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
{akhtar.munir,muhammad.haris,salman.khan,fahad.khan}@mbzuai.ac.ae

Abstract

Deep neural networks (DNNs) have enabled astounding progress in several vision-based problems. Despite showing high predictive accuracy, recently, several works have revealed that they tend to provide overconfident predictions and thus are poorly calibrated. The majority of the works addressing the miscalibration of DNNs fall under the scope of classification and consider only in-domain predictions. However, there is little to no progress in studying the calibration of DNN-based object detection models, which are central to many vision-based safety-critical applications. In this paper, inspired by the train-time calibration methods, we propose a novel auxiliary loss formulation that explicitly aims to align the class confidence of bounding boxes with the accurateness of predictions (i.e. precision). Since the original formulation of our loss depends on the counts of true positives and false positives in a minibatch, we develop a differentiable proxy of our loss that can be used during training with other application-specific loss functions. We perform extensive experiments on challenging in-domain and out-domain scenarios with six benchmark datasets including MS-COCO, Cityscapes, Sim10k, and BDD100k. Our results reveal that our train-time loss surpasses strong calibration baselines in reducing calibration error for both in and out-domain scenarios. Our source code and pre-trained models are available at https://github.com/akhtarvision/bpc_calibration

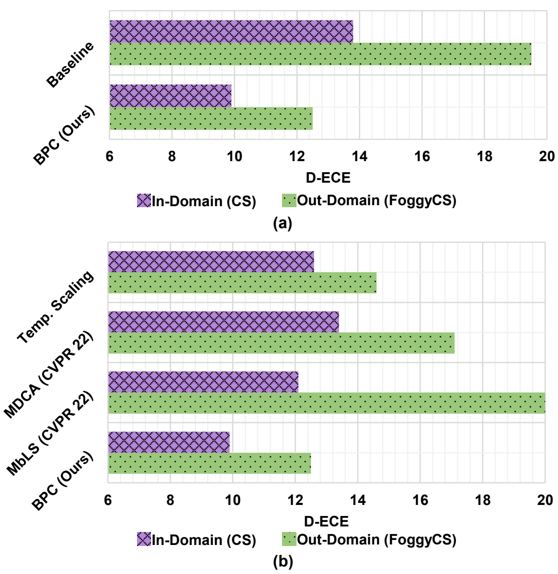

Figure 1. Comparison in terms of Detection Expected Calibration Error (D-ECE) on in-domain Cityscapes (CS) and out-domain FoggyCS datasets. (a) Detection model trained with our proposed BPC loss provides the lowest D-ECE (lower is better). (b) Post-hoc and train-time calibration methods for classification: MDCA [11] and MbLS [25] are sub-optimal for object detection. In contrast, BPC loss better calibrates in-domain and out-domain detections.

1. Introduction

Deep neural networks (DNNs) have shown remarkable results in various mainstream computer vision tasks, including image classification [5, 10, 33], object detection [29, 36, 39], and semantic segmentation [2, 34]. However, some recent works [9, 26] show that these deep models have the tendency to provide overconfident predictions. This greatly limits the overall trust in their predictions, especially when they are part of the decision-making system in safety-critical applications [6, 8, 32]. For instance, a decision system in an AI-powered healthcare diagnostic application can safely reject predictions with low confidence, however, if it mistakenly skips reviewing an incorrect prediction with high confidence, it can lead to serious consequences.

An important underlying reason behind the miscalibration of DNNs is training with zero-entropy supervision signal which makes them overconfident, and thus inadvertently miscalibrated. There have been few attempts towards improving the model calibration. A prominent technique is based on a post-processing step that transforms the outputs of a trained model with parameter(s) learned on a held-out validation set [9, 14, 15, 19, 28]. Although simple to implement, these methods are architecture and data-dependent [25], and further requires a separate held-out validation set which is not readily available in many real-world applications. An alternative approach is a train-time calibration method which tends to involve all model parameters during training. Existing train-time calibration methods [11,18,22,25] propose an auxiliary loss term that can be used in conjunction with an application-specific loss function (e.g., Cross Entropy or Focal loss [26]). Recently, [11] propose a differentiable auxiliary loss formulation to calibrate the class confidence of both the predicted label along with non-predicted labels.

Almost all work towards improving model calibration target the task of classification [9, 11, 20, 22, 25]. However, the calibration of object detection models has not been actively explored. Similar to classification models, object detection models also occupy an important position in many safety-critical applications. For instance, they form an integral part of the perception component of self-driving vehicles. Furthermore, the majority of efforts tackling model calibration focus on calibrating in-domain predictions. A deployed deep learning-based model can encounter samples from a distribution that is radically different from the training distribution. Therefore, a real-world model should be well-calibrated for both in-domain and out-domain predictions. So, in essence, well-calibrated object detectors, particularly under distribution shifts, not only contribute to algorithmic advancements but are of great importance to many vision-based safety-critical applications.

