Towards Improving Calibration in Object Detection Under Domain Shift

Anonymous Author(s)
Affiliation
Address
email

Abstract

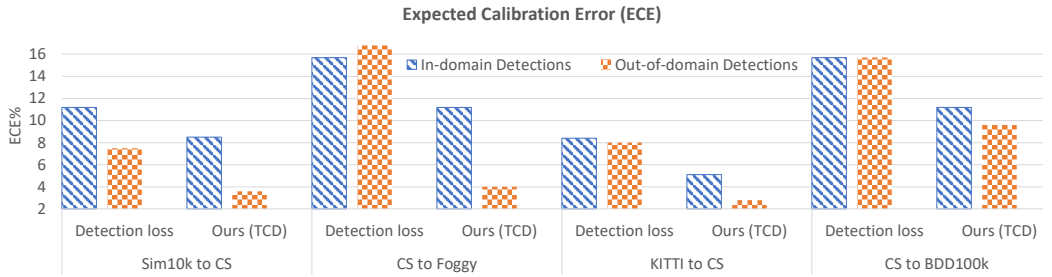
The increasing use of deep neural networks in safety-critical applications requires the trained models to be well-calibrated. Most current calibration techniques address classification problems while focusing on improving calibration on in-domain predictions. Little to no attention is paid towards addressing calibration of visual object detectors which occupy similar space and importance in many decision making systems. In this paper, we study the calibration of current object detection models, particularly under domain shift. To this end, we first introduce a plug-and-play train-time calibration loss for object detection. It can be used as an auxiliary loss function to improve detector's calibration. Second, we devise a new uncertainty quantification mechanism for object detection which can implicitly calibrate the commonly used self-training based domain adaptive detectors. We include in our study both single-stage and two-stage object detectors. We demonstrate that our loss improves calibration for both in-domain and out-of-domain detections with notable margins. Finally, we show the utility of our techniques in calibrating the domain adaptive object detectors in diverse domain shift scenarios.

1 Introduction

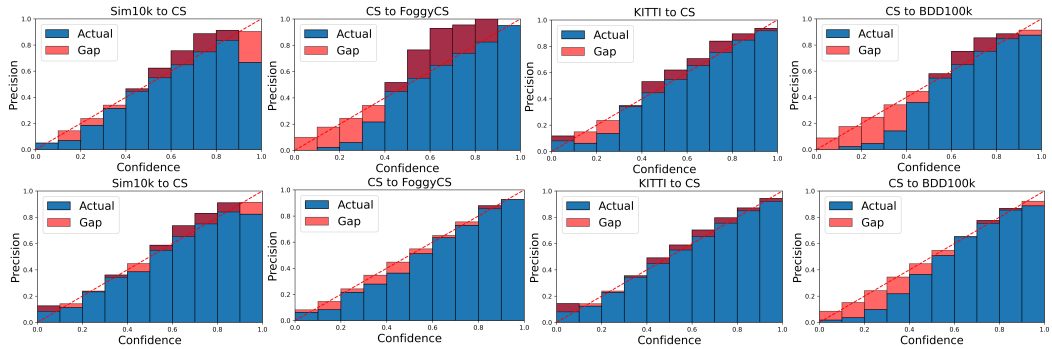
Owing to their high predictive power, deep neural networks (DNNs), in particular vision models, are increasingly deployed in different safety-critical applications, including autonomous driving [8], healthcare [5, 35], and legal research [41]. In such high-stake applications, it is of paramount importance that the model predictions are not only correct, but also well-calibrated. A model is said to be calibrated if the confidence of predictions aligns with the probability of being correct for all confidence levels [9]. For example, in autonomous driving application, we would like to defer predictions for which the model outputs low-confidence. However, skipping review due to confident, but incorrect, predictions, could lead to disastrous consequences.

We have witnessed tremendous effort towards pushing the predictive accuracy of models, however, considerably less attention is devoted to model calibration. Among a few works addressing model calibration, the majority of them focus on classification tasks [9, 19, 21, 23, 37]. A dominant class of methods address calibration by proposing different post-hoc approaches [31, 9]. In these methods, a validation set is leveraged to transform the predictions returned by a trained model such that the in-domain predictions are well-calibrated. Such methods involve limited parameters while calibrating the outputs of the models. Further, they are restrictive, since in many real-world scenarios, a validation set is not always available. To involve all model parameters, some methods propose train-time calibration techniques [19, 21, 23]. Typically, these methods only calibrate the label with maximum confidence for each image. In a multi-class scenario, this can lead to uncalibrated confidence scores for labels with non-maximum confidences.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Calibration performance (ECE) of a DNN-based detector (FCOS [36]) trained using task-specific detection loss and our method (task-specific detection loss+TCD), where TCD is the proposed auxiliary loss.



(b) Reliability diagrams. Top row: DNN-based detector (FCOS [36]) trained using task-specific loss. Bottom row: Ours, trained with adding the proposed TCD loss.

Figure 1: DNN-based object detectors trained with our proposed calibration loss results in better calibration.

Surprisingly, little to no attention has been paid towards addressing the calibration of modern visual object detection models, based on DNNs, that form an important part of many decision making systems. Also, most current efforts aim at improving model calibrations only for in-domain predictions. In many practical scenarios, once a model is deployed the distribution over observed data may shift and eventually becomes quite different from the original training data distribution. This requires that the model should be well-calibrated for not only in-domain predictions, but also for out-of-domain predictions. Besides carrying scientific value, well-calibrated object detectors, especially under domain drift, will substantially improve overall trust in many vision-based safety-critical applications and will be of great value to industry practitioners.

