This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

The Box Size Confidence Bias Harms Your Object Detector

Johannes Gilg Torben Teepe Fabian Herzog Gerhard Rigoll
Technical University of Munich

Abstract

Countless applications depend on accurate predictions with reliable confidence estimates from modern object detectors. However, it is well known that neural networks, including object detectors, produce miscalibrated confidence estimates. Recent work even suggests that detectors' confidence predictions are biased with respect to object size and position. In object detection, the issues of conditional biases, confidence calibration, and task performance are usually explored in isolation, but, as we aim to show, they are closely related. We formally prove that the conditional confidence bias harms the performance of object detectors and empirically validate these findings. Specifically, to quantify the performance impact of the confidence bias on object detectors, we modify the histogram binning calibration to avoid performance impairment and instead improve it through calibration conditioned on the bounding box size. We further find that the confidence bias is also present in detections generated on the training data of the detector, which can be leveraged to perform the de-biasing. Moreover, we show that Test Time Augmentation (TTA) confounds this bias, which results in even more significant performance impairments on the detectors. Finally, we use our proposed algorithm to analyze a diverse set of object detection architectures and show that the conditional confidence bias harms their performance by up to 0.6 mAP and 0.8 mAP$_{50}$. Code available at https://github.com/Blueblue4/Object-Detection-Confidence-Bias.

1. Introduction

Accurate probability estimates are essential for automated decision processes. They are crucial for accurate and reliable performance and for properly assessing risks. This is especially true for object detectors, which are regularly deployed in uniquely critical domains such as automated driving, medical imaging, and security applications, where human lives can be at stake.

Despite these high stakes, confidence calibration for object detectors receives comparatively little attention. Most of the attention in the design of object detectors goes toward chasing state-of-the-art results on performance benchmarks while ignoring problems in the confidence of their predictions.

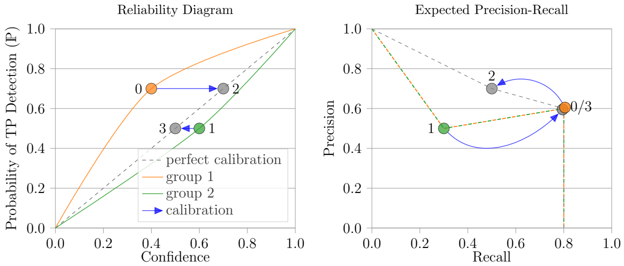

Figure 1. **Example illustration of conditional bias hurting object detection performance.** The made up data shows detections with two identifiable sub-groups. Conditioning the confidence calibration on differently miscalibrated sub-groups, shown in the reliability diagram (left), increases a detector's performance, seen in the precision-recall curve (right). The operating point '2' has strictly higher precision and recall than point '1' but it can only be reached if the conditional bias of the detector is removed. Best viewed in color and zoomed in.

Additionally, object detectors have recently been shown to produce conditionally biased confidences with respect to their regression outputs [22], i.e., the box size and position. This bias means that for detected objects with the same predicted confidence the probability for it being a true positive might vary considerably depending on the object size and position in an image. However, it is still unclear how this bias relates to the performance of the affected object detectors.

We formally show that the conditional confidence bias can hurt object detection performance and empirically validate this finding. A conditional confidence bias, *e.g.*, the bounding box bias, can prevent the object detector from reaching strictly better operation points in the precision-recall domain. In Fig. 1 a simplified illustration with mock data shows the impact of conditionally biased detections. We also quantify this performance impact by calibrating object detectors using a modified histogram binning conditioned on the bounding box size.

Our Contributions are:

1. We formally prove that a conditional bias in object detectors leads to a non-optimal expected Average Precision (AP).
2. We empirically verify this finding and quantify the performance impact using a modified histogram binning, conditioned on the bounding box size.
3. We demonstrate that Test Time Augmentation (TTA) can confound the problems caused by the conditional bias by conditionally calibrating the detections for each augmentation.
4. Using our proposed conditional calibration procedure with a heuristic performance metric on the training data, we are able to improve the performance of most of the tested object detectors on the standard COCO [24] evaluation and test-dev benchmark, and verify that the confidence bias has an actual performance impact.

2. Related Works

Confidence Calibration of Neural Networks. Confidence calibration is usually applied as a post-processing step for uncertainty estimation. Modern neural networks make highly miscalibrated predictions as shown by Guo *et al.* [13] and hinted at in earlier works [29, 41]. There are many ways to calibrate the confidence of predictive models, such as histogram binning [49], Bayesian Binning [26], isotonic regression [50] and Platt scaling [33], with the multi-class modification temperature scaling [13] and the more general Beta calibration [19]. Confidence calibration of deep learning object detectors was first addressed by Neumann *et al.* [27] as a learning problem. Küppers *et al.* generalized different calibration methods to conditional calibration of object detectors [22].

Measuring Calibration Errors. Along with the methods for calibration, measuring to what degree predictions are calibrated is also a long-standing field of study [4, 12, 47, 8]. Inspired by the earlier visualization through reliability diagrams [8], the nowadays widely used Expected Calibration Error (ECE) [26] still has many shortcomings exposed and modifications proposed [21, 30, 44] including an adaption to object detectors [22]. We note that we explicitly do not show ECE, as it does not capture conditional confidence biases.

