing Spearman Loss, we leverage the differentiable sorting operation from Blondel et al. (Blondel et al. 2020). When applying our Correlation Loss to NMS-free methods, which use an iterative multi-stage loss function, we incorporate $\mathcal{L}_{corr}$ to every stage.

5 Experimental Evaluation

We evaluate Corr. Loss on (i) the COCO dataset with five different object detectors of various types (Sparse R-CNN as NMS-free, FoveaBox as anchor-free, RetinaNet as anchor-based, ATSS and PAA using auxiliary head), and one instance segmentation method, YOLACT; and (ii) an additional dataset (Cityscapes) for the method with the largest gain, i.e., Sparse R-CNN.

5.1 Comparison with Methods Not Considering Correlation

We train these five object detectors and the instance segmentation method (Tables 3 and 5) with and without our Corr. Loss (as Concordance Loss or Spearman Loss).

NMS-based Detectors. Table 3 suggests $\sim 1.0AP_C$ gain on NMS-based detectors: (i) Spearman Loss ($\lambda_{corr} = 0.1$) improves RetinaNet by $1.0AP_C$ and oLRP, (ii) Concordance Loss ($\lambda_{corr} = 0.2$) enhances anchor-free FoveaBox by $0.7AP_C$, and (iii) Concordance Loss ($\lambda_{corr} = 0.3$) improves ATSS and PAA by $\sim 1AP_C$ and ~ 1oLRP.

NMS-free Detectors. Our results in Table 3 suggest that Sparse R-CNN, an NMS-free method, can also benefit from our Corr. Loss: (i) Both Concordance ($\lambda_{corr} = 0.3$) and Spearman Losses ($\lambda_{corr} = 0.2$) improve baseline; (ii) Spearman Loss improves AP_C significantly by up to 1.6; (iii) as hypothesized, the gains are owing to APs with larger IoUs, e.g., AP_{75} improves by up to 2.0, and (iv) gains persist in a stronger setting of Sparse R-CNN (Supp. Mat.).

Method	AP	AP_{50}	AP_{75}
Sparse R-CNN	39.0	63.1	37.6
w. Spear.Corr (Ours)	**40.8**	**64.4**	**40.8**

Table 4: Results on Cityscapes dataset.

Method	AP_C^{mask}	AP_{50}^{mask}	AP_{75}^{mask}
YOLACT	28.3	47.8	28.8
w. Conc.Corr (Ours)	28.8	**48.3**	29.6
w. Spear.Corr (Ours)	**29.0**	48.3	**30.0**

Table 5: Comparison with YOLACT.

Cityscapes dataset. To see the effect of Corr. Loss over different scenarios, we train Sparse R-CNN with Spearman Loss (the model that has the best gain over baseline in Table 3), on the Cityscapes dataset (Cordts et al. 2016) ($\lambda_{corr} = 0.6$), a dataset for autonomous driving object detection. Table 4 presents that (i) Spearman Loss also improves baseline Sparse R-CNN on Cityscapes by 1.8 AP and (ii) our gain mainly originates from APs with larger IoUs, i.e. AP_{75} improves by more than 3 points, from 37.6 to 40.8.

Instance Segmentation. We train YOLACT (Bolya et al. 2019) as an instance segmentation method with Corr. Loss and observed 0.7 mask AP gain using Spearman Loss ($\lambda_{corr} = 0.5$ - Table 5), implying 1.7% relative gain.

Aux.	QFL	RS Loss	Ours	AP_C	β_{img}	β_{cls}
				38.7	27.3	40.3
✓				39.3	28.7	42.5
	✓			39.7	33.2	45.7
		✓		39.9	30.9	43.2
			✓	39.8	31.6	45.2
✓			✓	40.0	31.1	44.8
	✓		✓	39.9	**34.6**	45.6
		✓	✓	**40.2**	33.6	**46.1**

Table 6: Comparison with methods enforcing correlation. Corr. Loss (i) reaches similar results with existing methods on ATSS, (ii) is complementary to those methods thanks to its simple design and (iii) once combined with RS Loss, outperforms compared methods. All measures are higher-better.