In this paper, we study the calibration of object detection models for both in-domain and out-of-domain detections. We observe that: (1) detection models demonstrate poor calibration for in-domain and out-of-domain detections (see Fig. 1), and (2) unsupervised domain adaptive detection models are rather miscalibrated when compared to their predictive accuracy in a target domain.

Towards developing well-calibrated object detection models, for both in-domain and out-of-domain scenarios, we propose a new plug-and-play loss formulation, termed as *train-time calibration for detection* (TCD). It can be used with task-specific loss functions during training phase and acts as a regularization for detections. In addition, we develop an implicit calibration technique for self-training based domain adaptive detectors. Finally, we empirically show that this technique is complementary to our loss function and they both can be utilized during adaptation to further boost calibration under challenging domain shift detection scenarios. We validate the effectiveness of our loss function towards improving calibration of different DNN-based object detection paradigms and different domain adaptive detection models under challenging domain shift scenarios.

2 Related Work

We discuss different techniques in literature for calibrating DNNs. They can be broadly categorized into either as post-hoc calibration or train-time calibration techniques. Post-hoc calibration method leverages a hold-out set to tune the calibration measures, whereas train-time calibration method

includes model calibration during the training phase. Further, there are other methods that achieve implicit calibration through using model's uncertainty or learning to reject OOD samples.

Post-hoc Calibration methods: In post-hoc calibration methods, a hold-out validation set is used to rescale the confidences produced by a trained DNN to achieve well-calibrated in-domain predictions. Temperature scaling (TS) [9], which is a variant of Platt scaling [31], divides the logits by a temperature parameter $T > 0$ to calibrate a trained DNN. The optimal value of T is learnt on a validation set. Since the ranking of logits remains unchanged in TS, it has the advantage of preserving the accuracy of the trained model. On the other hand, TS reduces the confidences of every prediction, including the correct one. A general version of TS transforms the output of a DNN using a matrix learnt on a hold-out set. Dirichlet calibration (DC) [17] is derived from Dirichlet distributions and generalizes the beta calibration method [18] from binary to multi-class one. It can be realized as an (extra) layer in a neural network on log-transformed class probabilities, which is learnt on a hold out validation data. Towards achieving meta-calibration, [1] proposed a differentiable metric for expected calibration error (ECE) and successfully uses it as an objective for meta-learning. Islam et al. [14] proposed class-distribution-aware TS and label smoothing (LS) by incorporating class frequency information in model calibration. Recently, [37] generalizes existing post-hoc calibration methods by transforming the validation set before performing the post-hoc calibration step to address poor calibration for out-of-domain predictions. In general, post-hoc calibration methods are not applicable without the availability of hold-out validation set. Also, majority of post-hoc methods asses model calibration for in-domain predictions.

Train-time Calibration methods: An initial attempt in the class of train-time methods is the Brier score for calibrating binary probabilistic forecast [2]. Guo et al. [9] revealed that the models trained with Negative Log-Likelihood (NLL) produce overconfident scores and therefore require re-calibration. To this end, various methods presented additional loss formulations in addition to task-specific losses during the training procedure. For instance, [30] leveraged entropy as a regularizer to penalize confident output distributions and [26] proposed label smoothing (LS) on soft-targets which helps in improving calibration. Recently, [25] showed that focal loss [22] minimizes regularized KL divergence between the predicted and target distributions which ensures minimisation of the KL divergence whilst increasing the entropy of the predicted distribution, thereby preventing the model to become overconfident. Liang et al. [21] formulated an auxiliary loss term for calibration based on ECE. Similarly, [19] proposed an auxiliary loss term for calibration which is computed using a reproducing kernel in a Hilbert space. Recently, [10] developed an auxiliary loss formulation for calibrating non-predicted labels along with the predicted one.

Other relevant methods: DNNs can be calibrated by learning to reject OOD samples. Hein et al. [11] identified ReLU activation is a core reason behind the overconfident predictions of DNNs far away from the training data. To circumvent this, they proposed a data augmentation using adversarial learning. Karimi et al. [16] leveraged spectral analysis on initial layers of a convolutional neural network (CNN) to discover OOD sample and calibrate the DNN. See [4, 12, 24, 29] for other representative works on improving calibration through OOD detection. Finally, some works report implicitly calibrating DNNs via some loss function [26] or selecting pseudo-labels during self-training based on model's uncertainty [27].

Note that, almost all aforementioned techniques for analyzing and improving DNN calibration focus on classification tasks. There is little to no literature studying the calibration of DNN-based object detectors, including under domain shift. Toward this aim, inspired by the train-time calibration methods, we propose a new auxiliary loss formulation (TCD) for DNN-based object detection calibration that can be integrated with any task-specific loss function while training. It is capable of significantly calibrating in-domain and out-of-domain object detections. In addition, we develop an implicit calibration technique, that is also complementary to our loss formulation, to boost the calibration of self-training based domain adaptive object detectors.