Bias in Deep Learning. Bias in deep learning is widely studied, usually in the context of fairness [51, 34, 3, 15, 46, 39], dataset biases [43, 52, 1, 18] and learning techniques to mitigate bias during training [17, 2, 1, 52, 39]. On the other hand, bias in object detectors is less explored, with the exception of the context bias of object detectors [55, 38]. Zhao *et al.* [52] have explored label biases in an object detection dataset. Küppers *et al.* [22] are the first to show conditional bias in the confidence estimates of object detectors with respect to its regressed bounding box outputs. In contrast, we show how the conditional confidence bias with respect to the bounding box is actually detrimental to the performance of object detectors.

3. Preliminaries and Technical Background

Object Detection. An object detector is a predictor that generates a set of detections $\mathcal{D}$ representing the presence and location of objects in an image. Each of the detector's $N+1 = |\mathcal{D}|$ detections $d_i = (k_i, b_i, c_i)$, consist of a category k_i, a rectangular bounding box b_i =(w, h, x, y) and a confidence c_i. The confidence c_i represents the *certainty* of the detector for the presence of an object with the category k_i at the location b_i.

Evaluating Object Detectors. Object detectors are evaluated against a ground truth set of objects ($\mathcal{G}$). The evaluation is performed separately for the detections of every object category. A detection d_i is categorized as a True Positive (TP) if the overlap of its predicted bounding box with a ground truth bounding box is larger than a threshold t_{IoU} and if c_i is highest among all detections that have a large enough overlap with the ground truth bounding box. The overlap is calculated using the Jaccard coefficient, in this context more fittingly termed Intersection over Union (IoU). We define an indicator variable τ_i, which is 1 if d_i is a TP detection and 0 otherwise. In the context of object detection, the notion of a True Negative is not well defined as it would correspond to an arbitrary number of "non-objects" in an image. Therefore, object detectors are evaluated using precision and recall metrics [31].

To compute the precision and recall of an object detector, its detections are sorted according to their confidence from largest to smallest ($c_i \geq c_{i+1}, \forall i \in [1, N-1]$). Then, the precision after i detections Prec(i) is the fraction TP predictions TP$_i$ out of the i evaluated predictions. By omitting the dependence on $\mathcal{D}$, $\mathcal{G}$, and t_{IoU} for brevity, we can simply write it as

$$\text{Prec}(i) = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} = \frac{\sum_{k=1}^{i} \tau_k}{i} \tag{1}$$

and, analogously, the recall after i detections is the fraction of TP predictions out of the number of available ground truth objects ($|\mathcal{G}|$):

$$\text{Rec}(i) = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} = \frac{\sum_{k=1}^{i} \tau_k}{|\mathcal{G}|}. \tag{2}$$

They can be unified into a single metric - the so-called average precision (AP$_{t_{\text{IoU}}}$)

$$\text{AP}_{t_{\text{IoU}}} = \sum_{i=1}^{N} \text{Prec}(i) \cdot \Delta\text{Rec}(i), \tag{3}$$

where $\Delta\mathrm{Rec}(i)$ denotes the change of recall from d_{i-1} to d_i. The $\mathrm{AP}_{t_{\text{IoU}}}$ is then averaged over a range of $t_{\text{IoU}} \in [0.50,\, 0.55,\, ...,\, 0.95]$, and over all object categories to get a final mean Average Precision (mAP) value, which is a unified performance indicator for a detector. The also used mAP_{50} is the class-averaged AP for $t_{\text{IoU}} = 0.50$.

Official benchmark implementations of the mAP metric apply maximum interpolations of the precision-recall curve and point sampling at specific recall values[24, 9, 31]. This can produce a slightly more optimistic estimates of the AP and mAP metrics than Eq. (3). We also use the official Common Objects in Context dataset [24] (COCO) evaluation script for better comparability on the benchmarks.

Confidence Calibration. The goal of confidence calibration is for the c_i for each prediction to be equivalent to the empiric object detector's probability for a TP prediction $\mathbb{P}(\tau_i{=}1|\, d{=}d_i)$. From here on we denote it as $\mathbb{P}_i$ in short. For the confidence calibration, we consider the object detector as a stochastic process. The label of a prediction d_i is now represented by the random variable $T_i \sim \mathrm{Bernoulli}(\mathbb{P}_i)$, from which τ_i with $t_{\text{IoU}} = 0.50$ is drawn as a sample. $\mathbb{P}_i$ can also be seen as the precision of the object detector for a set of detections with the same confidence c_i; we refer to $\mathbb{P}_i$ as probability of a "successful" or TP detection $\mathbb{P}(\tau_i{=}1|\, d{=}d_i)$ to avoid confusion with the metric defined in Eq. (1). This notation also makes the definition compatible with the confidence calibration of classification neural networks [13], as $\mathbb{P}(\tau_i{=}1)$ is equivalent to the empiric accuracy of a classifier. Most deep learning-based object detectors are not well calibrated with regard to their confidence estimates [27, 22]. Therefore, the goal of confidence calibration is to find a mapping $\hat{f}$ that estimates the true confidence calibration curve f over the input interval $(0, 1]$:

$$f(c_i) = \mathbb{P}(\tau_i = 1 | c = c_i). \tag{4}$$

Küppers *et al.* discovered that $\mathbb{P}_i$ also depends on the predicted bounding box size and position, not only on c_i:

$$f(d_i) = \mathbb{P}(\tau_i = 1 | c = c_i, b = b_i). \tag{5}$$

For simplicity, we only focus on the size of the predicted bounding boxes $(h \cdot w)$ for the conditional confidence calibration, ignoring the position (x, y). The challenge in confidence calibration is that we can only draw from each T_i once. The conditional probability $\mathbb{P}$ needs to be estimated from the binary outcomes τ over all possible confidence values $c \in (0, 1]$; hence, this is a density estimation problem.