	Method	Backbone	Ep.	AP_C	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Venue
NMS-based	ATSS (Zhang et al. 2020)	R-101-DCN	24	46.3	64.7	50.4	27.7	49.8	58.4	CVPR 2020
	GFLv2 (Li et al. 2019)	R-101-DCN	24	48.3	66.5	52.8	28.8	51.9	60.7	CVPR 2021
	aLRP Loss (Oksuz et al. 2020)	X-101-DCN	100	48.9	69.3	52.5	30.8	51.5	62.1	NeurIPS 2020
	VFNet (Zhang et al. 2021)	R-101-DCN	24	49.2	67.5	53.7	29.7	52.6	62.4	CVPR 2021
	DW (Li et al. 2022)	R-101-DCN	24	49.3	67.6	53.3	29.2	52.2	63.5	CVPR 2022
	TOOD (Feng et al. 2021)	R-101-DCN	24	49.6	67.4	54.1	30.5	52.7	62.4	ICCV 2021
	RS-Mask R-CNN+ (Oksuz et al. 2021a)	X-101-DCN	36	50.2	**70.3**	54.8	**31.5**	53.5	63.9	ICCV 2021
NMS-free	TSP R-CNN (Sun et al. 2021c)	R-101-DCN	96	47.4	66.7	51.9	29.0	49.7	59.1	ICCV 2021
	Sparse R-CNN (Sun et al. 2021b)	R-101-DCN	36	48.9	68.3	53.4	29.9	50.9	62.4	CVPR 2021
	Dynamic DETR (Dai et al. 2021)	X-101-DCN	36	49.3	68.4	53.6	30.3	51.6	62.5	ICCV 2021
	Deformable DETR (Zhu et al. 2021)	X-101-DCN	50	50.1	69.7	54.6	30.6	52.8	64.7	ICLR 2021
Ours	Corr-Sparse R-CNN	R-101-DCN	36	49.6	67.8	54.1	29.2	52.3	64.9	
	Corr-Sparse R-CNN	X-101-DCN	36	**51.0**	69.2	**55.7**	31.1	**53.7**	**66.3**	

Table 7: SOTA comparison on COCO *test-dev*. Our Corr-Sparse R-CNN (i) performs on-par or better compared to NMS-based methods also enforcing correlation; (ii) outperforms NMS-free methods. R-101: ResNet-101, X-101: ResNext-101, Ep.: Epoch.

5.2 Comparison with Methods Enforcing Correlation

Table 6 compares Corr. Loss. with using an aux. head, QFL and RS Loss on the common ATSS baseline:

Detection Performance. Reaching $39.8AP_C$ without an aux. head, Concordance Loss outperforms using an aux. head, which introduces additional learnable parameters (39.8 vs $39.3AP_C$). Concordance Loss also performs on-par with the recently proposed, relatively complicated loss functions, QFL (Li et al. 2020) and RS Loss (Oksuz et al. 2021a). Besides, owing to its simple usage, Concordance Loss is complementary to existing methods: It yields $40.0AP_C$ with an aux. head (+0.7 AP_C) and $40.2AP_C$ with RS Loss (+0.3 AP_C) without introducing additional learnable parameters.

Correlation Analysis. To provide insight, we report β_{img} (Eq. 1) and β_{cls} (Eq. 2) in Table 6: Our Concordance Loss (i) improves baseline correlation significantly, enhancing β_{img} (from 27.3% to 31.6%) and β_{cls} (from 40.3% to 45.2%) both by $\sim 5\%$, and (ii) results in better correlation than all methods wrt. β_{img} and β_{cls} once combined with QFL and RS Loss respectively. This set of results confirms that Concordance Loss improves correlation between classification and localization tasks in both image-level and class-level.