3 Improving Calibration in Object Detectors

3.1 Definitions of Calibration

Calibration for classification: We first review definition of calibration for classification. Let $\mathcal{D} = \langle(\mathbf{x}_i, y_i^*)\rangle_{i=1}^N$ be a dataset of N images from a joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$, where $\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$ is an input image with height H, width W, and number of channels C. Each image has an associated

ground-truth class label $y_i^* \in \mathcal{Y} = 1, 2, ..., K$. Given, a classification model $\mathcal{F}_{cls}$ that produces a label $\hat{y}$ and a corresponding confidence score $\hat{s}$, the following expression defines the perfect calibration [9]:

$$\mathbb{P}(\hat{y} = y^* | \hat{s} = s) = s \quad \forall s \in [0, 1]. \tag{1}$$

Where $\mathbb{P}(\hat{y} = y^* | \hat{s} = s)$ is the accuracy for a particular confidence score s. For the model to be calibrated, this accuracy should match with the predicted confidence.

Calibration for object detection: In object detection, a dataset consists of a ground-truth bounding box annotation $\mathbf{b}^* \in \mathcal{B} = [0, 1]^4$ and a corresponding class label y^* for each object. Lets assume an object detector $\mathcal{F}_{det}$ produces a confidence score $\hat{s}$ for a predicted label $\hat{y}$ and the corresponding bounding box prediction $\hat{\mathbf{b}}$. Then for object detection perfect calibration in defined as as [20] [1]:

$$\mathbb{P}(U = 1 | \hat{s} = s) = s \quad \forall s \in [0, 1]. \tag{2}$$

Where $U = 1$ represents a correctly classified prediction that aligns with a ground-truth object with a certain IoU score i.e. $\mathbb{1}[\mathrm{IoU}(\hat{\mathbf{b}}, \mathbf{b}^*) \geq \gamma^2] \mathbb{1}[\hat{y} = y^*]$.

Expected calibration error for classification and object detection: In classification problems, the expected calibration error (ECE) is used to quantify the miscalibration of a model [28, 9]. This score denotes the expected deviation of the accuracy from the predicted confidence:

$$\mathbb{E}_{\hat{s}} \left[|\mathbb{P}(\hat{y} = y^* | \hat{s} = s) - s| \right] \tag{3}$$

As $\hat{s}$ is a continuous random variable, ECE is approximated by binning the confidence space of $\hat{s}$ into M equally spaced bins as:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|I(m)|}{|\mathcal{D}|} \left| \mathrm{acc}(m) - \mathrm{conf}(m) \right|, \tag{4}$$

where $I(m)$ is the set of samples in m^{th} bin, and $|\mathcal{D}|$ is the total number of samples. $\mathrm{acc}(m)$ and $\mathrm{conf}(m)$ denote the average accuracy and average confidence in m^{th} bin, respectively. Along similar lines, detection ECE (D-ECE) can be defined as the expected deviation of precision from the predicted confidence [20]:

$$\mathbb{E}_{\hat{s}} \left[|\mathbb{P}(U = 1 | \hat{s} = s) - s| \right] \tag{5}$$

Similarly, we approximate D-ECE by partitioning the confidence space into M equally spaced bins:

$$\mathrm{D-ECE} = \sum_{m=1}^{M} \frac{|I(m)|}{|\mathcal{D}|} \left| \mathrm{prec}(m) - \mathrm{conf}(m) \right|, \tag{6}$$

where $\mathrm{prec}(m)$ denotes the average precision in a bin.

3.2 Train-time Calibration for Detection: TCD

In this section, we propose a new train-time calibration technique for object detection based on a novel auxiliary loss function. DNN-based object detectors predict a bounding box and corresponding class confidences for a detected region, in order to achieve detection calibration, it is imperative to take into account both of them. Our core idea is to jointly calibrate the estimated (class-wise) confidences and predicted bounding boxes. To achieve this, we propose to compute two quantities over a mini-batch during training: (1) the deviation between the accuracy and the confidence, and (2) the deviation between the predicted bounding box overlap and the predicted class confidence. Specifically, there

[1] Note that, Eq.(2) can also be extended to location-dependent calibration.

[2] γ is a fixed overlap threshold set to 0.5.

are two components in our loss formulation. Inspired by the train-time calibration techniques for classification, we adapt the confidence calibration loss [10] and develop the first component $\mathrm{d_{cls}}$. It measures the absolute difference between the average confidence and average accuracy.

$$\mathrm{d_{cls}} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{1}{L\times R}\sum_{l=1}^{L}\sum_{r=1}^{R} s_{l,r}[k] - \frac{1}{L\times R}\sum_{l=1}^{L}\sum_{r=1}^{R} q_{l,r}[k]\right| \tag{7}$$

Where L is the number of samples in a mini-batch, and R is the number of locations in the output class confidences channel map. $q_{l,r}[k] = 1$ if label k is the ground-truth label for sample l in location r, and 0 otherwise. The second component $\mathrm{d_{det}}$ computes the mean of the absolute difference between the bounding box overlap (with the ground-truth/pseudo ground-truth) and the confidence of the predicted class over all the positive regions. Let N^{l}_{pos} denotes the number of positive regions in the sample l. The loss is defined as:

$$\mathrm{d_{det}} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{N^{l}_{pos}}\sum_{n=1}^{N^{l}_{pos}}\left|[\mathrm{IoU}(\hat{\mathbf{b}}_n, \mathbf{b}^{*}_n) - \hat{s}_n]\right| \tag{8}$$

$$\mathcal{L}_{\mathrm{TCD}} = \frac{1}{2}(\mathrm{d_{cls}} + \mathrm{d_{det}}) \tag{9}$$

