International Journal of Computer Vision (2025) 133:1033–1047
https://doi.org/10.1007/s11263-024-02219-z

Systematic Evaluation of Uncertainty Calibration in Pretrained Object Detectors

Denis Huseljic[1] · Marek Herde[1] · Paul Hahn[1] · Mehmet Müjde[1] · Bernhard Sick[1]

Received: 1 October 2022 / Accepted: 5 August 2024 / Published online: 31 August 2024

Abstract
In the field of deep learning based computer vision, the development of deep object detection has led to unique paradigms (e.g., two-stage or set-based) and architectures (e.g., FASTER- RCNN or DETR) which enable outstanding performance on challenging benchmark datasets. Despite this, the trained object detectors typically do not reliably assess uncertainty regarding their own knowledge, and the quality of their probabilistic predictions is usually poor. As these are often used to make subsequent decisions, such inaccurate probabilistic predictions must be avoided. In this work, we investigate the uncertainty calibration properties of different pretrained object detection architectures in a multi-class setting. We propose a framework to ensure a fair, unbiased, and repeatable evaluation and conduct detailed analyses assessing the calibration under distributional changes (e.g., distributional shift and application to out-of-distribution data). Furthermore, by investigating the influence of different detector paradigms, post-processing steps, and suitable choices of metrics, we deliver novel insights into why poor detector calibration emerges. Based on these insights, we are able to improve the calibration of a detector by simply finetuning its last layer.

Keywords Object detection · Uncertainty calibration · Uncertainty modeling · Evaluation

1 Introduction

Over the last few years, deep object detection has had an impressive evolution. Architectures such as FASTER-RCNN (Ren, He, Girshick, and Sun, 2015) and DETR (Carion et al., 2020) showed remarkable performances on challenging datasets. However, these architectures usually lack in assessing their own uncertainty associated with their predictions (Feng, Harakeh, Waslander, and Dietmayer, 2021). Often, the quality of probabilistic predictions is poor and should not be trusted out of the box. For example, consider a detection task in an autonomous driving environment. Here, over- or underconfident predictions leading to wrong decisions may result in accidents with cars or even more vulnerable road users such as cyclists (Bieshaar, Zernetsch, Hubert, Sick, and Doll, 2018).

Ideally, we wish to have a detector capable of providing well-calibrated probabilistic predictions while also being able to recognize object-like entities in images that do not originate from the training data distribution. In the case of well-calibrated probabilistic predictions such as class probabilities, we would like a probability to represent the actual occurrence frequency. For instance, if our detector predicts "pedestrian" with a probability of 0.8, then we expect the ground truth to be a pedestrian 80 percent of the time. The concept of how well a model can reflect this is referred to as *uncertainty calibration* (Guo, Pleiss, Sun, and Weinberger, 2017). The latter property enables to identify distributional changes, which is crucial when deploying detectors in the real world, especially if the detector is faced data samples that

Communicated by Wenjun Kevin Zeng.

✉ Denis Huseljic
dhuseljic@uni-kassel.de

Marek Herde
marek.herde@uni-kassel.de

Paul Hahn
paul.hahn@uni-kassel.de

Mehmet Müjde
mehmet.muejde@uni-kassel.de

Bernhard Sick
bsick@uni-kassel.de

[1] Intelligent Embedded Systems, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Hesse, Germany

Fig. 1 Calibration plot and detector predictions from DETR trained on a subset of COCO

(a) Top-Label Calibration Plot (b) Confident Prediction (c) Uncertain Prediction

are not well covered by the training data distribution (Ovadia et al., 2019). Thereby, we can determine to which degree we can trust a prediction. Consider a scenario from autonomous driving in which a flock of sheep crosses a road and our detector was only trained on cars and pedestrians. In such a case, the network should correctly identify this problematic situation due to a distributional change of the samples and leave further decisions to the driver.

