Supplementary Material
Bridging Precision and Confidence: A Train-Time Loss for Calibrating Object Detection

Muhammad Akhtar Munir[1,2] Muhammad Haris Khan[1] Salman Khan[1,3] Fahad Shahbaz Khan[1,4]
[1]Mohamed bin Zayed University of AI [2]Information Technology University
[3]Australian National University [4]Linköping University

1. Overview

In this supplementary material, we provide the following items:

A.1 Further implementation details.
A.2 Additional experimental results.
A.3 More qualitative results.
A.4 Architecture Irrelevant: BPC with One-stage.
A.5 Formulation for Semantic Segmentation.
A.6 Error bars: Mean & Std. Dev.
A.7 More results on Common Corruptions.

A.1 Further Implementation Details

We choose Deformable-DETR (D-DETR) for our experiments. Our BPC loss is an auxiliary loss that jointly trains with the object detection losses to achieve better calibration. For our experiments, we use multi-gpu settings (4 GPUs) for training.

We provide further details on experimental settings for PascalVOC to watercolor1k, clipart1k, and comic1k domain shifts. We utilize train set of PascalVOC 2007 and 2012 for training and validation set of PascalVOC 2012 is used for evaluation purpose. For post-hoc method, it needs a hold-out validation set, and PascalVOC 2007 test set is used for that purpose. PascalVOC contains 20 categories of real images. Other out-domain datasets contain evaluation images, and respective 1k test set images are used for watercolor1k and comic1k, while whole 1k images of clipart1k are used for evaluation. Clipart1k also contains 20 categories, whereas 6 common categories are evaluated in watercolor1k and comic1k. We report calibration error (D-ECE) and mean average precision for reporting results.

A.2 Additional Experimental Results

PascalVOC: We compare the performance of our calibration loss with recent calibration methods, post-hoc calibration method, and the baseline on PASCALVOC to watercolor1k, clipart1k, and comic1k domain shifts. In Tab. 1, we present the results on PascalVOC, and our BPC loss reduces the D-ECE by 9.7%↓ over MbLS [4] and 1.6%↓ over MDCA [4].

Methods \ Scenario	In-Domain (PascalVOC)		
	D-ECE ↓	AP box	mAP@0.5
Baseline [6]	11.8	49.7	73.8
TS (post-hoc) [1]	13.0	49.7	73.8
MDCA [2]	12.8	48.9	73.2
MbLS [4]	20.9	49.7	73.6
BPC (Ours)	11.2	49.6	74.0

Table 1. Calibration results with baseline, train-time losses and post-hoc methods are reported. BPC shows improvement in detection calibration for in-domain scenario. AP box and mAP@0.5 are also reported.

Watercolor1k: In Tab. 2, our BPC loss improves the calibration performance without losing significant detection accuracy. Our loss shows calibration improvement of 7.0%↓ over MbLS [4] and 10.0%↓ over post-hoc method.

Clipart1k: Tab. 2 shows that our BPC loss comparatively improves the calibration performance without losing detection accuracy. Our loss shows calibration improvement of 1.5%↓ over MbLS [4] and 8.4%↓ over post-hoc method.

Comic1k: Our BPC loss improves the calibration performance along with the detection accuracy. In Tab. 2, our loss shows calibration improvement of 5.6%↓ over MbLS [4] and 3.2%↓ over MDCA [2].

A.3 More Qualitative Results

We show more qualitative calibration results on Cor-COCO (corrupted version of MS-COCO 2017 validation set) in Figs. 1 and 2. Detector trained with our loss forces the accurate predictions to be more confident whereas inaccurate predictions to be less confident.

Scenarios / Methods	Out-Domain (watercolor1k)			Out-Domain (clipart1k)			Out-Domain (comic1k)		
	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5
Baseline [6]	11.1	15.9	31.8	11.0	9.0	17.9	14.4	5.6	11.1
TS (post-hoc) [1]	19.3	15.9	31.8	19.2	9.0	17.9	22.0	5.6	11.1
MDCA [2]	9.8	17.5	34.8	10.5	9.3	17.8	15.3	4.9	8.8
MbLS [4]	16.3	16.8	34.5	12.3	9.2	17.9	17.7	5.5	10.4
BPC (Ours)	9.3	16.5	34.1	10.8	10.2	19.1	12.1	6.1	11.7

Table 2. Comparison of calibration performance with the baseline, train-time losses and post-hoc methods. Our BPC shows improvement over almost all competing approaches in three challenging out-domain scenarios i.e. from PASCALVOC to watercolor1k, clipart1k & comic1k. AP box and mAP@0.5 are also reported for each scenario.