In this paper, we study the calibration of object detection models for both in-domain and out-domain predictions. We observe that the recent state-of-the-art object detectors are rather miscalibrated when compared to their predictive accuracy (Fig. 1). To this end, inspired by the train-time calibration approaches [11, 18, 25], we propose a novel train-time auxiliary loss formulation (Fig. 2), which explicitly attempts to bridge the model's precision with the predicted class confidence (BPC). It leverages the count of true positives and false positives in a minibatch, which are then employed to construct a penalty for miscalibrated predictions. We develop a differentiable proxy to the actual loss formulation that is based on counts. Our loss function is designed to be used with other application-specific loss functions. We perform extensive experiments on both in-domain and out-domain scenarios, including the large-scale MS-COCO benchmark. Results reveal that our train-time auxiliary loss is capable of significantly improving the calibration of a state-of-the-art vision-transformer based object detector under both in-domain and out-domain scenarios.

2. Related Work

Most of the work for calibrating DNNs can be categorized as: post-hoc and train-time methods. Post-hoc methods require hold-out validation set and involve a few parameters, whereas train-time methods do not require validation data and involve all model parameters. We briefly discuss these methods and other works below.

Post-hoc methods: A simple and classic approach to improving model calibration is temperature scaling (TS) [9], which is an extension of Platt scaling [28] from binary to multi-class settings. TS uses a parameter to modulate the logits of a trained model, whereby this parameter is estimated using hold-out data. This lowers the predicted confidence to achieve calibration. A more general form of TS is matrix scaling for the transformation of logits. This matrix is learned in a similar way using hold-out validation set. Besides involving limited parameters, the majority of post-hoc methods are limited to calibrating in-domain predictions [27]. Further, these post-hoc calibration methods are prone to performing poorly for dense prediction tasks [11]. To improve post-hoc calibration under out-domain scenarios, [37] transforms the validation set prior to performing the post-hoc approach. In [4], a regression model is used to predict temperature parameter. Post-hoc calibration methods are simple and effective, however, they require hold-out validation data, and are dependent on architecture [25].

Train-time calibration methods: Models trained with zero-entropy supervision tend to give over-confident predictions. An example is negative log-likelihood (NLL), which is a widely-used task-specific loss. A model trained with NLL provides predictions that deviate from the accuracy, leaving the model poorly calibrated [9]. Train-time calibration methods are typically based on auxiliary loss functions, which are used in-tandem with task-specific losses. In [22], an auxiliary loss term DCA is proposed to calibrate the model. It is combined with a task-specific loss to penalize when it reduces but the accuracy remains unchanged. Likewise, [20] proposed an auxiliary loss function that is based on a reproducing kernel in a Hilbert space [7]. [18] calibrated uncertainty based on the relationship between accuracy and uncertainty. Recently in [11], proposed a loss known as the multi-class difference of confidence and accuracy which aims to calibrate the predicted confidence of all classes. Building on the label smoothing (LS) work [35], [25], introduced a margin constraint logit distances to achieve implicit model calibration.

Other methods: Model calibration with OOD detection in [12] suggested that the ReLU activation function causes the model to provide overconfident predictions for input samples that lie away from the training samples. To circumvent this, a model is forced to output low scores for samples distant from training data by leveraging data augmentation using adversarial training. In [17], OOD inputs are detected with spectral analysis over early layers in convolutional neural networks (CNNs), thereby achieving model calibration.

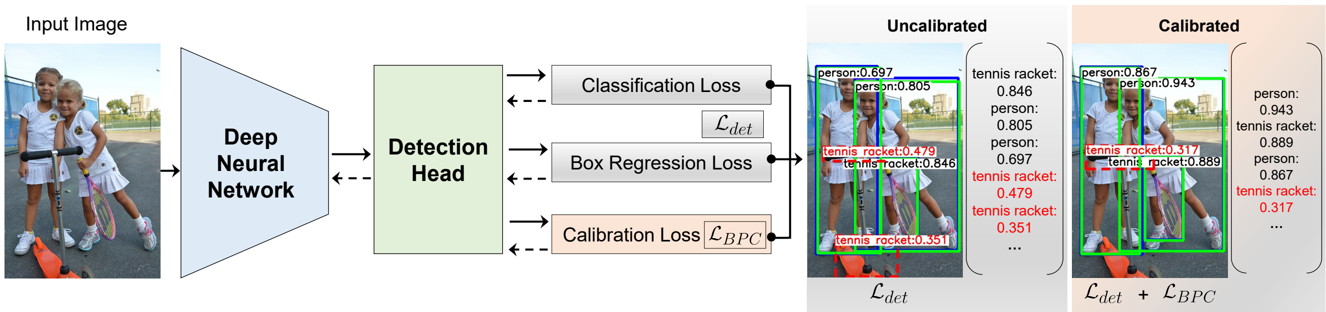

Figure 2. Main architecture: Our BPC loss function is integrated with object detection architecture and the detector predicts well-calibrated probabilities for accurate predictions (shown in Green), while lowering the probabilities of inaccurate prediction (shown in dashed Red). On the other hand, an uncalibrated model predicts scores, lower for accurate predictions and higher for inaccurate predictions. Blue color shows the ground truth boxes present for corresponding detections. Best viewed in color.

All post-hoc and train-time losses target the calibration of classification models, and there is almost no attention given to the calibration of object detection models. To this end, we explore the space of calibrating modern DNN-based object detectors. We propose a new train-time calibration method based on a new auxiliary loss function (BPC). It is differentiable, operates over mini-batches, and effectively calibrates modern object detectors for in-domain and out-domain detections.