The $\mathrm{d_{cls}}$ component in the proposed loss is capable of not only calibrating the confidence of the predicted class but also for the non-predicted classes. It penalizes the model, if for a given class k, the average confidence across mini-batch samples and possible output locations deviates from the average occurrence across mini-batch of this class. On the other hand, the $\mathrm{d_{det}}$ component penalizes the deviation between the IoU score (computed between the predicted and the ground-truth bounding box) and its corresponding predicted confidence for positive regions. It explicitly forces the detector to match the confidence of the predicted class with the tightness of the predicted bounding box over the detected object. Since above components are not based on binning, like Eqs.(4,6), as such, it avoids the non-differentibility issue

Note that, both loss components, $\mathrm{d_{det}}$ and $\mathrm{d_{cls}}$, operate over the mini-batch constructed during training and so the $\mathcal{L}_{\mathrm{TCD}}$ can be used as an auxiliary train-time calibration loss for calibrating object detectors, including domain-adaptive ones, in conjunction with various task-specific loss functions (secs. 4.1, 4.2). Further, we show that the proposed loss formulation can also be used with the pseudo-labels used in self-training based unsupervised domain-adaptive detection algorithms for improving calibration in target domain (sec.4.2). Finally, we show that our loss formulation is complementary to implicit calibration techniques and so can be deployed to further enhance calibration (sec.4.3).

3.3 Implicitly Calibrating Self-Training based Domain Adaptive Detectors

We propose a technique aimed at implicitly improving the calibration of self-training based domain adaptive detectors. Mostly, self-training based domain adaptive detectors involve constructing pseudo instance-level labels (termed pseudo-labels thereafter) corresponding to detections in a target domain. Based on these pseudo-labels, pseudo-targets are formed for computing (classification) loss during the adaptation phase. We observe that, these pseudo-targets are constructed as one-hot encoded channels, and so they struggle to reflect the predictive confidence or uncertainty of detections. As a result, the adaptation model fails to account for the noise in detections, mostly prevalent under domain shift, and could inadvertently learn to make overconfident predictions, hence negatively affecting calibration. To this end, we first present a new uncertainty quantification mechanism for object detection and then leverage this to modulate the one-hot encoded pseudo-targets.

Quantifying uncertainty in object detection: To quantify model's uncertainty for detections, we follow a three-step process. First, we apply Monte-Carlo dropout [6] (specifically, spatial dropout [38]) to the convolutional filters after the feature extraction layer [3]. Given an arbitrary image, we repeat N stochastic forward passes (inferences) using MC dropout. We define $\hat{\mathbf{z}}_{n,m} = (\hat{\mathbf{b}}_{n,m}, \hat{c}_{n,m})$ be the m_{th} detection in n_{th} inference, where $\hat{c}_{n,m}$ is the predicted class corresponding to the highest

[3] Assuming one-stage object detection model.

confidence $\hat{s}_{n,m}$ in the confidence vector $\mathbf{s}_{n,m}$, and $\hat{\mathbf{b}}_{n,m} \in \mathbb{R}^4$ is the corresponding predicted bounding box. Second, we identify and group the detections corresponding to $\hat{\mathbf{z}}_{n,m}$ across the inference space that have the same predicted class and an overlap with its bounding box higher than a certain threshold. Specifically, for each $\hat{\mathbf{z}}_{n,m}$, we create a set $\mathcal{A}_{n,m}$ by including all $\hat{\mathbf{z}}_{k,l}$, where $k \neq n$ and l is an arbitrary detection in k_{th} MC forward pass, such that $\hat{\mathbf{b}}_{n,m}$ has IoU with $\widehat{\mathbf{b}}_{k,l}$ greater than a specific threshold and $\hat{c}_{n,m} = \hat{c}_{k,l}$ [27].

$$\mathcal{A}_{n,m} = \{\forall_{k \neq n} \cup (\hat{\mathbf{b}}_{k,l}, \hat{c}_{k,l}), \mid \mathrm{IoU}(\hat{\mathbf{b}}_{n,m}, \hat{\mathbf{b}}_{k,l}) > \gamma \ , \ \hat{c}_{k,l} = \hat{c}_{n,m} \ \}. \tag{10}$$

Where γ is set to 0.5 throughout experiments. Finally, $\mathcal{A}_{n,m}$ is utilized to estimate the uncertainty in $\hat{\mathbf{z}}_{n,m}$. We aim to capture uncertainty in predicted confidences and localization. So, we first estimate the variance in predicted (class) confidences, center x, center y, and aspect ratio of the bounding boxes predictions in $\mathcal{A}_{n,m}$ and later aggregate them to construct a joint measure of uncertainty. Let $\{\psi\}_{j=1}^{J}$ and $\{\Psi\}_{j=1}^{J}$ be the vectors containing variances and means of predicted confidences, center-x, center-y, and aspect ratio of the predicted bounding boxes in $\mathcal{A}_{n,m}$, respectively. Let Ψ_{agg} be the combined mean, computed as $\Psi_{agg} = \frac{1}{J}\sum_{j=1}^{J} \Psi_j$. Then, the combined variance representing a single, joint measure of uncertainty is computed as:

$$u_{n,m} = \frac{1}{J}\sum_{j=1}^{J}[\psi_j + (\Psi_j - \Psi_{agg})^2]. \tag{11}$$