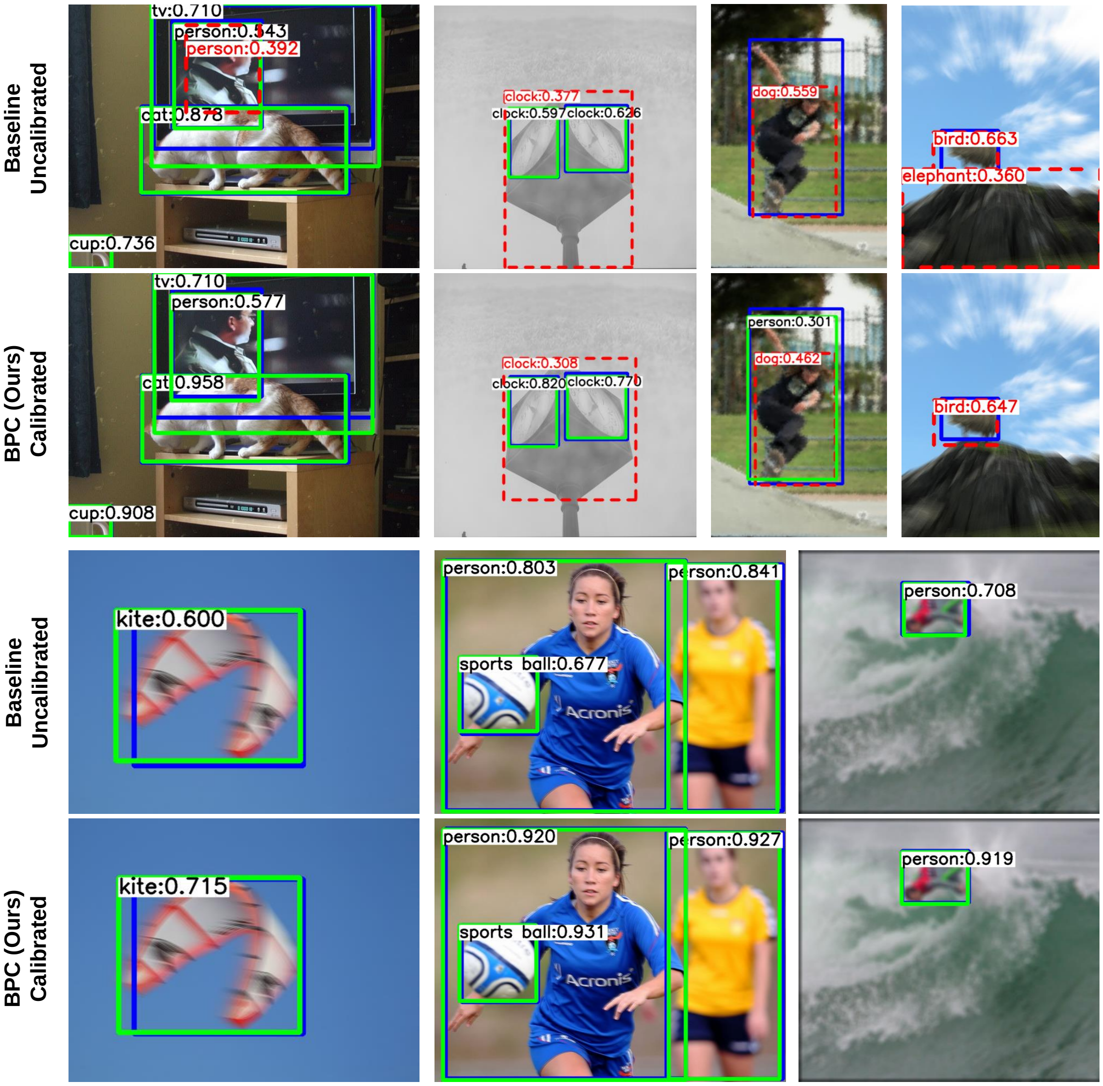


Figure 1. Baseline [6] vs. BPC (Ours): Qualitative results on CorCOCO dataset (Out-Domain of MS-COCO). Detector trained with our loss forces the accurate predictions to be more confident whereas inaccurate predictions to be less confident. Detection threshold is set to 0.3. Green boxes are accurate predictions with their respective confidence scores. Red (dashed) boxes are inaccurate predictions with corresponding scores. Blue shows the ground truth boxes for corresponding detections.