3. Method

3.1. What is calibration?

A model is well-calibrated when the predicted confidence is aligned with the likelihood of the sample being correct. For example, a prediction of a calibrated model with confidence s aligns with the occurrence of a sample with the same s. A model is overconfident when it satisfies the condition of correctness with $< s\%$, and underconfident when $> s\%$. Many recent works addressing model calibration target the task of classification. In the following, we briefly define calibration for classification and object detection.

Classification: Given a dataset $\mathcal{D}$ defined with the joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that N number of images belonging to C ground truth classes are available. Let $\mathcal{D} = \{(\mathbf{x}_n, y_n)_{n=1}^{N}\}$, where $\mathbf{x}_n \in \mathcal{X} \in \mathbb{R}^{H\times W\times d}$ (an input image with height H, width W, and number of channels d). For each image, we have a corresponding ground truth class label $y_n \in \mathcal{Y} = \{1, 2, ..., C\}$. Let $\mathcal{G}_{cls}$ be a classification model that predicts a label $\bar{y}$ with confidence score $\bar{s}$. Following [9], we define a perfect classification calibration as: $\mathbb{P}(\bar{y} = y|\bar{s} = s) = s$, s.t., $s \in [0, 1]$. According to this expression, accuracy and confidence must align for all confidence levels.

Object Detection: For object detection, the localization of an object is an integral component along with the class label. Therefore, bounding box ($\mathbf{b}$) annotations are also available with corresponding class labels. Specifically, for each object, let $\mathbf{b} \in \mathbb{R}^4$ and y correspond to its class label. Given the object detector $\mathcal{G}_{det}$, model predicts a bounding box $\bar{\mathbf{b}}$ and label $\bar{y}$ with confidence score $\bar{s}$. Following [21], we can define the perfect calibration in object detection as: $\mathbb{P}(K = 1|\bar{s} = s) = s, \ \forall s \in [0, 1]$ [1]. Where $K = 1$ denotes an accurate detection in which both the class prediction matches with the ground truth class and the IoU (between the predicted and the ground truth box) is greater than a certain threshold i.e. $\mathbb{1}[IoU(\bar{\mathbf{b}}, \mathbf{b}) \geq \rho]\mathbb{1}[\bar{y} = y]$.

3.2. Measuring Calibration

Classification: Expected calibration error (ECE) is a widely used metric to quantify the miscalibration of a classification model. It measures the expected deviation of accuracy from the confidence for all confidence levels [9].

$$\mathbb{E}_{\bar{s}}\Big[|\mathbb{P}(\bar{y} = y|\bar{s} = s) - s|\Big] \tag{1}$$

As the confidence score is a continuous random variable, the confidence levels are divided into L equally-spaced bins. The approximation of ECE is computed as:

$$\mathbf{ECE} = \sum_{l=1}^{L} \frac{|B(l)|}{|\mathcal{D}|} \left|\mathrm{acc}(l) - \mathrm{conf}(l)\right| \tag{2}$$

where $|\mathcal{D}|$ is the total number of samples, $B(l)$ is the set of samples in the l^{th} bin. Further, $\mathrm{acc}(l)$ and $\mathrm{conf}(l)$ denote the average accuracy and average confidence over samples in the l^{th} bin, respectively.

Object detection: Similar to ECE for classification, we can define the detection expected calibration error (D-ECE) as the expected deviation of precision from the confidence for all confidence levels [21]:

$$\mathbb{E}_{\bar{s}}\Big[|\mathbb{P}(K = 1|\bar{s} = s) - s|\Big] \tag{3}$$

[1]Note that, this definition of calibration for object detectors is extendable to include box properties.

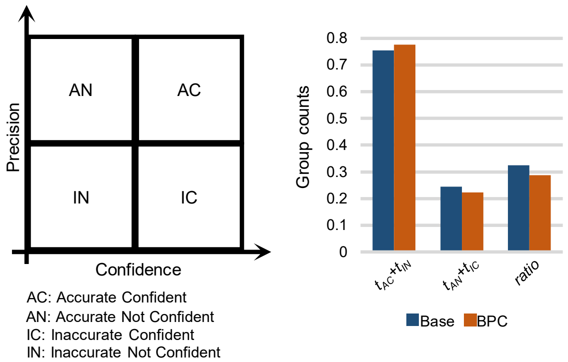

Figure 3. Left: We divide the confidence and precision space into four groups for categorizing accurate and inaccurate predictions over the minibatch according to their predicting confidence. Right: Compared to baseline, our BPC loss function has increased the joint count of accurate/confident and inaccurate/not-confident detections, while at the same time, it has decreased the joint count of accurate/not-confident and inaccurate/confident detections. Further, when compared to the baseline, the ratio between the latter and the former (defined in Eq. (9)) is smaller.

As confidence is a continuous variable, similar to eq.(2), we can approximate D-ECE:

$$\textbf{D-ECE} = \sum_{l=1}^{L} \frac{|B(l)|}{|\mathcal{D}|} \left|\text{prec}(l) - \text{conf}(l)\right| \tag{4}$$

where $\text{prec}(l)$ denotes the precision in l^{th} bin. Different from Eq. (2), here, $B(l)$ is the set of object instances in l^{th} bin and $|\mathcal{D}|$ is the total number of object instances.