Histogram Binning. One of the most straightforward black-box calibration methods is histogram binning [49]. For histogram binning the predictions are grouped into M confidence intervals C_m of equal size, so the interval of the m-th bin is $C_m = \left(\frac{m-1}{M}, \frac{m}{M}\right]$. The density estimation is performed over the individual intervals separately: The estimated probability of a TP detection $\hat{\mathbb{P}}_m$ in interval m is calculated by taking the detections with confidences that lie in the confidence interval and calculating the fraction of detections that are TPs. The histogram binning calibration of some detection d_i with confidence c_i is a simple lookup of the calculated average $\hat{\mathbb{P}}_{\tilde{m}}$ of the corresponding bin $C_{\tilde{m}} | c_i \in C_{\tilde{m}}$. Histogram binning can be extended to a multivariate calibration scheme [22]. For the conditional-dependent binning we first split the detections according to their box size into bins B and then perform the previously described histogram binning for each of the disjoint detection sub-groups. This more general calibration function $\hat{f}_{C,B}(d)$ produces an estimate for the conditional probability $\mathbb{P}$, as described in Eq. (5).

4. Bias in Confidence of Object Detectors

We have the hypothesis that the conditional confidence bias [22] is hurting object detectors' performance. In Fig. 1 we visualize this idea based on an exaggerated example of two groups of detections with different calibration curves. Each of the groups only has detections with a single respective confidence value and with this example it is obvious that a detector with a confidence threshold of 0.55 would have a precision of 50% for the uncalibrated detections (0,1) and a precision of 70% if the detector was perfectly calibrated (2,3). A related improvement can be observed in the precision recall curve. The area under this curve is closely related to the AP metric [31]. Our simple example and our hypothesis indicate that bias in the confidence estimates of object detectors with respect to bounding box size and position [22] is hurting the performance of the detectors. We are interested in a formal proof of this hypothesis.

4.1. Maximizing Average Precision

To prove our assumption that the confidence bias is hurting the performance of object detectors, we take a look at how the $\mathrm{AP}_{t_{\text{IoU}}}$ for any t_{IoU} relates to $\mathbb{P}$ and how it can be maximized for a set of detections $\mathcal{D}$. An object detector can be seen as a stochastic process (see Sec. 3) so we need to analyze the expected AP. From Eq. (3) we get

$$\mathbb{E}_T\left[\mathrm{AP}_{t_{\text{IoU}}}\right] = \mathbb{E}_T\left[\sum_{i=1}^{N} \mathrm{Prec}(i) \cdot \Delta\mathrm{Rec}(i)\right]. \tag{6}$$

Substituting Eqs. (1) and (2) and our stochastic indicator variable T, we get:

$$\mathbb{E}_T\left[\mathrm{AP}_{t_{\text{IoU}}}\right] = \mathbb{E}_T\left[\sum_{i=1}^{N}\left(\frac{\sum_{k=1}^{i-1}(T_k) + T_i}{i} \cdot \frac{T_i}{|\mathcal{G}|}\right)\right]. \tag{7}$$

If we assume independence of $\mathbb{P}_i$ and $\mathbb{P}_j$ for every i, j with $i \neq j$

$$\mathbb{E}_T[\mathrm{AP}_{t_{\mathrm{IoU}}}] = \frac{1}{|\mathcal{G}|} \sum_{i=1}^{N} \left(\frac{\sum_{k=1}^{i-1}(\mathbb{P}_k) + 1}{i} \cdot \mathbb{P}_i \right). \quad (8)$$

With some simple arithmetic we can reformulate this as:

$$\mathbb{E}_T[\mathrm{AP}_{t_{\mathrm{IoU}}}] = \frac{1}{|\mathcal{G}|} \sum_{i=1}^{N} \underbrace{\left(\frac{\mathbb{P}_i}{i} + \mathbb{P}_i \sum_{k=i+1}^{N} \frac{\mathbb{P}_k}{k} \right)}_{h_i(\mathbb{P}_i, \mathbb{P})}. \quad (9)$$

Here, we see that $h_i(l, \mathbb{P}) > h_{i+1}(l, \mathbb{P})$ for $i \in \mathbb{N}$ and $l \in (0, 1]$. We can therefore maximize the sum in the expected $\mathrm{AP}_{t_{\mathrm{IoU}}}$ calculation by sorting the predictions according to their $\mathbb{P}$ from larges to smallest. Since the detections are sorted according to their confidence before evaluating the $\mathrm{AP}_{t_{\mathrm{IoU}}}$ (see Sec. 3), it is maximized under the following condition:

$$\mathbb{P}_n < \mathbb{P}_m \; \forall \, n, m \, | \, c_n < c_m. \quad (10)$$

Under the assumption that this condition then holds across different t_{IoU}'s it also maximises the mAP. We verify that this condition largely holds in practice by showing the choice of t_{IoU} used for Eq. (10) has little effect on the mAP (see supplementary material). It follows that there are only two circumstances under which the calibration can be beneficial for the expected performance of an object detector. The obvious case is when the confidence calibration curve for one class is not monotonic. This is usually not the case. During training, the monotonic loss function guides the predictions with an empirically higher probability $\mathbb{P}_i$ of being a TP to have a higher confidence c_i through gradient descent, all else being equal. The second case is when there is a conditional confidence bias. When identifiable sub-groups within the predictions of an object detector $d_n \in \mathcal{D}^A$ and $d_m \in \mathcal{D}^B$ have different conditional success probability $\mathbb{P}_n \neq \mathbb{P}_m$ for the same confidence $c_n = c_m$, this clearly violates Eq. (10). For optimal performance the two subgroups would have to have their confidences c transformed so that their combined predictions would have to be equally precise for equally confident predictions, *i.e.*, have the same monotonic calibration curve. Then their combined detections can satisfy Eq. (10) which is the required condition to maximize the expected AP. The interested reader is the referred to the supplementary material for a more detailed version of the proof.

4.2. Conditional Confidence Calibration

The non-optimal ordering of the object detector's predictions caused by the conditional bounding box bias reduces the detector's expected performance. Now we want to know how much the performance is actually deteriorated. To estimate the performance impact, we try to correct the variation between calibration curves and see how much it increases the performance metrics. The variation is eliminated if we find a mapping for detection confidences that eliminates the conditional bias, resulting in equal calibration curves f. This can be reached by mapping the confidences to be equal to their probability of success for each bounding box size. Of course the probability is generally not known, but confidence calibration deals with exactly the problem of finding a function to map confidence scores to their empirical success probability (see Sec. 3).

According to our reasoning, conditional confidence calibration should reduce the box size confidence bias of object detectors. Reducing this bias should revert the performance impact and increase the AP of the detector. We try to validate this using the publicly available object detector CenterNet [54], with the Hourglass [28] backbone, trained on COCO. We split the 2017 COCO validation set 60:40, calibrate on the first split and evaluate the calibrated detections on the smaller second hold out split. We calibrate class-wise for each of the 80 object categories to account for variations of different categories [30] and then split the detections of each class into three equally sized sub-groups B of bounding box sizes. Each sub-group is calibrated using histogram binning with 7 confidence bins C. The calibrated detections perform significantly worse with 35.7 mAP than the un-calibrated detections with 40.1 mAP (see Tab. 1). This result seems to contradict our initial reasoning and formal proof.

Modifying Histogram Binning. We take a closer look at histogram binning to understand why it drastically reduces the performance of the tested detector. Finding that it violates some of our prior assumptions, we modify the standard histogram binning calibration to actually verify our hypothesis, that we can use conditional calibration to improve prediction performance. To this end, we infuse histogram binning with the following assumptions.

Our first assumption is that calibration improves our ability to order the predictions according to their probability of being a TP. Histogram binning maps confidence ranges to a single estimated precision value, discarding the fine-grained confidence differences (see Fig. 2). A higher confidence implies a higher probability of being a correct prediction. Since we already split the detections according to their size into sub-groups, we can assume that detectors produce a meaningful confidence ordering within these subgroups: This is, after all, its training objective. As we want to maintain the ordering within each sub-group we add linear splines between the centers of histogram bins.

This leaves the question of how to extrapolate above the center of the last and below the center of the first bin. We add $\hat{f}(0) = 0$ and $\hat{f}(1) = 1$ as the outermost interpola-

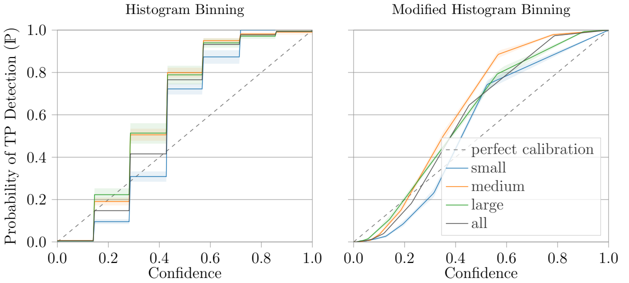

Figure 2. **Calibration curve of histogram binning and modified version** for category "person" with 3 size splits and 7 confidence bins (left) and 3 size splits and 14 confidence bins (right), respectively. With the 95% bootstrap confidence interval for supports. Note that the confidence interval is smaller for the modified version, despite double the number of bins.

Calibration Method	mAP	mAP_{50}
none	40.10	58.82
Conditional Histogram Binning	35.71(−4.39)	53.25(−5.57)
+ Linear Interpolation	37.54(−2.56)	55.80(−3.02)
+ Added Bounds	37.79(−2.31)	56.21(−2.61)
+ Adaptive Bins	39.84(−0.26)	58.16(−0.66)
+ Weighted Supports	40.40(+0.30)	59.18(+0.36)

Table 1. **Ablation of histogram binning modifications:** mAP and mAP_{50} of described modifications for $B = 3$ box splits and $C = 7$ confidence bins are shown.

tion points for the splines to get a mapping of the complete input range $(0, 1]$ to the possible probabilities $(0, 1]$. This encoded prior is also implicit in the beta calibration [19], as confidences reach their extrema, so should the probability of success: $\lim_{c\to 0} \mathbb{P} = 0$ and $\lim_{c\to 1} \mathbb{P} = 1$. The monotonic mapping is crucial at the boundaries 0 and 1 where the confidence values are concentrated, which is desirable property called sharpness [10].