Figure 1 illustrates the attempt of assessing the calibration quality for the popular pretrained object detector DETR fine-tuned on a subset of the COCO dataset (Lin et al., 2014) with two object classes (giraffe and elephant). Figure 1a represents the *Top-Label Calibration Plot* (TCP) that only considers the highest predicted probabilities of a detector as done in the literature (e.g., expected calibration error in (Neumann, Zisserman, and Vedaldi, 2018)). As these should reflect the actual occurrence, this diagram shows accuracy as a function of predicted probability. A perfectly calibrated detector would yield a model calibration (red) lying on the diagonal (black), with the predicted probability equal to the number of correct predictions. Hence, in this example, we seem to observe a well-calibrated detector. Note, the red curve starts at about 0.5 because there are no lower maximum probabilities. In Fig. 1b and c, we see detector predictions on two images. The actual predictions are marked red, while the ground truth boxes are marked green. Figure 1b demonstrates a very confident but reasonable prediction for the class giraffe. Accordingly, this prediction will positively influence the assessment in the calibration plot on the left. In contrast, the right figure shows an uncertain false prediction for the background class presumably because of the fence. Here, the probability for the class background is only slightly higher than the one for the correct class giraffe. This outlines the importance of considering all available information for evaluation (i.e., all probabilities, not only the highest ones). Thus, the TCP should not be used for evaluation. In the remainder of this article, we will also see that we should assess the calibration of different classes individually.

2 Contributions

In this work, we investigate multi-class calibration properties of different **pretrained state-of-the-art object detection architectures** without and with distributional changes, i.e., shifted and out-of-distribution data. We provide detailed analyses of two different detector paradigms (i.e., two-stage and set-based) with two architectures (i.e., FASTER- RCNN and DETR), examine their ability to identify distributional changes, and their calibration quality from two perspectives. In the first perspective, we consider the inference of an object detection architecture as suggested by the literature with the aim to achieve a high generalization performance in the form of mean averge precsision (mAP) (Lin et al., 2014). More precisely, this means that the raw predictions of a detector are often *post-processed* (e.g., removing duplicates) to obtain a more appropriate prediction set for an image. We often encounter such post-processing steps (e.g., non-maximum suppression, NMS) when working with popular object detection architectures such as FASTER- RCNN. In the second perspective, we focus on the *raw outputs of the DNN* with the goal to examine the quality of the probabilistic predictions before applying any post-processing steps. Consequently, we focus on assessing the DNN's calibration and not the calibration of the whole detection pipeline. This way, we intend to give deeper insights, hopefully guiding future research and making it easier to develop techniques for uncertainty modeling in object detection. Furthermore, in contrast to related work evaluating only the highest class probability (Schwaiger et al., 2021), we focus on a *multi-class setting* in which all class probabilities are considered when evaluating the calibration of the detectors. As will be shown in this article, while detectors often appear to be well-calibrated regarding the highest predicted class probability, their probabilistic outputs of the remaining classes are not. This can lead to severe problems in many cases. For instance, suppose a detector's highest predicted probability is 0.6 for the class pedestrian and its second-highest probability is 0.4 for the class e-scooter driver. In such an example, it is evident that this probability must also be well-calibrated to avoid

wrong decisions. Moreover, since changes in the *data distribution* are widespread in real applications, it is essential that detectors are able to recognize and handle them (Ovadia et al., 2019). Accordingly, we evaluate the uncertainty of detectors under distributional shifts (i.e., different sample but approx. same class distributions) and also their ability to identify out-of-distribution samples (i.e., both sample and class distributions differ). More specifically, we construct shifted and out-of-distribution dataset versions to assess the properties of the detectors. In summary, we investigate the following research questions:

- How can we build a modular calibration evaluation framework that is suitable for various object detection architectures and ensures an unbiased and repeatable evaluation?
- Which metrics should be used to evaluate multi-class object detection architectures regarding their calibration and how should these metrics be applied?
- How do post-processing steps in a detection pipeline influence the calibration and do architectures that avoid them deliver better calibrated predictions?
- How well are the class probabilities of pretrained object detectors calibrated when the detectors are applied on samples from shifted versions of the training distribution?
- How well can a detector identify new objects from out-of-distribution data by means of its probabilistic outputs?

Based on the experimental results and findings, we improve the calibration of a pretrained object detector by simply finetuning its last layers through changing the importance of the background class. Our implementation is publicly available at https://github.com/ies-research/uncertainty-object-detection. Our contributions can be summarized as follows:

- We formally define a detailed evaluation process that distinctively assesses uncertainty calibration of pretrained object detectors, encompassing all predictions to ensure a comprehensive and accurate evaluation.
- We conduct an in-depth empirical study to answer the previously defined research questions and provide qualitative results for an intuitive understanding.
- We derive a recalibration approach from the insights of our empirical study and validate its effectiveness through case studies on two datasets.
- We identify and suggest several important directions for future research.