3.3. BPC: Train-time Calibration Loss for Detection

Motivation: DNNs-based object detectors are trained with the objective to predict with high confidence, leaving them miscalibrated for both in-domain and out-of-domain detections. The rationale behind this behavior is the lack of direct supervision for the model to promote higher confidence for accurate predictions and lower confidence for inaccurate predictions. Motivated by this observation, we leverage the statistics associated with high-scoring and low-scoring box predictions to calibrate the detection model. We utilize the true positives and false positives to span the precision and confidence space in order to maximize the probability scores for accurate predictions and minimize the same for inaccurate predictions. Specifically, we discretize the confidence and precision space into four partitions for categorizing the accurate and inaccurate detections (Fig. 3). This lets us develop a simple auxiliary objective, which attempts to distribute the detections according to their respective confidences.

Formulation: We propose a train-time method for calibrating object detectors, at the core of which is a simple auxiliary loss function. It is differentiable, operates on minibatches, and is formulated to be used with other task-specific detection losses.

Inspired by the train-time calibration loss for classification [18], we formulate a loss function specific to object detection. We divide the confidence and precision space into four partitions and categorize the true positive (TP) and false positive (FP) detections over a minibatch. The four partitions for TP and FP are: **(1)** accurate and confident (AC) **(2)** accurate and not confident (AN) **(3)** inaccurate and confident (IC) and **(4)** inaccurate and not confident (IN). Let t_{AC}, t_{AN}, t_{IC} and t_{IN} represent the number of detections in AC, AN, IC, and IN, respectively. In principle, we need accurate detections to be more confident and inaccurate ones to be less confident, so we define the following objective that should be maximized:

$$\text{PC} = \frac{\mathrm{t}_{AC} + \mathrm{t}_{IN}}{\mathrm{t}_{AC} + \mathrm{t}_{IN} + \mathrm{t}_{AN} + \mathrm{t}_{IC}} \tag{5}$$

In object detection, the obtained predictions are either accurate or inaccurate. Given the predicted class label, bounding boxes, $\mathbb{1}$ as an indicator function, and th is the threshold on score, we define the following:

$$\mathrm{t}_{AC} = \sum_{i} \mathbb{1}[IoU(\bar{\mathbf{b}}_i, \mathbf{b}_i) \geq \rho]\mathbb{1}[\bar{y}_i = y_i] \ \mid \ \bar{s}_i \geq th \tag{6}$$

$$\mathrm{t}_{AN} = \sum_{i} \mathbb{1}[IoU(\bar{\mathbf{b}}_i, \mathbf{b}_i) \geq \rho]\mathbb{1}[\bar{y}_i = y_i] \ \mid \ \bar{s}_i < th \tag{7}$$

t_{IC} & t_{IN}: The remaining detections after populating t_{AC} and t_{AN} are false positives (inaccurate). Similar to Eq. (6) and Eq. (7), we categorize them based on their confidence scores.

In our loss formulation, we consider precision since it includes true positives and false positives, for which we have confidence scores. Whereas false negatives cannot be considered as they do not have confidence scores because of no detections. Since Eq. (5) is not differentiable owing to the indicator functions for t_{AC}, t_{AN}, t_{IC} and t_{IN}, we formulate its differentiable version to approximate these quantities. Let t_{AC}, t_{AN}, t_{IC} and t_{IN} be the approximations to t_{AC}, t_{AN}, t_{IC} and t_{IN}, respectively. We express the differentiable formulation as:

$$t_{AC} = \sum_{i \in \left(\substack{K_i = 1 \ \& \\ \bar{s}_i \geq th}\right)} \bar{s}_i \odot \tanh(\bar{s}_i)$$

$$t_{AN} = \sum_{i \in \left(\substack{K_i = 1 \ \& \\ \bar{s}_i < th}\right)} \bar{s}_i \odot (1 - \tanh(\bar{s}_i))$$

$$t_{IC} = \sum_{i \in \left(\substack{K_i=0\ \& \\ \bar{s}_i \geq th}\right)} (1 - \bar{s}_i) \odot \tanh(\bar{s}_i)$$

$$t_{IN} = \sum_{i \in \left(\substack{K_i=0\ \& \\ \bar{s}_i < th}\right)} (1 - \bar{s}_i) \odot (1 - \tanh(\bar{s}_i))$$

This is based on the rationale that when a detection is accurate, the confidence score satisfies to $\bar{s} \rightarrow 1$, and otherwise $\bar{s} \rightarrow 0$. Where $\tanh$ denotes the hyperbolic tangent function that modulates the penalization to the confidence score. The $\tanh$ function tapers off the confidence values in cases where the prediction is accurate so emphasize less on easy cases (well-calibrated) and focus more on the hard cases (not well-calibrated). Now we define the differentiable surrogate approximation to our auxiliary loss function:

$$\mathcal{L}_{BPC} = -\log\left(\frac{t_{AC} + t_{IN}}{t_{AC} + t_{IN} + t_{AN} + t_{IC}}\right) \tag{8}$$

This above relation Eq.(8) can be simplified as to minimize the following:

$$\mathcal{L}_{BPC} = \log\left(1 + \frac{t_{AN} + t_{IC}}{t_{AC} + t_{IN}}\right) \tag{9}$$

We note that, the calibration loss $\mathcal{L}_{BPC}$ is model-agnostic and differentiable. and can be integrated with task specific losses of modern object detection methods. Let $\mathcal{L}_{det}$ be the object detection loss that contains classification (*e.g.* Focal Loss) and localization (*e.g.* Generalized IoU & L1) losses, we can add our $\mathcal{L}_{BPC}$ to it and obtain the total loss as:

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \mathcal{L}_{BPC} \tag{10}$$

Since accurate predictions should have high confidence scores, our train-time loss attempts to align higher accuracy with higher confidence scores and vice versa. As shown in Fig. 3, compared to baseline, it increases the joint count of accurate/confident and inaccurate/not-confident detections, while at the same time, it decreases the joint count of accurate/not-confident and inaccurate/confident detections. Further, when compared to the baseline, the ratio between the latter and the former (defined in Eq. (9)) is decreased.