5.3 Comparison with SOTA

Here, we prefer Sparse R-CNN owing to its competitive detection performance and our large gains. We train our "Corr-Sparse R-CNN" for 36 epochs with DCNv2 (Zhu et al. 2019) and multiscale training by randomly resizing the shorter side within [480, 960] similar to common practice (Oksuz et al. 2021a; Zhang et al. 2021; Sun et al. 2021b). Table 7 presents the results on COCO test-dev (Lin et al. 2014):

NMS-based Methods. On the common ResNet-101-DCN backbone and with similar data augmentation, our Corr-Sparse R-CNN yields $49.6AP_C$ at 13.7 fps (on a V100 GPU) outperforming recent NMS-based methods, all of which also enforce correlation, e.g., (i) RS-R-CNN (Oksuz et al. 2021a) by $1.8AP_C$, (ii) GFLv2 (Li et al. 2019) by more than $1AP_C$, and (iii) VFNet (Zhang et al. 2021) in terms of not only AP_C but also efficiency (with 12.6 fps on a V100 GPU). On ResNeXt-101-DCN, our Corr-Sparse R-CNN provides $51.0AP_C$ at 6.8 fps, surpassing all methods including RS-Mask R-CNN+ ($50.2AP_C$ at 6.4 fps), additionally using masks and Carafe FPN (Wang et al. 2019).

NMS-free Methods. Our Corr-Sparse R-CNN outperforms (i) TSP R-CNN (Sun et al. 2021c) by more than $2AP_C$ on ResNet-101-DCN with significantly less training, (ii) Sparse R-CNN (Sun et al. 2021b) by $\sim 2AP_C$ and Deformable DETR (Zhu et al. 2021), a recent strong NMS-free method, by $\sim 1AP_C$ on ResNeXt-101-DCN.

5.4 Ablation & Hyper-parameter Analyses

Optimizing Different Correlation Coefficients. Spearman Loss yields better localization performance, i.e. the lowest localization error wrt. $oLRP_{Loc}$ in all experiments while it rarely yields the best $oLRP_{FP}$ or $oLRP_{FN}$, implying its contribution to classification to be weaker than Concordance Loss (see Supp.Mat. for components of oLRP). We also tried Pearson Correlation Coefficient on ATSS and Sparse R-CNN but it performed worse compared to either using Spearman or Concordance (Supp.Mat.).

Backpropagating Through Different Heads. On Sparse R-CNN, we observed that the performance degrades when we backpropagate either only localization head (37.5 AP) or both heads (38.9 AP). Hence, we preferred backpropagating the gradients only through the classification head (39.3 AP).

Effect on Training Time. Using Spearman or Concordance Loss to train Sparse R-CNN, computing the loss for 6 times each iteration, increases iteration time 0.50 sec to 0.51 sec on V100 GPUs, suggesting a negligible overhead.

Sensitivity to λ_{corr}. We found it sufficient to search over $\{0.1, 0.2, 0.3.0.4, 0.5, 0.6\}$ to tune λ_{corr}. Supp. Mat. presents empirical results for grid search.

6 Conclusion

In this paper, we defined tools to evaluate object detectors wrt. correlation, provided analyses on several methods and proposed Correlation Loss as a plug-in loss function for object detectors. Our extensive experiments on six detectors show that Correlation Loss. consistently improves the detection and correlation performance, and reaches SOTA results.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) (under grant 120E494). We also gratefully acknowledge the computational resources kindly provided by TÜBİTAK ULAKBIM High Performance and Grid Computing Center (TRUBA) and METU Robotics and Artificial Intelligence Center (ROMER). Dr. Akbas is supported by the "Young Scientist Awards Program (BAGEP)" of Science Academy, Turkey.