The detector's sharpness also leads to problems estimating the probability of a TP prediction for each bin. Since the predictions' confidences c are non-uniformly distributed, the fixed confidence bins contain a varying amount of detections - sometimes even no detections, which is the case in Fig. 2 for predictions of small bounding boxes in the confidence interval $0.86 - 1.0$. Due to varying prediction densities some bins with few detections have a high variance. Nixon *et al*. discovered a similar problem in estimating the calibration error [30]. We follow their solution and implement a quantile binning scheme. Bin sizes for quantile binning are chosen such that each bin has the same number of predictions, thereby reducing differences in their variance.

We also set the support for the splines to be at the average confidence of the detections in each bin, to minimize errors from unevenly distributed confidences within each bin. The reduced variance at the supports along with all the modifications can be seen in Fig. 2 (right). We test each modification and the final modified calibration function on the same object detector as before. The results as seen in Tab. 1 verify the individual modifications and our original hypothesis, which is that the box size confidence bias reduces the performance of the object detector and our calibration can reduce this bias and increase the performance.

Other Calibration Methods. We briefly explore other calibration methods applied in a conditional calibration scheme. All methods relying on binning or producing calibration functions with flat regions deteriorate performance for conditional calibration (e.g. Platt Binning [20]: -4.19 mAP, Isotonic Regression [6]: -0.94 mAP). The best performing calibration method we found is the Beta calibration [19] (+0.14 mAP), about half as effective as our method. We continue our experiments using our modified histogram method for conditional calibrations. We explicitly do not claim that it is a superior calibration method. We only rely on it because its expressiveness is well suited to our goal of quantifying the performance impact of the conditional bias.

4.3. Quantifying Confidence Bias

We proved our initial assumption, but the question remains of how much impact the confidence bias has on the object detector's performance. To quantify this, we continue to try and maximize the performance improvement we can get from conditional calibration. The initial tests were performed by splitting the detections into 3 equally sized bounding box size B sub-groups and using 7 confidence bins C to calculate the spline supports. The number of splits are arbitrarily chosen parameters, so there are likely to be better choices for the split sizes since we can change the split sizes for each category. The number of instances per class can vary widely and so can the distributions of bounding box sizes for these instances for different categories. In the COCO validation dataset split, there are about 250 k instances for the most frequently occurring category and fewer than 1000 for the rarest category. There are bound to be different optima in the bias variance trade-offs for the confidence calibration. There is also the trade-off between the number of box size sub-groups B which enable a more fine grained estimate of the differences in box size-dependent confidence differences and the number of confidence bins C that increase the granularity of the calibration curve within a box size split. We need metrics to guide our choice of these parameters for each class.

Average Precision. The most obvious objective is to maximize the AP for each category to get the highest mAP on the evaluation split. We explored the connection between the expected AP and the correct ordering of the detections according to their probability of being a TP in Sec. 4.1. As a metric for quantifying the conditional bias, the AP can be vulnerable to confidence changes of outlier TP detections.

Proper Scoring Rules. A proper scoring rule is a function L that, in our notation, satisfies the condition $\mathbb{P}_i = \arg\max_{c_i} \mathbb{E}_{T_i} L(T_i, c_i)$ [11]. Its expected value is maximized for the TP probability $\mathbb{P}_i$ of the Bernoulli random variable T_i. The two most commonly used proper scoring rules are the squared difference, also called the Brier score [4] and the logarithmic score [12, 11]. We derive loss functions by negating the respective scoring functions. The Brier loss is then defined as

$$L_{\text{Brier}} = \frac{1}{N} \sum_{i=1}^{N} (c_i - \tau_i)^2, \tag{11}$$

while the log loss is defined as

$$L_{\log} = \frac{1}{N} \sum_{i=1}^{N} -\big[\tau_i \log(c_i) + (1 - \tau_i) \log(1 - c_i)\big]. \tag{12}$$

Both loss functions are minimized for any c_i only if $c_i = \mathbb{P}_i$. We do not want to estimate a single $\mathbb{P}_i$: we want to estimate $\mathbb{P}$ across the continuous calibration curve f. Both loss functions are minimized in expectation when $\mathbb{P}$ is estimated correctly, but the losses favor more confident predictions [47]. For each of the $\mathbb{P}_i$ values, deviations and outliers are also penalized by different loss magnitudes.

Mean Squared Error Estimation. Ideally, we would like to use an empirical loss function that corresponds to the expected squared error $\mathbb{E}_{d_i \in \mathcal{D}}[(\hat{f}(d_i) - \mathbb{P}_i)^2]$. We can only estimate $\mathbb{P}_i$ from the training data, which brings us right back to the original problem of needing good parameters to estimate the true calibration function f. The expected error can be split into its bias and variance components:

$$\mathbb{E}_{d \in \mathcal{D}}\Big[\big(\hat{f}(d) - f(d)\big)^2\Big] = \mathit{Bias}_{\mathcal{D}}\big(\hat{f}(\mathcal{D})\big)^2 + \mathit{Var}_{\mathcal{D}}\big(\hat{f}(\mathcal{D})\big). \tag{13}$$