4. Experiments & Results

Datasets: For both in-domain and out-domain scenarios, we perform experiments on various object detection datasets, including large-scale ones. **MS-COCO** [24] contains 118K images for training as train2017, 41K as test2017, and 5K images as val2017, that are used for evaluation. It consists of 80 object categories in real world images. **CorCOCO** is a corrupted version of MS-COCO val2017 dataset for evaluations in out-domain scenarios. It incorporates random corruptions out of specified settings in [13], with arbitrary severity levels. **Cityscapes** [3] is an urban driving scene dataset consisting of 8 categories: *person, rider, car, truck, bus, train, motorbike, and bicycle*. It contains 2975 training images and 500 validation images used for evaluation. **Foggy Cityscapes** [30] consists of images simulating foggy weather on Cityscapes, and its validation set with severe level of fog is used for evaluation for out-domain scenario. **Sim10k** [16] is a dataset of synthetic images containing car category. It contains 10K images from which we split 8K as training set and 1K is used for evaluation. **BDD100k** [38] consists of 70K training images, 20K test images and 10K validation images. We only consider daylight subset of validation set for the evaluation of out-domain scenario which counts to 5.2K images. This dataset contains class categories similar to Cityscapes.
Datasets (post-hoc): We use validation sets based on three in-domain scenarios for temperature scaling as a post-hoc method. We opt Object365 [31] validation dataset in case of MS-COCO with similar categories, subset of BDD100k train set for Cityscapes and for Sim10k, its validation split.
Implementation Details: We use a SoTA detector Deformable-DETR (D-DETR) as a baseline and integrate our loss function with it. D-DETR uses focal loss [23] for classification and generalized IOU & L1 losses [1] for localization. Default settings are used and more details can be seen in [39]. In addition to the comparison of our proposed train-time loss with post-hoc method [9], we also compare calibration performance with recent calibration losses, MDCA [11] and MbLS [25]. D-DETR is trained with respective train-time losses and in-domain datasets.
Evaluation: For both in-domain and out-domain, we report detection expected calibration error (D-ECE) [21] as object detection calibration measure along with mean average precision of detectors.

4.1. Results

We have performed extensive experiments with post-hoc method and recent train-time losses over various in-domain and out-domain scenarios. For post-hoc method, we need validation set and for temperature scaling (TS) we optimize calibration parameter T. We compare all of these methods with our proposed loss function, specifically designed for object detectors. Our results show significant improve-

Methods \ Scenarios	In-Domain (COCO)			Out-Domain (CorCOCO)		
	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5
Baseline [39]	12.8	44.0	62.9	10.8	23.9	35.8
TS (post-hoc) [9]	14.2	44.0	62.9	12.3	23.9	35.8
MDCA [11]	12.2	44.0	62.9	11.1	23.5	35.3
MbLS [25]	15.7	44.4	63.4	12.4	23.5	35.3
BPC (Ours)	10.3	43.7	62.8	9.4	23.2	34.9

Table 1. Calibration performance on COCO in-domain and out-domain scenarios. Results show that our proposed BPC improves calibration of object detection as compared to baseline, other train time losses and post-hoc methods. AP box and mAP@0.5 are also reported in the table.

Methods \ Scenarios	In-Domain (Cityscapes)			Out-Domain (Foggy Cityscapes)			Out-Domain (BDD100k)		
	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5
Baseline [39]	13.8	26.8	49.5	19.5	17.3	29.3	11.7	10.2	21.9
TS (post-hoc) [9]	12.6	26.8	49.5	14.6	17.3	29.3	24.5	10.2	21.9
MDCA [11]	13.4	27.5	49.5	17.1	17.7	30.3	14.2	10.7	22.7
MbLS [25]	12.1	27.3	49.7	20.0	17.1	29.1	11.6	10.5	22.7
BPC (Ours)	9.9	26.8	48.7	12.5	17.7	30.2	10.6	11.0	23.6

Table 2. Calibration results with baseline, train-time losses and post-hoc methods are reported. BPC shows improvement in detection calibration for all the scenarios of in-domain (Cityscapes) and out-domain (Foggy Cityscapes & BDD100k). AP box and mAP@0.5 are also reported for each scenario.

