Supplementary: The Box Size Confidence Bias Harms Your Object Detector

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

1. Effect of Model Size

We observed that the performance change from conditional calibration appears to be negatively correlated with the object detectors model size and performance. We verify the observation on the EfficientDet [7], which is available in 8 different size and performance versions. This ensures that there are no other influences, such as loss functions, augmentations or similar factors. The trend largely holds for the different EfficientDet model variants, but there ares some minor outliers (see Tab. 2).

2. Parameter Search Space

We chose a fixed search space of $B_0 = \{2, 3, 4, 5, 6\}$ and $C_0 = \{4, 5, 6, 8, 10, 12, 14\}$, which we kept constant to have comparable results for all detectors and methods. We now take a closer look at the influence the search space has on the performance. We define two additional sets $B_1 = \{8, 10, 12, 14, 20\}$ and $C_1 = \{14, 16, 18, 20, 24, 28, 34, 40, 50\}$ and explore different combinations of the four sets for the parameter search space. The results show that a larger search space increases the performance changes (see Tab. 1). If, however, the search space excludes low values for the number of confidence bins like in C_1, the performance for categories with few detections is decreased.

3. Optimization Metrics

There are a range of metrics that could be explored for the optimization of the bin size parameter space. The explored AP, L_{Brier}, $L_{\log}$, and $L_{\widehat{\text{MSE}}}$ each have a good theoretical justification for usage in this application. We explore some of the possible metrics which we did not include in the main section, and give justification for their exclusion.

Absolute Difference. The absolute difference, or absolute deviations, could be considered a reasonable choice besides L_{Brier} and $L_{\log}$. It is calculated as

$$L_{\text{diff}} = \frac{1}{N} \sum_{i=1}^{N} |c_i - \tau_i|, \tag{1}$$

but, in contrast to L_{Brier} and $L_{\log}$, it is not a proper scoring rule [1]. It is not minimized for $c_i = \mathbb{P}_i$, but rather by the majority label, *i.e.* by $c_i = 1$ for $\mathbb{P}_i > 0.5$ and $c_i = 0$ for $\mathbb{P}_i < 0.5$. Unsurprisingly, it performs even worse than the proper scoring rules for the performance measured in mAP (see Fig. 1) and mAP_{50} (see Fig. 2)

Expected Calibration Error. Since our goal is to perform a conditional confidence calibration a intuitive choice for the optimization metric is the Expected Calibration Error (ECE) [4]. If we let $\hat{f}_{1,C}$ be the un-modified histogram binning with C confidence bins, the ECE is calculated as:

$$\text{ECE} = \frac{1}{N} \sum_{i=1}^{N} |c_i - \hat{f}_{1,C}(d_i)|. \tag{2}$$

The ECE is also a proper scoring rule [1], but it also has its limitations in general [5] and for this application: It only tries to captures the calibration error not the conditional calibration bias. For the parameter optimization we follow [2] and set the number of confidence bins to $C = 10$. We calculate the ECE separately for each class because we want search for the class-wise optimal parameters. Because of the described drawbacks the ECE does not perform well as a optimization metric (see Fig. 1).

Estimated AP. Instead of the 11-point maximum interpolated AP metric used by the COCO benchmark-evaluation we could use the the AP computed over all the available detections for each class:

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

To distinguish it from the COCO-benchmark AP metric we refer to it as $\text{AP}_{\text{est.}}$. On average, $\text{AP}_{\text{est.}}$ performs almost as good as AP when used as the optimization metric. It is, however, also more susceptible to outliers and can thereby sometimes severely degrade the performance on the hold-out set (see Fig. 1).