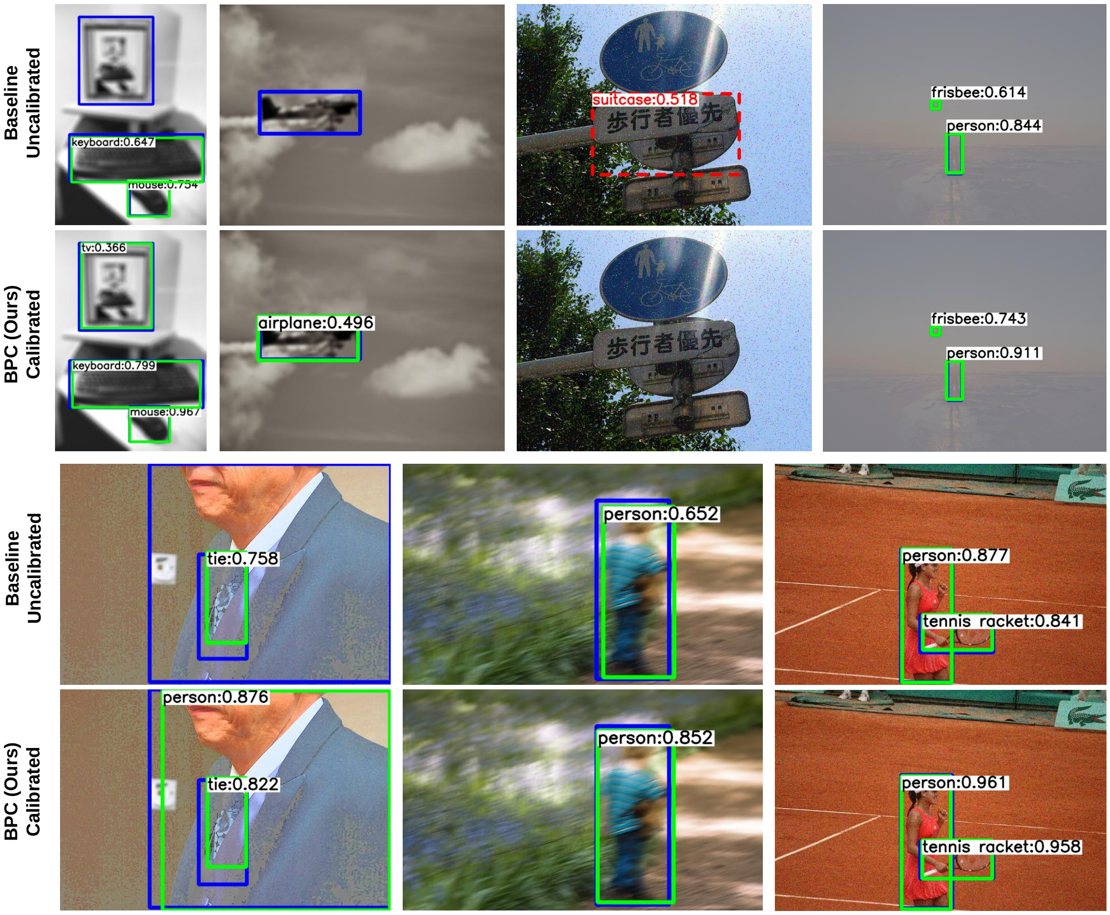


Figure 2. Baseline [6] vs. BPC (Ours): Qualitative results on CorCOCO dataset (Out-Domain of MS-COCO). Detector trained with our loss forces the accurate predictions to be more confident whereas inaccurate predictions to be less confident. Detection threshold is set to 0.3. Green boxes are accurate predictions with their respective confidence scores. Red (dashed) boxes are inaccurate predictions with corresponding scores. Blue shows the ground truth boxes for corresponding detections.