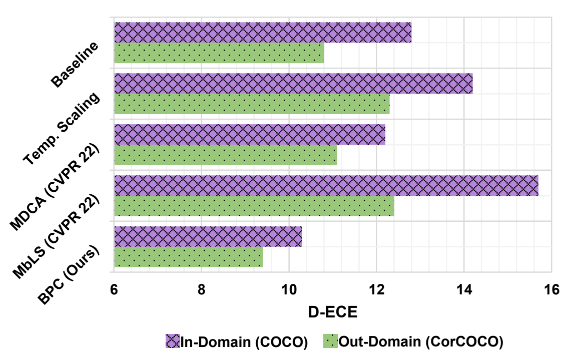

Figure 4. Our BPC loss reduces D-ECE for MS-COCO in-domain and Cor-COCO out-domain. We note that the classification calibration losses are sub-optimal for detection calibration and BPC loss improves calibration over post-hoc and train-time losses.

ment over calibration scores (**lower the better**), while having comparable performance in detection accuracy.

Real and Corrupted domains: To see the effectiveness of our loss, we perform experiments with a large-scale benchmark dataset, MS-COCO. We show our results for in-domain and out-domain scenarios, note that a corrupted version of the MS-COCO (CorCOCO) validation set is used for out-domain evaluation. In Tab. 1, the post-hoc based TS method fails to improve calibration in both domains, that usually considered to be a good performer in the in-domain. Results show train time losses that are designed for classification-based model calibration, are also not ideal for the calibration of object detectors. We report D-ECE, and our proposed loss function shows improvement in calibration scores for both in-domain (→COCO, 2.5%↓) and out-domain (→CorCOCO, 1.4%↓) scenarios from the baseline (Fig. 4). In comparison with MbLS, our loss improves calibration scores of 5.4%↓ and 3.0%↓ for in-domain and out-domain respectively.

Weather domains: We consider weather shift scenario for evaluation in both domains. For in-domain Cityscapes (CS) and out-domain Foggy CS, we see in Tab. 2 our loss shows improvement over post-hoc, for both in-domain (→CS, 2.7%↓) and out-domain (→Foggy CS, 2.1%↓). Also, we show improvement as compared to train-time losses, notably (→Foggy CS, 7.5%↓) over MbLS.

Scene domains: To have CS as in-domain in scene shift, BDD100k is evaluated as an out-domain scenario. Both belong to urban driving scenes but there is a large scene deviation among them. We show results in this scenario in Tab. 2 and find that TS performs the worst, followed by classification-based train time losses (Fig. 5). We outperform TS in out-domain (→BDD100k, 13.9%↓) and the recent MDCA (→BDD100k, 3.6%↓) approach.

Synthetic and Real domains: Sim10k is a synthetic dataset and considered as in-domain, while BDD100k as a daylight subset is considered as out-domain. We extract the car category from the BDD100k evaluation set and report the results. Our loss shows improved calibration scores for in-domain (→Sim10k, 3.9%↓) and out-domain (→BDD100k, 2.5%↓) scenarios over the MDCA loss (Fig. 6). Also we show calibration improvement of 16.4%↓ and 10.5%↓ over MbLS for in-domain and out-domain respectively (Tab. 3).

Qualitative Figures: We show qualitative detection results in Fig. 7. Detector trained with our loss forces the accurate predictions to be more confident whereas inaccurate predictions to be less confident.

Reliability Diagrams: We show reliability diagrams to see

Methods \ Scenarios	InDomain (Sim10k)			OutDomain (BDD100k)		
	D-ECE ↓	AP box	mAP@0.5	D-ECE ↓	AP box	mAP@0.5
Baseline [39]	10.3	65.9	90.7	7.3	23.5	46.6
TS (post-hoc) [9]	15.7	65.9	90.7	10.5	23.5	46.6
MDCA [11]	10.0	64.8	90.3	8.8	22.7	45.7
MbLS [25]	22.5	63.8	90.5	16.8	23.4	47.4
BPC (Ours)	6.1	65.4	90.5	6.3	23.4	45.6

Table 3. Calibration performance with our proposed BPC loss is improved over baseline, train-time losses and post-hoc methods for both in-domain (Sim10k) and out-domain (BDD100k). Car class is considered in this scenario for evaluations. AP box and mAP@0.5 are also reported.

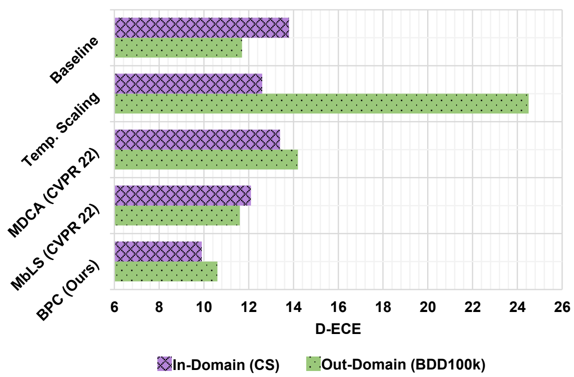

Figure 5. After integrating our BPC loss the calibration performance is improved in Cityscapes (CS) and CS to BDD100k.

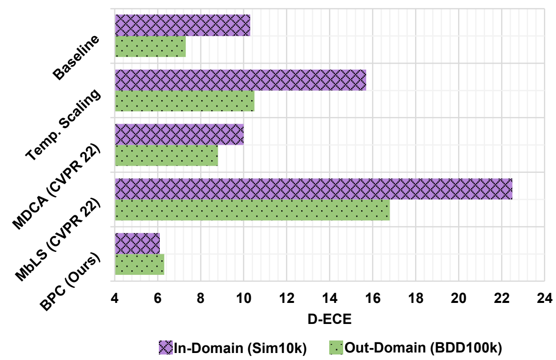

Figure 6. Our BPC loss reduces D-ECE in Sim10k and BDD100k as in-domain and out-domain scenarios, respectively.