The variance is easily estimated using K-folds to generate K calibration functions $\hat{f}_{B,C,k}$ and calculating the variance over the population of K calibrated confidences averaged over the entire calibration split. Calculating the actual bias would again require the true calibration function f, which is unknown. We can, however, estimate how much the bounding box size bias is reduced compared to a non-conditioned calibration scheme $\hat{f}_{1,C,k}$. We are then able to set the bias to the maximum bias reduction we achieved over the whole explored parameter range $|B| \times |C|$. Department

$$\begin{aligned}\widehat{\mathit{Bias}}_{B,C}\big(\hat{f}(\mathcal{D})\big) = &\max_{B,C}\left[\frac{1}{N}\sum_{i=1}^{N}\big(\hat{f}_{B,C}(d_i) - \hat{f}_{1,C}(d_i)\big)\right] \\ &- \frac{1}{N}\sum_{i=1}^{N}\big(\hat{f}_{B,C}(d_i) - \hat{f}_{1,C}(d_i)\big).\end{aligned} \tag{14}$$

Optimized Metric	ΔmAP	ΔmAP$_{50}$
Average Precision (AP)	$+0.09 \pm 0.10$	$+0.21 \pm 0.16$
Brier Loss (L_{Brier})	-0.15 ± 0.38	-0.22 ± 0.57
Log Loss ($L_{\log}$)	$+0.04 \pm 0.38$	$+0.03 \pm 0.62$
Est. Mean Squared Error ($L_{\widehat{\text{MSE}}}$)	$+0.25 \pm 0.18$	$+0.32 \pm 0.28$
oracle	$+0.67 \pm 0.08$	$+0.97 \pm 0.13$

Table 2. **Ablation of optimization metrics of calibration on validation split:** Comparison of performance change after calibrating and optimizing C and B with metrics on split of validation data and evaluating on the hold out set, average and max deviation of 10 random splits shown. Optimizing for L_{Brier} or $L_{\log}$ metrics does not improve the mAP, or even decreases it.

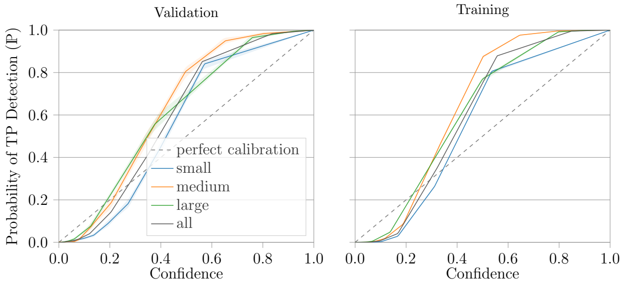

Figure 3. **Bounding box size bias on train and val data detections** for category "person", with ($B = 3$, $C = 10$) and ($B = 3$, $C = 20$) respectively. Note the 3 size bins here correspond to the official COCO size splits. Light shaded color represents the 95% bootstrap confidence interval.

With this, we can calculate $L_{\widehat{\text{MSE}}}$ as:

$$L_{\widehat{\text{MSE}}} = \left(\widehat{\mathit{Bias}}_{B,C}\big(\hat{f}(\mathcal{D})\big)^2 + \mathit{Var}_{\mathcal{D}}\big(\hat{f}(\mathcal{D})\big)\right). \tag{15}$$

We test the different metrics on the parameter search space of $B = \{2, 3, 4, 5, 6\}$ and $C = \{4, 5, 6, 8, 10, 12, 14\}$ and keep it constant for all following experiments. To our surprise the estimated mean squared error $L_{\widehat{\text{MSE}}}$ performs best among the optimization metrics, even compared to directly evaluating the AP on the calibration split (see Tab. 2). Still, optimizing for the $L_{\widehat{\text{MSE}}}$ achieves less than half the performance gain of an *oracle* evaluation on the hold-out split.

4.4. Bias in Training Predictions

One criticism of our applied conditional calibration approach might be that it captures the train test distribution shift. This would lead to us over-estimating the actual performance impact of the conditional confidence bias. Arguably, the additional data the calibration needs could finetune the detector. Motivated to solve this issue, we take a look at the object detector's predictions on the training data. The calibration curves Fig. 3 reveal striking similarities between the box-size confidence bias on the training and vali-

Optimized Metric	mAP	mAP$_{50}$
none	40.29	59.09
Average Precision (AP)	40.25(−0.04)	59.20(+0.11)
Brier Loss (L_{Brier})	40.36(+0.07)	59.28(+0.19)
Log Loss ($L_{\log}$)	40.37(+0.08)	59.28(+0.19)
Est. Mean Squared Error ($L_{\widehat{\text{MSE}}}$)	40.52(+0.23)	59.39(+0.30)
oracle	40.78(+0.49)	59.78(+0.69)

Table 3. **Ablation of optimization metrics of calibration on training data.** Calibration and parameter optimization on COCO training data, evaluated on validation data. Optimization metrics as described in Sec. 4.3. The $L_{\widehat{\text{MSE}}}$ is the best optimization metric and achieves about half the performance gain of oracle evaluation. The AP performs even worse than on the validation splits and the L_{Brier} and $L_{\log}$ perform slightly better (compare Tab. 2).

dation set. They are not identical, but the bias could be similar enough in the training dataset predictions to get a comparable performance gain to the calibration on a validation data split. The calibration curves also show a smaller confidence interval for the training data predictions. The training data with about 860 k detections is significantly larger than the validation data with only about 37 k detections. The added detections enable a more fine-grained estimation of the reliability diagram, which is indicated by the smaller confidence intervals on the training split (see Fig. 3). We test the calibration on the detector's predictions on the train set and verify them on the whole validation data. The performance changes, shown in Tab. 3, are very similar to the validation data split (see Tab. 2).