Search Space		Calibration (train)		*Oracle*	
Box Bins	Confidence Bins	mAP	mAP_{50}	mAP	mAP_{50}
B_0	C_0	40.52(+0.23)	59.39(+0.30)	40.78(+0.49)	59.78(+0.69)
B_0	C_1	40.43(+0.14)	59.48(+0.39)	40.88(+0.59)	59.88(+0.79)
$B_0 \cup B_1$	C_0	40.61(+0.32)	59.57(+0.48)	40.89(+0.60)	59.89(+0.80)
$B_0 \cup B_1$	$C_0 \cup C_1$	40.61(+0.32)	59.58(+0.49)	40.99(+0.70)	59.99(+0.90)
Baseline		40.29	59.09	40.29	59.09

Table 1. **Influence of parameter search spaces on performance changes.** Performance of calibrated CenterNet detector with parameters optimized with $L_{\widehat{\text{MSE}}}$ metric and *oracle* evaluation. Calibration on COCO train split, evaluation on validation data. Larger search space enables larger performance gains, but excluding smaller sized confidence bins from the search space (C_0) can reduce mAP when optimizing for the $L_{\widehat{\text{MSE}}}$ metric.

Version	#Param.	Calib.	mAP	mAP_{50}
D0	3.9M	-	34.24	52.48
		train	34.30(+0.06)	52.62(+0.14)
		oracle	34.50(+0.26)	53.00(+0.52)
D1	6.6M	-	40.09	58.85
		train	40.16(+0.07)	58.95(+0.10)
		oracle	40.33(+0.24)	59.30(+0.45)
D2	8.1M	-	43.38	62.52
		train	43.42(+0.04)	62.64(+0.12)
		oracle	43.61(+0.23)	62.99(+0.47)
D3	12.0M	-	47.05	65.86
		train	47.08(+0.03)	65.90(+0.04)
		oracle	47.23(+0.18)	66.18(+0.32)
D4	20.7M	-	49.15	68.24
		train	49.16(+0.01)	68.27(+0.03)
		oracle	49.33(+0.18)	68.58(+0.34)
D5	33.7M	-	51.03	70.09
		train	51.08(+0.05)	70.16(+0.07)
		oracle	51.25(+0.22)	70.45(+0.36)
D6	51.9M	-	51.99	70.94
		train	52.00(+0.01)	70.98(+0.04)
		oracle	52.17(+0.18)	71.27(+0.33)
D7	51.9M	-	53.06	72.12
		train	53.05(−0.01)	72.14(+0.02)
		oracle	53.21(+0.15)	72.42(+0.30)

Table 2. **Influence of calibration method on different sized versions of EfficientDet [7].** Ordered by increasing model size: Calibration and parameter optimization on COCO train, evaluated on validation data. The calibration is not very effective and its impact decreases with increasing model size.

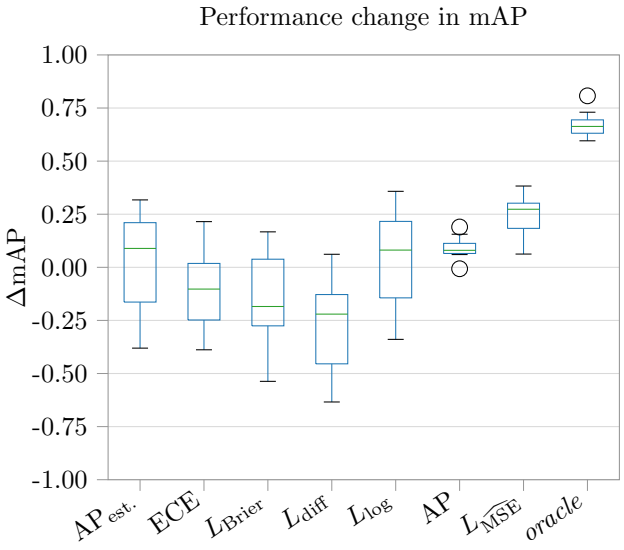

Figure 1. **Boxplot of Performance Change in mAP for Extended Optimization Metrics.** CenterNet [8] calibrated on 60% of COCO validation split, evaluated on the remaining 40% with 10 random splits. The box ranges from the lower to upper quantile values, the green line is the median performance change.