While the work presented in (Harakeh and Waslander, 2021) shows similarities to ours, our study differs in several key aspects. First, in contrast to their work, we initially bypass the need to train a *probabilistic object detector*. We believe many practitioners, especially those unfamiliar with deep learning research, prefer leveraging pretrained object detectors for application integration. Therefore, our focus is the evaluation of uncertainty of these pre-trained models. This approach is pragmatic for practitioners seeking to utilize existing models without engaging in the extensive training of large object detectors. Second, we present a detailed and formal definition of uncertainty evaluation in this setting to simplify the process for new researchers, helping to streamline the evaluation process in the field. Lastly, our study includes an in-depth focus on qualitative results, offering a more comprehensive and intuitive view of the problem. This includes the introduction of a straightforward recalibration method, readily applicable in the discussed scenarios.

The remainder of this article is organized as follows: In Sect. 3, we give a formal definition of our problem setting and introduce the considered object detection architectures which will be exemplarily used as they are the most prominent representatives of their paradigms. Section 4 analyzes related research regarding uncertainty calibration in classification and object detection. Afterward, in Sect. 5, we propose our evaluation framework and the employed metrics, which allow us to assess the calibration and uncertainty of object detection architectures, and in Sect. 6, we address the aforementioned research questions by quantitative and qualitative analyses. Based on these insights, in Sect. 7, we recalibrate a detector by finetuning its last layers. Finally, in Sect. 8, we highlight potential future research directions and conclude our work in Sect. 9.

3 Problem Setting

This section introduces the notation regarding object detection and calibration used throughout this article and the architectures we evaluate.

3.1 Notation

In our setting, we consider object detection problems for computer vision. We represent a color image (i.e., input sample to an object detector) by a tensor $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^{W \times H \times C}$ describes the space of all possible images with $H, W, C \in \mathbb{N}$ as height, width, and number of color channels. An image $\boldsymbol{x}$ can contain an unknown number of objects, of which each is represented by a box $\boldsymbol{b} \in \mathcal{B}$ describing its position and a label $y \in \mathcal{Y}$ explaining its class. The sets $\mathcal{B} = [0, 1]^4$ and $\mathcal{Y} = \{1, \ldots, K\}$ define the space of all possible boxes and $K \in \mathbb{N}_{>1}$ class labels, respectively. We define the target set for a single image as $\mathcal{T} \in \mathcal{P}(\mathcal{Y} \times \mathcal{B})$ where $\mathcal{P}(\cdot)$ denotes the power set. For example, an image $\boldsymbol{x}_n$ with two objects would have the target set $\mathcal{T}_n = \{(y_1, \boldsymbol{b}_1), (y_2, \boldsymbol{b}_2)\}$. Finally, we describe a dataset consisting of images and targets as $\mathcal{D} = \{(\boldsymbol{x}_n, \mathcal{T}_n)\}_{n=1}^{N}$ where we have a total number

of $N \in \mathbb{N}_{>0}$ images. The images $\boldsymbol{x}_n$ are distributed according to the distribution $p(\boldsymbol{x})$ and boxes $\boldsymbol{b}_{nt}$ and labels y_{nt} in the target set $\mathcal{T}_n$ are assumed to be distributed according to $p(\boldsymbol{b}|\boldsymbol{x}_n)$ and $p(y|\boldsymbol{b}_{nt}, \boldsymbol{x}_n)$, respectively.

Formally, an object detector is a function $f^{\boldsymbol{\omega}} : \mathcal{X} \to \mathcal{P}(\Delta_K \times \mathcal{B})$, where $\boldsymbol{\omega}$ is the set of trainable parameters usually optimized with techniques such as gradient descent (Bishop, 2006) and Δ_K is the K-simplex within the $K+1$ dimensional unit hypercube spanned by the K classes and the additional background class $\otimes$ (Ren, He, Girshick, and Sun, 2015; Carion et al., 2020). Most object detectors employ a background class to distinguish whether an object is present in a proposed region. Hence, an object detector is a function that takes an image $\boldsymbol{x}_n$ as input and outputs a set of predictions $\hat{\mathcal{T}}_n = \{(\hat{\boldsymbol{p}}_{nt}, \hat{\boldsymbol{b}}_{nt})\}_{t=1}^{|\hat{\mathcal{T}}_n|}$ where $\hat{\boldsymbol{p}}_n \in \Delta_K$ are class probabilities for the corresponding box $\hat{\boldsymbol{b}}_n \in \mathcal{B}$. The number of predictions per image $|\hat{\mathcal{T}}_n| \in \mathbb{N}$ might vary depending on the architecture.