4.5. Test Time Augmentation

TTA is widely used in image recognition tasks to improve prediction performance [40, 54]. It is used to produce better and more reliable predictions from a prediction model. The model predictions for an image are generated across different image augmentations, usually geometric transformations, and for object detectors by up- and down-scaling of the images by constant factors. The predictions of the detectors are then combined using some form of Non-maximum suppression [37] (NMS). When the image is down-scaled the model is forced to predict objects with smaller bounding boxes and, according to our reasoning, this should exaggerate the observed confidence bias. As we argued in Sec. 4.1, when differently calibrated sub-groups, the bounding box size groups in this case, are combined the performance is non-optimal and our calibration scheme should improve performance. We can also define the predictions of the detector for one augmentation as a subgroup within all TTA predictions and calibrate them separately to satisfy Eq. (10). Our experiments indeed show, that combining the individually calibrated predictions of each scale augmentation is about three times as effective as only calibrating the combined predictions (see Tab. 4). This indicates that the performance impact of conditional confidence bias is confounded by TTA.

Augmentation	Cond. Calib.	mAP	mAP$_{50}$
0.50x	-	34.45	51.85
	✓	34.70(+0.25)	52.13(+0.28)
0.75x	-	40.82	59.35
	✓	41.06(+0.24)	59.62(+0.27)
1.00x	-	42.25	61.13
	✓	42.49(+0.24)	61.48(+0.35)
1.25x	-	40.80	59.98
	✓	41.08(+0.28)	60.45(+0.47)
1.50x	-	38.53	56.94
	✓	38.78(+0.25)	57.39(+0.45)
TTA + NMS	-	44.95	64.12
	✓	45.08(+0.13)	64.30(+0.18)
cal.(TTA) + NMS	(✓)	45.43(+0.48)	64.64(+0.52)

Table 4. **Effect of individual calibration on TTA,** calibration on COCO train, evaluated on validation data. Compares performance of Centernet with TTA (0.5x, 0.75x, 1.0x, 1.25x, 1.5x-scale augmentation), without calibration, with conditional calibration and with predictions for each augmentation separately conditionally calibrated.

5. Evaluation and Discussion

Finally, to estimate the box size bias's impact on different detection architectures, we tested our calibration method on a wide range of deep learning object detectors. We kept the parameter search space as before, calibrate on the COCO train split, and evaluated on the validation and test-dev splits. The results are shown in Tab. 5. The performance changes vary across the different architectures, ranging from a slight decrease of 0.1 mAP and mAP$_{50}$ to a large gain of 0.6 mAP and 0.8 mAP$_{50}$. The largest gains are on par with some proposed model improvements highlighting the impact the box size bias can have in practice. There are a variety of influences and limitations on the performance impacts.

Model Size. The performance change seems to be negatively correlated with the object detectors model size and absolute performance. To verify this trend we test our calibration method on the popular EfficientDet [42], which is available in 8 different size and performance implementation. The trend also holds within this architecture type across the model size scales, with minor outliers (see supplementary material for exact values). The larger models appear to learn to reduce the conditional bias to some extent.

Two Stage vs. One Stage. Our approach does not lead to performance improvements on both two-stage detectors,

Detector	Backbone	Cond. Calibration	Validation		Test-Dev	
			mAP	mAP_{50}	mAP	mAP_{50}
YOLOv5X [16]	CSPDarkNet-53 [35, 45]	-	50.38	68.76	50.3	68.4
		train	50.42(+0.04)	68.81(+0.05)	50.3(+0.0)	68.4(+0.0)
		oracle	50.63(+0.25)	69.04(+0.28)	-	-
RetinaNet [23]†	ResNeXt-101 [48]	-	40.82	60.48	41.2	61.1
		train	40.79(−0.03)	60.53(+0.05)	41.2(+0.0)	61.2(+0.1)
		oracle	41.09(+0.27)	60.76(+0.28)	-	-
CenterNet [54]	Hourglass-104 [28]	-	40.29	59.10	40.2	59.1
		train	40.59(+0.30)	59.53(+0.43)	40.5(+0.2)	59.5(+0.4)
		oracle	40.80(+0.51)	59.85(+0.75)	-	-
DETR [5]†	ResNet-50 [14]	-	40.11	60.62	39.9	60.7
		train	40.39(+0.28)	60.68(+0.06)	40.3(+0.4)	60.8(+0.1)
		oracle	40.72(+0.61)	60.86(+0.24)	-	-
CenterNet [54]†	ResNet-18 [14]	-	29.55	46.14	29.7	46.7
		train	30.06(+0.51)	46.87(+0.73)	30.3(+0.6)	47.5(+0.8)
		oracle	30.34(+0.79)	47.20(+1.06)	-	-
YOLOv3-320 [35]†	DarkNet-53 [35]	-	27.91	49.10	27.8	49.0
		train	28.22(+0.31)	49.65(+0.55)	28.1(+0.3)	49.5(+0.5)
		oracle	28.54(+0.63)	49.95(+0.85)	-	-
SSD300 [25]‡	ResNet-50 [14]	-	25.03	42.33	24.9	42.5
		train	25.17(+0.14)	42.58(+0.25)	24.9(+0.0)	42.7(+0.2)
		oracle	25.34(+0.31)	42.83(+0.50)	-	-
CenterNet2 [53]	ResNet-50 [14]	-	42.86	59.52	43.1	59.9
		train	42.84(−0.02)	59.50(−0.02)	43.1(+0.0)	59.9(+0.0)
		oracle	43.08(+0.22)	59.74(+0.22)	-	-
Faster-RCNN [36]†	ResNeXt-101 [48]	-	41.60	61.93	42.0	62.8
		train	41.52(−0.08)	61.90(−0.03)	41.9(−0.1)	62.7(−0.1)
		oracle	41.80(+0.20)	62.06(+0.13)	-	-