4. Extended Look at Maximizing AP

Taking an extended look at the formal proof we start again with the expected $\text{AP}_{t_{\text{IoU}}}$:

$$\mathbb{E}_T[\text{AP}_{t_{\text{IoU}}}] = \mathbb{E}_T\left[\sum_{i=1}^{N} \text{Prec}(i) \cdot \Delta\text{Rec}(i)\right]. \tag{4}$$

Substituting precision and recall and our stochastic indicator variable T we first get:

$$\mathbb{E}_T[\text{AP}_{t_{\text{IoU}}}] = \mathbb{E}_T\left[\sum_{i=1}^{N} \left(\frac{\sum_{k=1}^{i}(T_k)}{i} \cdot \frac{T_i}{|\mathcal{G}|}\right)\right]. \tag{5}$$

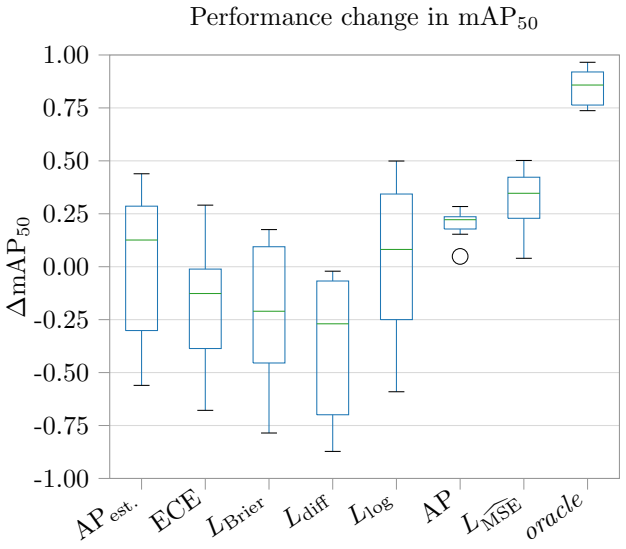

Figure 2. **Boxplot of Performance Change in mAP$_{50}$ for Extended Optimization Metrics.** Same settings as in Fig. 1

We can move T_i out of the inner sum:

$$\mathbb{E}_T[\mathrm{AP}_{t_{\mathrm{IoU}}}] = \mathbb{E}_T\left[\sum_{i=1}^{N}\left(\frac{\sum_{k=1}^{i-1}(T_k)+T_i}{i}\cdot\frac{T_i}{|\mathcal{G}|}\right)\right]. \quad (6)$$

We assume independence of T_n and T_m for every m, n with $m \neq n$. This is actually only the case if the detections d_n and d_m don't try to detect the same ground truth object. The introduced error, however, is minuscule on a large dataset the number of detections for the same ground truth object are significantly smaller than the overall number of detections. The number of detections for the same object are further decreased through non-maximum suppression [6] (NMS).
The number of detections $|\mathcal{G}|$ is constant and can be moved to the front. Since $T_i \sim \mathrm{Bernoulli}(\mathbb{P}_i)$ it follows that $(T_i)^2 = T_i$.

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

First we move the inner sum to the back,

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

and then reformulate it. First we can split the outer sum into the sum over the first term and the double-sum over the second term:

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

Then we can switch the sums and their limits:

$$\mathbb{E}_T[\mathrm{AP}_{t_{\mathrm{IoU}}}] = \frac{1}{|\mathcal{G}|}\sum_{i=1}^{N}\left(\frac{\mathbb{P}_i}{i}\right)+\sum_{i=1}^{N}\sum_{k=i+1}^{N}\frac{(\mathbb{P}_i\cdot\mathbb{P}_k)}{k}. \quad (10)$$

Then we re-combine the term of the first sum into the outer of the second sums:

$$\mathbb{E}_T[\mathrm{AP}_{t_{\mathrm{IoU}}}] = \frac{1}{|\mathcal{G}|}\sum_{i=1}^{N}\underbrace{\left(\frac{\mathbb{P}_i}{i}+\mathbb{P}_i\sum_{k=i+1}^{N}\frac{\mathbb{P}_k}{k}\right)}_{h_i(\mathbb{P}_i,\mathbb{P})}. \quad (11)$$