Uncertainty-guided soft pseudo-targets: We leverage (joint) uncertainty to modulate the one-hot encoded pseudo-targets, formed according to (selected) pseudo-labels, to account for the entropy in object detections under target domain. Let H_i^k be i_{th} location in the class k one-hot encoded channel map corresponding to a detected bounding box, with uncertainty u_i^k and (pseudo) class label k. The uncertainty-guided soft pseudo-target is constructed as:

$$\widehat{\mathrm{H}}_i^k = \begin{cases} \mathrm{H}_i^k.(1 - u_i^k) & \text{if } \bar{s}_i^k \geq \kappa_1 \\ \mathrm{H}_i^k.\bar{s}_i^k.(1 - u_i^k) & \text{if } \kappa_2 \leq \bar{s}_i^k < \kappa_1 \end{cases} \tag{12}$$

Where $\bar{s}_i^k = \frac{1}{T}\sum_j \hat{s}_i^j$ is the mean (class) confidence of the predicted bounding boxes in $\mathcal{A}_i$. κ_1 is threshold for ranking between highly confident and relatively less confident detections, and κ_2 is the confidence threshold below which there is no detection considered.

4 Experiments

Datasets: Cityscapes [3] dataset contains images of road and street scenes and offers 2975 and 500 examples for training and validation, respectively. It consists of following categories: *person, rider, car, truck, bus, train, motorbike, and bicycle*. **Foggy Cityscapes** [34] dataset is constructed using the Cityscapes dataset by simulating foggy weather utilizing depth maps provided in Cityscapes with three levels of foggy weather. **Sim10k** [15] dataset is a collection of synthesized images, offering 10K images and their corresponding bounding box annotations. **KITTI** [7] dataset is similar to Cityscapes and offers images of road scenes with wide view of area, except that KITTI images were captured with a different camera setup. Following prior works, we consider car class for experiments when adapting from KITTI or Sim10k. **BDD100k** [40] contains 100k images with bounding box annotations and class labels. Out of 100k images, 70k are in training set and 10k is in the validation set. Following [39], we make a subset of 36.7k images from the training set and 5.2k images from the validation set that has daylight conditions and used with common categories as in Cityscapes.

Evaluation and implementation details: We report calibration performance using detection expected calibration error (D-ECE) and also report test accuracy. Further, we also plot reliability diagrams for visualizing calibration. Our TCD loss is developed to be used with the task-specific loss of DNN-based object detectors, including the SOTA domain-adaptive ones. For instance, the task specific losses for single-stage detectors use Focal loss [22] and IoU loss. Similarly Cross-Entropy and Smooth L1 loss are used in training the two-stage detectors. Let $\mathcal{L}_D$ be the task-specific loss,

Methods	Sim10k		KITTI		Cityscapes	
	D-ECE ↓	AP@0.5 ↑	D-ECE ↓	AP@0.5 ↑	D-ECE ↓	AP@0.5 ↑
In-domain Detections						
Single-stage	11.2	79.2	8.4	94.1	15.7	44.9
Single-stage + post-hoc	20.7	78.7	20.3	94.1	23.0	44.3
Single-stage + TCD	8.5	83.4	5.1	94.7	11.0	48.3
Out-of-domain Detections						
	Sim10K→CS		**KITTI→CS**		**CS→CS-foggy**	**CS→BDD100K**
Single stage	7.5	37.6	8.0	37.4	16.8	20.4	15.7	19.5
Single-stage + post-hoc	17.4	38.1	21.6	37.5	20.3	20.3	22.3	19.3
Single-stage + TCD	3.6	42.4	2.8	40.3	4.0	22.4	9.6	22.0

Table 1: Calibration performance and test accuracy with single-stage detector (FCOS [36]) trained with its application specific losses. Calibration with post-hoc temperature scaling and with our proposed TCD loss.

then the total loss in our method is computed as: $\mathcal{L} = \mathcal{L}_D + \mathcal{L}_{TCD}$. For further implementation details, refer to the supplementary.

Considered object detectors: Most current state-of-the-art (SOTA) object detectors are built on either a single-stage or a two-stage backbone. To include a representative detector model in both these variants, we consider a state-of-the-art single-stage detector FCOS [36] and a commonly used two-stage detector Faster-RCNN [32]. Our choice is partially motivated by the fact that the same backbones are also used in the current SOTA domain-adaptive object detectors. For the domain adaptive detectors we include EPM [13] and also SSAL [27] which are among the best performing single-stage domain adaptive detectors built on FCOS as source model. For the two-stage domain adaptive variant we include SWDA [33] that is built on FasterRCNN.

4.1 Experiments with One-stage and Two-stage Detectors

In this setting, we train a DNN-based object detector (1) with its task-specific loss functions and (2) after adding our TCD auxiliary loss function. For comparison we also include a post-hoc calibration technique based on temperature scaling. The temperature parameter T is obtained using a hold-out validation set to perform temperature scaling at inference time in the target domain. We measure the performance and calibration errors in two settings. **In-domain Detection:** Here we measure the performance on the test set of the dataset used in training and **Out-of-domain Detection:** where we test it on an unseen target domain i.e., a different dataset.