Calibration expresses the quality of the predicted probabilities of a model. Formally, for all probability vectors $\hat{\boldsymbol{p}}$ of a trained detector on the simplex Δ_K, we want to satisfy

$$P(Y = y|\hat{\boldsymbol{p}}) = \hat{p}_y \text{ for all } y \in \mathcal{Y} \cup \{\otimes\}, \tag{1}$$

where Y is the random variable for the true class. Intuitively, for an object detector, this means that all predicted probabilities $\hat{p}_y$ for a prediction $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{b}})$ should match their true (but unknown) probability. For example, collecting all predictions where the object detector returned a probability of 0.2 for class pedestrian, we want 20% of them to actually be a pedestrian. Furthermore, this should hold for all classes $y \in \mathcal{Y}$ and probabilities $p_y \in [0, 1]$. In contrast to similar works which assess the calibration based solely on the predicted class probability max $\hat{\boldsymbol{p}}$ (Kuppers, Kronenberger, Shantia, and Haselhoff, 2020; Neumann, Zisserman, and Vedaldi, 2018), we focus on calibration always considering all class probabilities as suggested in Kumar et al. (2019).

3.2 Detection Architectures

This work examines two essential kinds of object detection paradigms, namely *two-stage* and *set-based*. Specifically, we focus on the architectures FASTER- RCNN and DETR, as they represent the paradigms' most prominent representatives. Since architectures of the *one-stage* paradigm do not directly output multi-class probability vectors as defined in our problem setting, we leave its investigation for future work. In the following, we briefly describe how the architectures and their detection pipeline work, including the respective post-processing steps.

FASTER- RCNN (Ren, He, Girshick, and Sun, 2015) is one of the most prominent object detection architectures and its forward-propagation is performed in two stages. First, it uses a so-called region proposal network (RPN) to predict multiple regions that may contain potential objects. The second stage utilizes these regions and deploys another DNN to solve a simple classification and regression problem. The output of this network is called prediction and is denoted by $f^{\boldsymbol{\omega}}(\boldsymbol{x})$. In this article, we only focus on the calibration of the second stage and leave the classification problem in the RPN (first stage) for future work. For specifying the post-processing steps, we consider the implementation of detectron2 (Wu, Kirillov, Massa, Lo, and Girshick, 2019). Out of 1000 predictions, we discard those where the maximum probability within $\hat{\boldsymbol{p}}$ is below a certain threshold. Additionally, to avoid duplicate predictions for the same ground truth object, we filter them out using NMS. That means, that we have a varying number of predictions per image $|\hat{\mathcal{T}}_n|$.

DETR (Carion et al., 2020) is a set-based object detection architecture based on transformers Vaswani et al. (2017). The term set-based arises as DETR views object detection as a direct set prediction problem in which we can detect objects directly without the need for anchors or region proposals. To achieve this, DETR learns attention weights in a transformer describing the pixel and object relationships in an image. In contrast to other object detection architectures, DETR learns to avoid duplicates through its set-based loss function and therefore does not require any post-processing steps. This means, that it predicts a fixed number of $|\hat{\mathcal{T}}_n| = 100$ objects per image, as suggested by the authors.

4 Related Work

Originally, calibration properties have been researched in the context of image classification (Ovadia et al., 2019), and in recent years, considerable work has been done.

For image classification, Guo et al. (2017) showed that although modern architectures such as ResNet (He, Zhang, Ren, and Sun, 2016) achieve outstanding generalization performance, they provide poorly calibrated and overconfident outputs. They introduced temperature scaling, a multi-class extension to Platt scaling, which improves a DNN's calibration by scaling its predictions with a parameter learned from a separate dataset. Kull et al. (2019) noticed that while temperature scaling improves the calibration of the highest predicted probability of a vector containing all class probabilities, the remaining probabilities are still poorly calibrated. Accordingly, they proposed Dirichlet Calibration for the multi-class setting to improve the quality of all predicted probabilities. Furthermore, in a large-scale evaluation, Ovadia et al. (2019) assessed the behavior of various models under distributional changes and showed a deteriorated quality of their uncertainty estimates.

For classification problems, we only need to consider that the input of a model is assigned to a single class. In object detection, however, we need to consider an unknown number of objects per image, jointly solve regression and classification problems, and often use heuristics such as NMS to obtain our final predictions. These complexities make the transfer of existing uncertainty modeling and calibration evaluation concepts to the object detection task challenging.