Method	In-Domain (COCO)			Out-Domain (CorCOCO)		
	D-ECE ↓	AP box	mAP	D-ECE ↓	AP box	mAP
One-stage (FCOS)	22.0	38.7	57.2	24.7	20.4	32.1
BPC (Ours)	20.9	38.4	56.9	23.5	20.3	32.0

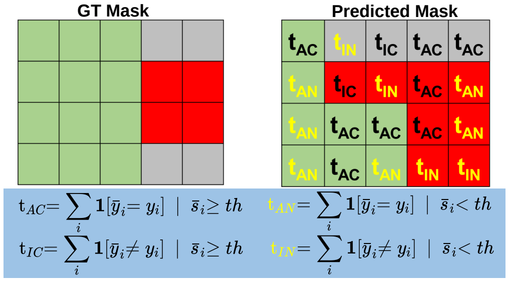
Table 3. Calibration results on One-Stage object detector (FCOS).

A.4 Architecture Irrelevant: BPC with One-stage

In Tab. 3, we show results with a one-stage detector (FCOS [5] with ResNet-50). Our BPC loss improves calibration in both in-domain (COCO) and out-domain (CorCOCO).

A.5 Formulation for Semantic Segmentation

Our loss BPC is extensible for semantic segmentation. We provide a sketch formulation to extend BPC loss for semantic segmentation (Fig. 3).



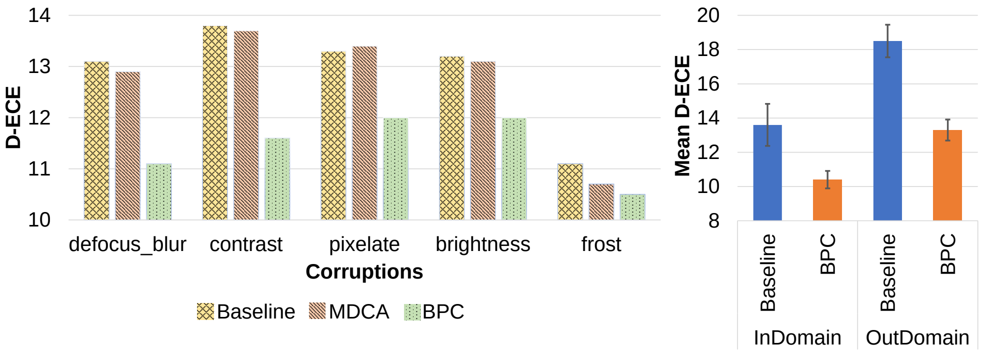
$$t_{AC} = \sum_i \mathbf{1}[\bar{y}_i = y_i] \mid \bar{s}_i \geq th \qquad t_{AN} = \sum_i \mathbf{1}[\bar{y}_i = y_i] \mid \bar{s}_i < th$$

$$t_{IC} = \sum_i \mathbf{1}[\bar{y}_i \neq y_i] \mid \bar{s}_i \geq th \qquad t_{IN} = \sum_i \mathbf{1}[\bar{y}_i \neq y_i] \mid \bar{s}_i < th$$

Figure 3. BPC loss for Semantic Segmentation task. 3 classes as box colors (green, red & grey). Black text in box (confident) and yellow (not-confident) predictions.

A.6 Error bars: Mean & Std. Deviation

Fig. 4(b) plots the mean and std.dev D-ECE for BPC and baseline in CS (in-domain) and Foggy-CS (out-domain).



(a) Calibration performance (Corruptions)

(b) Mean & Std.dev

Figure 4. (a) Calibration performance on COCO corruptions. (b) Error bars on CS/Foggy-CS dataset.

A.7 More results on Common Corruptions

We corrupt COCO evaluation set (val2017) after sampling 5 different corruptions with a fixed severity level of 2 from Common Corruptions [3]. Fig. 4(a) shows that our BPC loss can improve the calibration performance on all five corruptions.

References

[1] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 1, 2

[2] Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16081–16090, June 2022. 1, 2

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019. 4

[4] Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 80–88, June 2022. 1, 2

[5] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 3

[6] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 1, 2, 3