Table 1 reports results with a one-stage detector (FCOS [36]) trained with its task-specific losses i.e. Focal loss and IoU loss. We report calibration errors (D-ECE) and performance of the model. We show the impact on calibration after adding and training with the proposed auxiliary loss TCD. For comparison we include post-hoc temperature scaling based calibration method. For in-domain detections, we see that our proposal, single-stage detector trained after adding TCD, achieves the best D-ECE score across all datasets. Furthermore, our proposal allows boosting the detection performance by notable margins compared to a single-stage detector in all datasets. For out-of-domain detections, we significantly improve the calibration performance over baselines in all domain shift scenarios. For instance, it decreases the D-ECE score by 12.8% and 6.18% compared to the single-stage detector in CS→CS-foggy and CS→BDD100K, respectively. At the same time, it provides notable gains in detection performance over single-stage detector e.g., a 4.8% gain (AP@0.5) in Sim10K→CS.

We report results with a two-stage detector Faster-RCNN [32], trained with a task-specific loss, taking a pre-trained two-stage detector and applying post-hoc temperature scaling, and a two-stage trained with task-specific loss and our TCD loss (Table 2). For both in-domain and out-of-domain detections, we observe that our proposal, two-stage detector trained after adding TCD, delivers the best D-ECE score across all datasets.

4.2 Experiments with Domain-adaptive Detectors

In this setting, we incorporate our TCD auxiliary loss in the recent state-of-the-art domain-adaptive detectors and observe impact on calibration performance. In single-stage paradigm, we chose EPM [13], which accounts for pixel-wise centerness and objectness for target domain images in an adversarial alignment framework. Further, we also choose a method from the line of self-training based domain-adaptive detectors. Particularly, we select uncertainty-guided pseudo-labelling (UGPL)

Methods	Sim10K		CS			
	D-ECE	**AP@0.5**	**D-ECE**	**AP@0.5**	**D-ECE**	**AP@0.5**
In-domain Detections						
Two-stage	30.6	66.2	20.2	38.3	-	-
Two-stage + post-hoc	21.7	66.3	17.0	38.0	-	-
Two-stage + TCD	18.2	66.7	15.7	39.9	-	-
Out-of-domain Detections						
	Sim10K to CS		**CS to CS-foggy**		**CS to BDD100K**	
Two-stage	17.3	33.9	9.2	22.7	19.0	23.3
Two-stage + post-hoc	13.8	34.1	8.7	22.7	16.0	23.3
Two-stage + TCD	13.1	33.9	8.54	25.2	12.9	23.5

Table 2: Calibration results with a two-stage detector (Faster-RCNN [32]) trained with its task-specific loss, applying post-hoc temperature scaling on a pre-trained two-stage detector, and training two-stage detector after adding our TCD loss.

Methods	Sim10K→CS		CS→CS-foggy	
	D-ECE	**AP@0.5**	**D-ECE**	**AP@0.5**
Single-stage Domain Adaptive Detectors				
EPM [13]	11.5	46.7	16.2	38.6
EPM + TCD	3.1	47.7	14.6	39.9
SSAL(UGPL) [27]	12.2	49.5	22.3	35.0
SSAL(UGPL) + TCD	5.3	51.4	18.1	35.2
Two-stage Domain Adaptive Detectors				
	Sim10K→CS		**CS→CS-foggy**	
SWDA [33]	17.3	40.9	18.8	36.6
SWDA + TCD	16.6	40.0	11.3	37.5

Table 3: Calibration results with two different single-stage domain-adaptive detectors, EPM [13] and SSAL(UGPL) [27], and a two-stage detector SWDA [33]. We report both calibration performance (ECE) and test accuracy (AP@0.5).

from SSAL [27]. In two-stage paradigm, we choose SWDA [33], which proposes strong local alignment and weak global alignment in an adversarial alignment framework.

Table 3 reports results with two single-stage domain-adaptive detectors: EPM [13], UGPL from [27]. After adding our proposed loss TCD, the calibration performance for both domain-adaptive detectors is significantly improved. In Sim10K→CS, both EPM+TCD and SSAL(UGPL)+TCD reduce ECE score by 8.44% and 6.83%, respectively. Further, we observe that after adding our auxiliary loss (TCD), there are notable gains in AP@0.5.

We show the impact of our TCD loss on the calibration performance of a two-stage domain-adaptive detector (SWDA [33]) in Table 3. Upon adding our loss TCD in SWDA, we observe that the calibration of SWDA is improved in both adaptation scenarios.

4.3 Experiments with Implicit Calibration Technique

We validate the effectiveness of our implicit calibration technique (ICT) (sec. 3.3) by integrating in a self-training based uncertainty-guided pseudo-labelling baseline (UGPL)) from [27]. UGPL is a strong baseline in terms of benchmarking model calibration because it leverages uncertainty to select pseudo instance-level labels in an unlabelled target domain for self-training. Table 4 reports results for following methods: SSAL(UGPL) [27], SSAL(UGPL)+ICT, SSAL(UGPL)+TCD, SSAL(UGPL)+ICT+TCD. We see that SSAL(UGPL)+ICT decreases D-ECE by 3.92% in Sim10→CS shift. Further, adding TCD shows significant improvement in calibration performance (D-ECE). Finally, using both ICT and TCD achieves the best calibration scores in both Sim10K and CS→CS-foggy shifts, thereby revealing that ICT and TCD are complementary.