One of the first methods regarding uncertainty calibration in deep object detection was proposed by Neumann et al. (2018). Their article demonstrates the poor calibration of pedestrian detection models and proposes an extension of temperature scaling, avoiding the need for a separate calibration dataset. More recently, Kuppers et al. (2020) discovered that the calibration of a model depends on an object's position in an image and proposed box-sensitive recalibration methods for improvement. As most works focus on the evaluation of the entire detection pipeline, Schwaiger et al. (2021) assessed the influence of NMS on the calibration. They examined the highest predicted probability of DNNs used in different detectors and showed the negative impact of NMS on the calibration quality. Küppers et al. (2022) introduced multivariate confidence calibration as an extension of well-known calibration methods. Their extension allows for a recalibration that is aware of additional features such as bounding boxes and shape information. Furthermore, they extend the commonly used expected calibration error for object detection and segmentation tasks. Munir et al. (2022) investigated the calibration of object detectors in both in- and out-of-distribution scenarios. They found these to be poorly calibrated and addressed this by proposing to use a new regularization term during training. Pathiraja et al. (2023) proposed a calibration method utilizing an auxiliary loss that jointly calibrates multi-class probabilities and bounding box localization by leveraging predictive uncertainties. The auxiliary loss can be incorporated in popular architectures and is optimized during training, avoiding the computational overhead of post-hoc recalibration.

There were also several advancements in which object detectors were studied in the context of out-of-distribution (OOD) detection. Du, Gozum, et al. (2022) improved the detection of OOD samples by focusing on the learned feature representations of detectors. These were highly irregular, not following a Gaussian distribution. They proposed an additional loss term employing the von Mises-Fisher distribution as a solution. Liu et al. (2022) tackled the challenge of semi-supervised object detection by developing an OOD sample filtering process to improve training by excluding OOD objects. Du, Wang, Gozum, and Li (2022) enhanced OOD detection in videos by utilizing temporal and spatial information for regularization. Liang et al. (2023) propose a score for better detection by replacing NMS with a clustering-based selection method. Wilson et al. (2023) introduced the concept of Sensitivity-Aware Features by identifying high-sensitivity layers and training a multi-layer perceptron to distinguish these features from in-distribution samples.

In this article, we focus on these aspects regarding the evaluation of pretrained detectors. An in-depth study on recalibration techniques of detectors is subject to our future work.

5 Evaluation Framework

Our framework strategically builds on classification concepts, recognizing that, at its core, object detection fundamentally hinges on classification tasks. However, despite the similarity to classification, it is not straightforward to adapt evaluation protocols to object detectors. The literature suggests many techniques for evaluating the calibration and uncertainty of object detection architectures. However, most of these do not clearly define the steps toward an evaluation metric. In particular, it is not clearly described how to incorporate the predictions of object classes (i.e., true and false positives, TP and FP), the predictions of the background class (i.e., true negatives, TN) and missing predictions (i.e., false negatives, FN). *As an attempt to answer our first research question and to make the evaluation understandable and repeatable, we propose a modular evaluation pipeline consisting of three steps* (*cf.* Fig. 2). Starting point for the evaluation is the raw output $f^{\boldsymbol{\omega}}(\boldsymbol{x})$ of an object detectors's forward propagation. To obtain an evaluation metric, we look at the subsequent three steps:

1. Post-Processing: Involves filtering out redundant predictions as done by NMS.
2. Creation of an Evaluation Set: Involves matching predictions with ground truth objects.
3. Metric Evaluation: Involves defining calibration metrics based on the evaluation set.

Each of these steps can be adapted according to the requirements of a specific architecture. The upcoming subsections highlight essential design choices in each step that are necessary for a fair evaluation of all considered architectures.

5.1 Post-processing

Since there exist a wide range of object detection paradigms and architectures, there are also a lot of different detector-specific ways to obtain a final prediction set for an image. Many detection pipelines employ NMS and filter out several predictions based on different criteria to achieve satisfying generalization performances (measured, e.g., with mAP) (Ren, He, Girshick, and Sun, 2015). However, these post-processing steps remove predictions from the pre-