Method	In-Domain		
	D-ECE ↓	AP box	mAP@0.5
BPC (th=0.4)	9.7	50.2	80.5
BPC (th=0.5)	9.1	50.1	80.2
BPC (th=0.6)	11.2	50.9	81.4

Table 4. Impact of probability thresholds on BPC loss. We perform experiments using train and test subsets of Sim10k train set for ablation study.

Method	In-Domain		
	D-ECE ↓	AP box	mAP@0.5
BPC (BS=1)	10.5	50.3	79.6
BPC (BS=2)	9.1	50.1	80.2
BPC (BS=3)	8.9	48.6	78.7
BPC (BS=4)	10.2	47.3	78.3

Table 5. Impact of batch sizes on BPC loss. We observe little degradation in detection accuracy by varying batch size (BS) and observe calibration performance is not much sensitive.

the behaviour of calibration in Fig. 8. A perfect calibration is achieved if the confidence is exactly same as precision.

4.2. Ablation & Analysis

We perform ablation studies on score threshold, batch sizes and random initialization. For this purpose, we select the subsets of Sim10k training set as train and validation to empirically find score threshold hyper-parameter. With similar data splits, we show impact of batch sizes and random weight initialization on our loss.

Score Threshold: We study the impact of score threshold that is used for penalizing the probabilities of instances present in the batch. Varying the score threshold shows some degradation in detection performance for in-domain but calibration still stands out the best and we find our approach is not much sensitive to it. We empirically find in Tab. 4 that $th = 0.5$ improves calibration.

Batch Size: We observe in Tab. 5 the impact of batch sizes on our proposed loss function. We see that increasing batch size has little effect on the detection accuracy and calibration performance is not sensitive for given scenario. To get the best for both metrics and without sacrificing the drop in detection performance, we opt for batch size 2 for all experiments.

Random Weight Initialization: Impact of different seeds with calibration loss is studied by setting different initialization points for experiments. This shows that calibration is not much influenced by random initialization (Tab. 6). We

Method	In-Domain		
	D-ECE ↓	AP box	mAP@0.5
BPC (seed=30)	9.0	49.2	79.7
BPC (seed=42)	9.1	50.1	80.2
BPC (seed=60)	8.6	51.0	80.5

Table 6. Impact of different seeds on BPC loss. We observe changing seeds for initialization has little effect on calibration performance.