4.4 Ablation Study and Analysis

For ablation experiments with TCD, we use one-stage detector (FCOS [36]. For ICT ablation experiments, we use the UGPL component of SSAL [27], which is a domain-adaptive detector.

Mitigating Under/Over-Confidence: Fig. 2 shows that our proposal, task-specific loss in conjunction with proposed TCD loss, facilitates a one-stage detector in mitigating both under-confident and over-confident detections in two different multi-class shift scenarios.

Method/Shift scenarios	Sim10k → CS		CS → CS-foggy	
	D-ECE	**AP@0.5**	**D-ECE**	**mAP@0.5**
SSAL(UGPL) [27]	12.2	49.5	22.3	35.0
SSAL(UGPL)+ICT	8.2	51.3	19.4	34.2
SSAL(UGPL)+TCD	5.3	51.4	19.1	36.0
SSAL(UGPL)+ICT+TCD	5.0	50.7	17.0	36.9

Table 4: Calibration results with our implicit calibration technique (ICT).

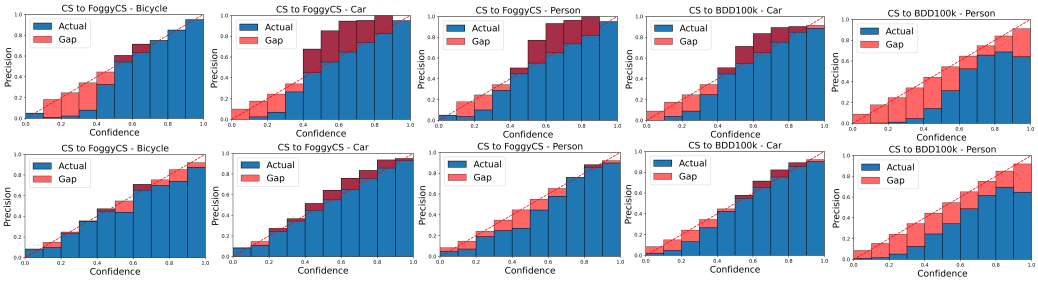


Figure 2: Class-wise reliability diagrams. Top row: One-stage detector trained with task-specific loss. Bottom row: One-stage detector trained with task-specific loss and our loss TCD.

Impact on test accuracy: We observe the impact on test accuracy after adding our TCD loss with the task-specific loss of one-stage detector in Table 5. Our proposal is capable of providing gains in test accuracy with visible margins over higher IoU thresholds and over the spectrum of large, medium and small objects.

Shift scenario	Method	mAP	AP@0.5	AP@0.75	AP@S	AP@M	AP@L
Sim10K → CS	One-stage	17.8	37.6	14.9	3.8	19.7	36.6
	One-stage+TCD	22.5	42.6	21.4	4.2	24.8	46.5
CS → BDD100K	One-stage	9.5	19.5	7.9	3.5	11.7	20.0
	One-stage+TCD	10.7	22.0	9.0	3.2	13.2	23.7

Table 5: Test accuracy after adding our TCD loss with the task-specific loss of one-stage detector (FCOS [36]).

Without detection component $\mathrm{d_{det}}$: Table 6 shows the impact on calibration performance without $\mathrm{d_{det}}$ component in TCD. We see that without $\mathrm{d_{det}}$, the calibration performance (D-ECE) degrades as well as AP@0.5 in all four domain shift scenarios.

Scenarios	Sim10k to CS				CS to CS-foggy				KITTI to CS				CS to BDD100K			
	OOD		In-domain		OOD		In-domain		OOD		In-domain		OOD		In-domain	
	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5	D-ECE	AP@0.5
w/o d_det	4.6	44.9	10.7	82.3	6.4	23.8	13.6	44.8	5.7	38.7	4.3	94.3	13.4	19.8	13.6	44.8
with d_det	3.6	42.4	8.5	83.4	4.0	22.4	11.2	48.3	2.8	40.3	5.1	94.7	9.6	22	11.2	48.3

Table 6: Impact on calibration performance without $\mathrm{d_{det}}$ component of TCD in four domain shift scenarios.

On different uncertainty quantification methods for ICT: Table 7 reports the calibration performance of ICT with different methods of quantifying uncertainty. We use variances across: predicted confidences only [27] $\mathrm{u_{conf.}}$; predicted confidences and center-x,center-y $\mathrm{u_{conf.,x,y}}$; predicted confidences and center-x,center-y, aspect-ratio (ours) $\mathrm{u_{conf.,x,y,ar}}$. Among different methods, our proposed technique of estimating detection uncertainty in ICT provides the lowest D-ECE score.

Methods	D-ECE	AP@0.5
SSAL(UGPL) [27]	12.2	49.5
SSAL(UGPL)+ICT($\mathrm{u_{conf.}}$)	9.2	50.5
SSAL(UGPL)+ICT($\mathrm{u_{conf.,x,y}}$)	8.4	51.1
SSAL(UGPL)+ICT($\mathrm{u_{conf.,x,y,ar}}$)	8.2	51.3

Table 7: Calibration performance of ICT when using different ways of quantifying detection uncertainty. See text for details.

Limitation: Current study does not explore the impact on calibration with respect to objects sizes. Further, our train-time calibration loss could potentially result in less improvement in classes with relatively less instances.