References

Blondel, M.; Teboul, O.; Berthet, Q.; and Djolonga, J. 2020. Fast differentiable sorting and ranking. In *International Conference on Machine Learning (ICML)*.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. YOLACT: Real-time Instance Segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision (ECCV)*.

Chen, K.; Lin, W.; li, J.; See, J.; Wang, J.; and Zou, J. 2020. AP-Loss for Accurate One-Stage Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1–1.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv*, 1906.07155.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, X.; Chen, Y.; Yang, J.; Zhang, P.; Yuan, L.; and Zhang, L. 2021. Dynamic DETR: End-to-End Object Detection With Dynamic Attention. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Feng, C.; Zhong, Y.; Gao, Y.; Scott, M. R.; and Huang, W. 2021. TOOD: Task-aligned One-stage Object Detection. In *The International Conference on Computer Vision (ICCV)*.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

He, Y.; Zhu, C.; Wang, J.; Savvides, M.; and Zhang, X. 2019. Bounding Box Regression With Uncertainty for Accurate Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; and Wang, X. 2019. Mask Scoring R-CNN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of Localization Confidence for Accurate Object Detection. In *The European Conference on Computer Vision (ECCV)*.

Kim, K.; and Lee, H. S. 2020. Probabilistic Anchor Assignment with IoU Prediction for Object Detection. In *The European Conference on Computer Vision (ECCV)*.

Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; and Shi, J. 2020. FoveaBox: Beyound Anchor-Based Object Detection. *IEEE Transactions on Image Processing*, 29: 7389–7398.

Law, H.; and Deng, J. 2018. CornerNet: Detecting Objects as Paired Keypoints. In *The European Conference on Computer Vision (ECCV)*.

Li, S.; He, C.; Li, R.; and Zhang, L. 2022. A Dual Weighting Label Assignment Scheme for Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2019. Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2020. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lin, T.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature Pyramid Networks for Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(2): 318–327.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *The European Conference on Computer Vision (ECCV)*.

Liu, J.; Li, D.; Zheng, R.; Tian, L.; and Shan, Y. 2021. RankDetNet: Delving Into Ranking Constraints for Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 264–273.

Oksuz, K.; Cam, B. C.; Akbas, E.; and Kalkan, S. 2020. A Ranking-based, Balanced Loss Function Unifying Classification and Localisation in Object Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Oksuz, K.; Cam, B. C.; Akbas, E.; and Kalkan, S. 2021a. Rank & Sort Loss for Object Detection and Instance Segmentation. In *The International Conference on Computer Vision (ICCV)*.

Oksuz, K.; Cam, B. C.; Kalkan, S.; and Akbas, E. 2021b. One Metric to Measure them All: Localisation Recall Precision (LRP) for Evaluating Visual Detection Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6): 1137–1149.

Roh, B.; Shin, J.; Shin, W.; and Kim, S. 2022. Sparse DETR: Efficient End-to-End Object Detection with Learnable Sparsity. In *The International Conference on Learning Representations (ICLR)*.

Sun, P.; Jiang, Y.; Xie, E.; Shao, W.; Yuan, Z.; Wang, C.; and Luo, P. 2021a. What Makes for End-to-End Object Detection? In *International Conference on Machine Learning (ICML)*.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; and Luo, P. 2021b. SparseR-CNN: End-to-End Object Detection with Learnable Proposals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Z.; Cao, S.; Yang, Y.; and Kitani, K. M. 2021c. Rethinking Transformer-Based Set Prediction for Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C. C.; and Lin, D. 2019. CARAFE: Content-Aware ReAssembly of FEatures. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, H.; Wang, Y.; Dayoub, F.; and Sünderhauf, N. 2021. VarifocalNet: An IoU-aware Dense Object Detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, X.; Wan, F.; Liu, C.; Ji, R.; and Ye, Q. 2019. FreeAnchor: Learning to Match Anchors for Visual Object Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable ConvNets V2: More Deformable, Better Results. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable {DETR}: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations (ICLR)*.