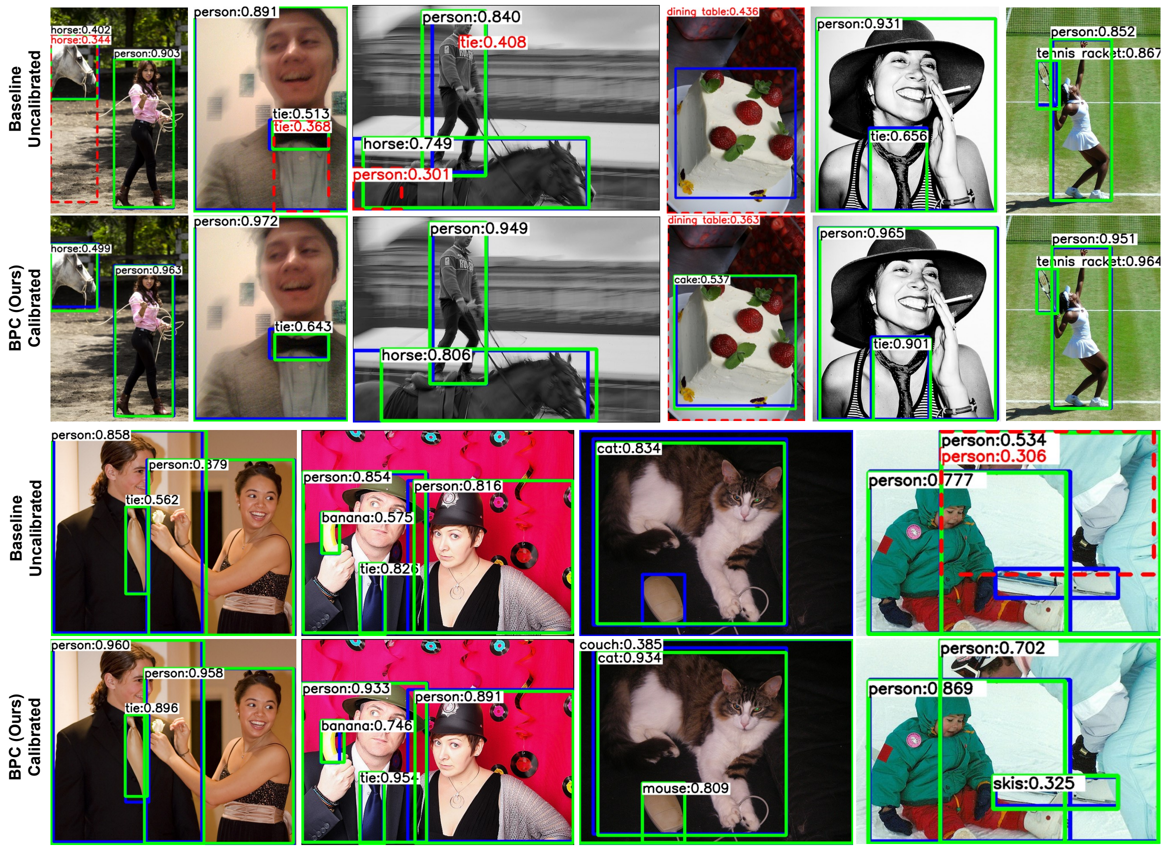

Figure 7. Baseline [39] vs. BPC (Ours): Qualitative results on MS-COCO dataset (In-Domain). Detector trained with our loss forces the accurate predictions to be more confident whereas inaccurate predictions to be less confident. Detection threshold is set to 0.3. Green boxes are accurate predictions with respective confidence scores. Red (dashed) boxes are inaccurate predictions with corresponding scores. Blue shows the ground truth boxes present for corresponding detections.

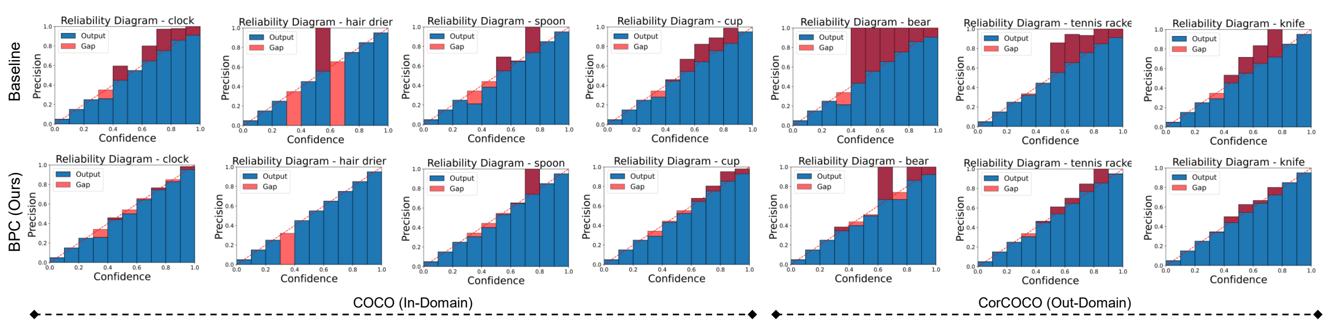

Figure 8. Reliability Diagrams: MS-COCO (In-Domain) and CorCOCO (Out-Domain). Top: Baseline [39] trained as D-DETR and Bottom: D-DETR trained with our proposed BPC loss.

set seed 42 as default in our experiments.

5. Conclusion

In this paper, we presented a new train-time calibration method for object detection which is based on an auxiliary loss function (BPC). It utilizes true positive and false positive statistics to maximize the confidence scores for accurate predictions and minimize scores for inaccurate predictions. We perform extensive experiments on several in-domain and out-domain scenarios, including large-scale detection dataset, to show effectiveness of our loss function for calibrating object detectors. Results show that our method outperforms several train-time calibration methods in terms of improving calibration of both in-domain and out-domain predictions while also preserving the detection accuracy.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV*, 2020. 5

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 1

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[4] Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. Local temperature scaling for probability calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6889–6899, 2021. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1

[6] Michael W Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M Dai. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 204–213, 2020. 1

[7] Arthur Gretton. Introduction to rkhs, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 16:5–3, 2013. 2

[8] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. 1

[9] 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, 3, 5, 6, 7

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[11] 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, 5, 6, 7

[12] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019. 2

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019. 5

[14] Mobarakol Islam, Lalithkumar Seenivasan, Hongliang Ren, and Ben Glocker. Class-distribution-aware calibration for long-tailed visual recognition. *arXiv preprint arXiv:2109.05263*, 2021. 1

[15] Byeongmoon Ji, Hyemin Jung, Jihyeun Yoon, Kyungyul Kim, and Younghak Shin. Bin-wise temperature scaling (bts): Improvement in confidence calibration performance through simple scaling techniques. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4190–4196, 2019. 1

[16] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 746–753. IEEE, 2017. 5

[17] Davood Karimi and Ali Gholipour. Improving calibration and out-of-distribution detection in deep models for medical image segmentation. *IEEE Transactions on Artificial Intelligence*, 2022. 2

[18] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 2020. 2, 4

[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. PMLR, 2017. 1

[20] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR, 2018. 2

[21] Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 326–327, 2020. 3, 5

[22] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. In *British Machine Vision Conference (BMVC)*, 2020. 2

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[25] 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, 6, 7

[26] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020. 1, 2

[27] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 2

[28] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 1, 2

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

[30] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 5

[31] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8429–8438, 2019. 5

[32] Monika Sharma, Oindrila Saha, Anand Sriraman, Ramya Hebbalaguppe, Lovekesh Vig, and Shirish Karande. Crowdsourcing for chromosome segmentation and deep classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 34–41, 2017. 1

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[34] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7242–7252, 2021. 1

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[36] 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. 1

[37] Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10132, 2021. 2

[38] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 5

[39] 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, 5, 6, 7, 8