Table 5. **Conditional calibration method on different models to estimate performance impact of confidence bias.** Calibration and parameter optimization on COCO train, evaluated on validation data and test-dev benchmark. Models are sorted by performance and separated into one- and two-stage architectures. Official implementation are evaluated unless noted as: †: [7], ‡: [32]

even with the *oracle* parameter choice there are little performance changes (see Tab. 5). When the region proposal is separated from the classification and confidence prediction, as in two-stage detectors, it makes sense that it is less likely that a conditional bias on the confidence predictions with respect to the bounding box values is introduced.

Limitations. Detectors that are conditionally calibrated on the training data should not be considered well calibrated since the object detectors performance on the training data usually overestimates its performance on unseen data. For a calibration beyond the reduced confidence bias, it has to be performed on a hold-out set.

Broader Impact. Object detectors are part of many real-world systems, most of which have a positive impact, but there are also many systems with negative societal impacts. These harms can be caused by unintended biases or intentionally, as in autonomous weapons systems. Our proposed de-biasing can be used to increase the performance of object detectors, so it can clearly be used to increase the harm of intentionally designed harmful systems. We believe, however, that our research will have a net-positive impact, as it can improve applications with a positive impact and also reduce unintended harms caused by biased and uncalibrated predictions.

6. Conclusion

We formally proved that the conditional confidence bias is non-optimal for object detectors' performance. To quantify the performance impact on object detectors, we show how a slightly modified version of the popular histogram binning can be leveraged to compensate the bounding box bias and improve the performance of the object detectors. We show that this is even possible on the detectors' training data. This underlines that the performance improvements stem from removing the conditional box size confidence bias, not a train-test data distribution shift. We utilized our proposed algorithm to analyse a diverse set of object detection architectures and show that the conditional confidence bias harms their performance by up to 0.6 mAP and 0.8 mAP_{50}. Our formal and empiric results, linking conditional bias to object detectors performance, demonstrate the crucial need for researchers and practitioners to pay closer attention to conditional biases and confidence calibration.

References

[1] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *ECCVW*, pages 556–572, 2018.

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, pages 528–539, 2020.

[3] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[4] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[6] Nilotpal Chakravarti. Isotonic median regression: a linear programming approach. *Mathematics of operations research*, 14(2):303–308, 1989.

[7] Kai et al. Chen. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. (Apache-2.0): `https://github.com/open-mmlab/mmdetection`.

[8] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[10] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

[11] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[12] Irving J Good. Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114, 1952.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330, 2017.

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

[15] Lingxiao Huang and Nisheeth Vishnoi. Stable and fair classification. In *ICML*, pages 2879–2890, 2019.

[16] Glenn Jocher. YOLOv5, 2020. (GNU GPL): `https://github.com/ultralytics/yolov5`.

[17] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, pages 9012–9020, 2019.

[18] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *ICCV*, pages 14992–15001, 2021.

[19] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631, 2017.

[20] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[21] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, volume 80, pages 2805–2814, 2018.

[22] Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *CVPRW*, pages 326–327, 2020.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[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, 2014. (CC-BY 4.0): `https://cocodataset.org`.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[26] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

[27] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. In *NeurIPSW*, 2018.

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016.

[29] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

[30] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPRW*, volume 2, 2019.

[31] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020.

[32] Adam et al. Paszke. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. (BSD-3): `https://pytorch.org/`.

[33] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[34] Geoff Pleiss, Manish Raghavan, JonKleinberg FelixWu, and KilianQ Weinberger. On fairness and calibration. In *NeurIPS*, pages 5680–5689, 2017.

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

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *PAMI*, 28:91–99, Jun 2017.

[37] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on Computers*, 100(5):562–569, 1971.

[38] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *CVPR*, pages 11070–11078, 2020.

[39] Tomáš Sixta, Julio CS Jacques Junior, Pau Buch-Cardona, Eduard Vazquez, and Sergio Escalera. Fairface challenge at eccv 2020: Analyzing bias in face recognition. In *ECCV*, pages 463–481, 2020.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[42] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. (Apache-2.0): `https://github.com/google/automl/tree/master/efficientdet`.

[43] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011.

[44] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *Artificial Intelligence and Statistics*, volume 89, pages 3459–3467, 2019.

[45] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *CVPRW*, pages 1571–1580, 2020.

[46] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, pages 8919–8928, 2020.

[47] Robert L Winkler and Allan H Murphy. "good" probability assessors. *Journal of Applied Meteorology and Climatology*, 7(5):751–758, 1968.

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

[49] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, volume 1, pages 609–616, 2001.

[50] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[51] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, pages 325–333, 2013.

[52] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[53] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. (Apache-2.0): `https://github.com/xingyizhou/CenterNet2`.

[54] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. (MIT License): `https://github.com/xingyizhou/CenterNet`.

[55] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019.