5 Conclusion

In this paper, we approached the challenging problem of calibrating DNN-based object detectors for in-domain and out-of-domain detections and improving their calibration in the domain-adaptation context. We proposed an auxiliary train-time calibration loss TCD that jointly calibrates the class-wise confidences and localization performance. Further, we develop an implicit calibration technique (ICT) for self-training based domain adaptive detectors. Results show that TCD loss is capable of improving calibration of both one-stage and two-stage detectors. Finally, we show that ICT and TCD together results in well-calibrated domain-adaptive detectors.

References

[1] Bohdal, O., Y. Yang, and T. Hospedales (2021). Meta-calibration: Meta-learning of model calibration using differentiable expected calibration error. *arXiv preprint arXiv:2106.09613*.

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

[3] Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] DeVries, T. and G. W. Taylor (2018). Learning confidence for out-of-distribution detection in neural networks. *stat 1050*, 13.

[5] Dusenberry, M. W., D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai (2020). Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 204–213.

[6] Gal, Y. and Z. Ghahramani (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR.

[7] Geiger, A., P. Lenz, and R. Urtasun (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361. IEEE.

[8] Grigorescu, S., B. Trasnea, T. Cocias, and G. Macesanu (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics 37*(3), 362–386.

[9] Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR.

[10] Hebbalaguppe, R., J. Prakash, N. Madan, and C. Arora (2022). A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. *arXiv preprint arXiv:2203.13834*.

[11] Hein, M., M. Andriushchenko, and J. Bitterwolf (2019). 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*, pp. 41–50.

[12] Hendrycks, D., M. Mazeika, and T. Dietterich (2018). Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*.

[13] Hsu, C.-C., Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang (2020). Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *European Conference on Computer Vision*, pp. 733–748. Springer.

[14] Islam, M., L. Seenivasan, H. Ren, and B. Glocker (2021). Class-distribution-aware calibration for long-tailed visual recognition. *arXiv preprint arXiv:2109.05263*.

[15] Johnson-Roberson, M., C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan (2017). 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)*, pp. 746–753. IEEE.

[16] Karimi, D. and A. Gholipour (2022). Improving calibration and out-of-distribution detection in deep models for medical image segmentation. *IEEE Transactions on Artificial Intelligence*.

[17] Kull, M., M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems 32*.

[18] Kull, M., T. Silva Filho, and P. Flach (2017). Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pp. 623–631. PMLR.

[19] Kumar, A., S. Sarawagi, and U. Jain (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR.

[20] Küppers, F., J. Kronenberger, A. Shantia, and A. Haselhoff (2020, June). Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[21] Liang, G., Y. Zhang, X. Wang, and N. Jacobs (2020). Improved trainable calibration method for neural networks on medical imaging classification. *arXiv preprint arXiv:2009.04057*.

[22] Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.

[23] Maroñas, J., D. Ramos, and R. Paredes (2021). On calibration of mixup training for deep neural networks. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 67–76. Springer.

[24] Meronen, L., C. Irwanto, and A. Solin (2020). Stationary activations for uncertainty calibration in deep learning. *Advances in Neural Information Processing Systems 33*, 2338–2350.

[25] Mukhoti, J., V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania (2020). Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems 33*, 15288–15299.

[26] Müller, R., S. Kornblith, and G. E. Hinton (2019). When does label smoothing help? *Advances in neural information processing systems 32*.

[27] Munir, M. A., M. H. Khan, M. Sarfraz, and M. Ali (2021). Ssal: Synergizing between self-training and adversarial learning for domain adaptive object detection. *Advances in Neural Information Processing Systems 34*.

[28] Naeini, M. P., G. Cooper, and M. Hauskrecht (2015). Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[29] Padhy, S., Z. Nado, J. Ren, J. Liu, J. Snoek, and B. Lakshminarayanan (2020). Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *arXiv preprint arXiv:2007.05134*.

[30] Pereyra, G., G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton (2017). Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

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

[32] Ren, S., K. He, R. Girshick, and J. Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99.

[33] Saito, K., Y. Ushiku, T. Harada, and K. Saenko (2019). Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6956–6965.

[34] Sakaridis, C., D. Dai, and L. Van Gool (2018). Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision 126*(9), 973–992.

[35] Sharma, M., O. Saha, A. Sriraman, R. Hebbalaguppe, L. Vig, and S. Karande (2017). Crowdsourcing for chromosome segmentation and deep classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 34–41.

[36] Tian, Z., C. Shen, H. Chen, and T. He (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 9627–9636.

[37] Tomani, C., S. Gruber, M. E. Erdem, D. Cremers, and F. Buettner (2021). Post-hoc uncertainty calibration for domain drift scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10124–10132.

[38] Tompson, J., R. Goroshin, A. Jain, Y. LeCun, and C. Bregler (2015). Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 648–656.

[39] Xu, C.-D., X.-R. Zhao, X. Jin, and X.-S. Wei (2020). Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11724–11733.

[40] Yu, F., W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell (2018). Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687 2*(5), 6.

[41] Yu, R. and G. S. Alì (2019). What's inside the black box? ai challenges for lawyers and researchers. *Legal Information Management 19*(1), 2–13.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Since our work belongs to the fundamental problem in machine learning, it inherits all the benefits of object detection.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No] Our work is not theoretical.
   (b) Did you include complete proofs of all theoretical results? [No] Our work is not theoretical.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In supplementary.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We are using implementations of existing published works, and those are cited.
   (b) Did you mention the license of the assets? [No] We are citing the previous work whose publicly shared code we use for reporting baselines.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.