Here we see that $h_i(l,\mathbb{P}) > h_{i+1}(l,\mathbb{P})$ for $i \in \mathbb{N}$ and $l \in (0,1]$. This can be seen more clearly if we split $h_i(l,\mathbb{P})$ into the two components of its sum (I) and(II),

$$h_i(\mathbb{P}_i,\mathbb{P}) = \underbrace{\frac{\mathbb{P}_i}{i}}_{\text{(I)}} + \underbrace{\mathbb{P}_i\sum_{k=i+1}^{N}\frac{\mathbb{P}_k}{k}}_{\text{(II)}}, \quad (12)$$

for the (I) it is obvious that for any $l \in (0,1]$ and $i \in \mathbb{N}$:

$$\frac{l}{i} > \frac{l}{i+1}. \quad (13)$$

For the second term (II) we can see that for any i it can be split as follows:

$$l\sum_{k=i+1}^{N}\frac{\mathbb{P}_k}{k} = l\underbrace{\frac{\mathbb{P}_{i+1}}{i+1}}_{\text{(III)}} + l\sum_{k=i+2}^{N}\frac{\mathbb{P}_k}{k}, \quad (14)$$

where (III) is > 0 for $l \in (0,1]$ and $i \in \mathbb{N}$ and the second term is (II) of $h_{i+1}(l,\mathbb{P})$. Which proves that $h_i(l,\mathbb{P})$ is strictly larger than $h_{i+1}(l,\mathbb{P})$ in the relevant intervals $l \in (0,1]$ and $i \in \mathbb{N}$. So the expected $\mathrm{AP}_{t_{\mathrm{IoU}}}$ is a sum of functions h, that given the same input value have strictly decreasing output for larger values of i. It can thereby be maximized for some fixed set of $\mathcal{D}$, by sorting the detections by their $\mathbb{P}$. Since the detections are already sorted according to their c for the evaluation we need to ensure that this also sorts $\mathbb{P}$ i.e. that the confidence calibration curve is monotonic:

$$\mathbb{P}_n < \mathbb{P}_m \;\forall\, n, m \,|\, c_n < c_m. \quad (15)$$

Under the assumption that this condition then holds across different t_{IoU}'s it also maximises the mAP. The actual influence of t_{IoU} is discussed in the next section.

5. Intersection over Union (IoU) Threshold

The calibrations are all performed with $t_{\mathrm{IoU}} = 0.5$; we analyze the influence of this choice in Fig. 3. Unsurprisingly,

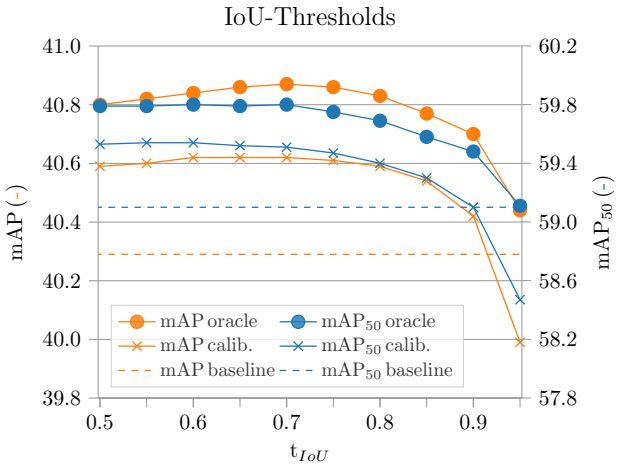

Figure 3. **Ablation of IoU thresholds,** performance of conditionally calibrated CenterNet for different values of t_{IoU} that determine the required bounding box overlap for TP detections.

the mAP$_{50}$ metric is maximized for t_{IoU} of around 0.5, since this is also the threshold used for identifying TP detection. Slightly higher performance in mAP can be achieved with $t_{\text{IoU}} \approx 0.7$ since this is roughly the median of the threshold range for the evaluation of the mAP, as described in the background section. Regardless of the t_{IoU} the performance change does not vary by much up to a t_{IoU} of 0.95. For t_{IoU} =0.95 our assumption made in Sec. 4 starts to break down. An good confidence ordering with t_{IoU} =0.95 is not a good predictor for the best ordering for smaller t_{IoU}. This is likely due to a significantly reduced number of TP predictions which leads to a higher variance in the estimate of $\mathbb{P}$ and a consequently less accurate $\hat{f}$.

References

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

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

[3] 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`.

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

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

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

[7] 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`.

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