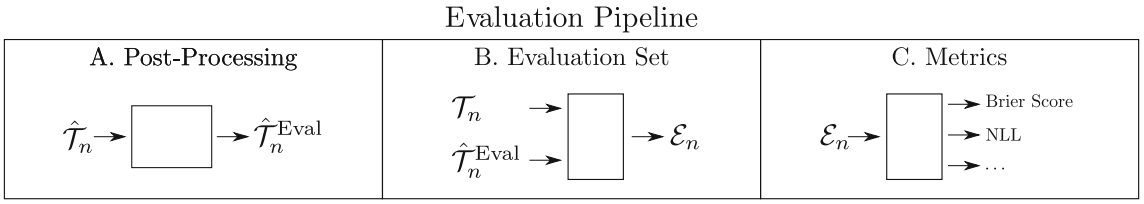


Fig. 2 Our modular evaluation framework for each architecture consists of three steps

dictions set $\hat{\mathcal{T}}_n$, which might carry valuable information regarding calibration and uncertainty assessment. Thus, it is essential to be aware of specific post-processing steps that a detection pipeline uses as these might bias the evaluation, especially when evaluating uncertainties (Schwaiger et al., 2021). In our evaluation, we specify a set $\hat{\mathcal{T}}_n^{\text{Eval}}$ to define the (final) prediction set for sample $\boldsymbol{x}_n$ that are considered during evaluation. For example, when considering NMS as a post-processing method, this set will be a subset of the original predictions, i.e., $\hat{\mathcal{T}}_n^{\text{Eval}} \subseteq \hat{\mathcal{T}}_n$.

On one hand, we consider this set from a perspective which treats a detector as a black box. In particular, we use post-processing steps as it would be done when deploying the detection pipeline in practice. For example, for the detection architecture FASTER- RCNN, the predictions $\hat{\mathcal{T}}_n^{\text{Eval}}$ for image $\boldsymbol{x}_n$ after post-processing are defined by i) removing all predictions $(\hat{\boldsymbol{p}}_{nt}, \hat{\boldsymbol{b}}_{nt})$ where the maximum of $\hat{\boldsymbol{p}}_{nt}$ is below a certain threshold, ii) removing all duplicate predictions with NMS, and iii) keeping a certain number of predictions with the highest maximum probability. At this point, it becomes apparent that with this perspective we will neglect many predictions in the subsequent course of the evaluation. These might have better calibrated class probabilities or we might lose potentially helpful predictions which we could exploit to identify distributional changes.

On the other hand, we also intend to evaluate a detector in a more detailed perspective without post-processing. Specifically, we determine the calibration quality of $f^{\boldsymbol{\omega}}$ and not of the entire detection pipeline by considering the raw predictions that were made by the DNN (i.e., $f^{\boldsymbol{\omega}}$) without filtering any predictions. Thus, our set for evaluation $\hat{\mathcal{T}}_n^{\text{Eval}}$ is given by all the predictions $\hat{\mathcal{T}}_n$ of the DNN.

5.2 Evaluation Set

In order to be able to evaluate the predictions, it is necessary to have a suitable assignment of predictions from the prediction set $\hat{\mathcal{T}}_n^{\text{Eval}}$ to ground-truth objects in the target set $\mathcal{T}_n$. We can realize such an assignment in multiple ways. However, it is important to note that the previously mentioned perspectives leading to our prediction set $\hat{\mathcal{T}}_n^{\text{Eval}}$ play a critical role in that selection. Generally, if the detector aims at avoiding duplicates during inference (e.g., FASTER- RCNN with post-processing), then we have to force an assignment of a single prediction to a single ground truth object. Otherwise, we would not penalize duplicate predictions (FP) in this case. Conversely, if our detector predicts duplicates (e.g., FASTER- RCNN without post-processing), we must ensure that multiple predictions can be assigned to a single ground truth object. Otherwise, we would penalize duplicate predictions, in this case, even if the detector is supposed to make multiple per object. As an example, consider evaluating the raw predictions of FASTER- RCNN. Here, we must ensure that we assign multiple predictions to a single ground truth object. FASTER- RCNN without NMS may correctly predict multiple boxes for a single ground truth object. On the other hand, it is essential to use a one-to-one assignment for DETR, since we expect that a DETR based detector learned to avoid duplicate predictions during inference.

To build our evaluation set $\mathcal{E}_n$ for a sample, we therefore focus on a simple matching of predictions to ground truth objects based on the *Intersection over Union* (IOU) between predicted and ground truth box. In particular, anything above an IOU threshold of 0.5 is considered as matched. We match the pair with the highest IOU when we need a one-to-one assignment. Note that we prefer this simple matching over the matching used in the respective architecture (e.g., DETR's Hungarian matcher) to compare different architectures against each other in the same evaluation setting. We define the index sets of matches based on the predictions $\hat{\mathcal{T}}_n^{\text{Eval}}$ and targets $\mathcal{T}_n$ as

$$\mathcal{M}_n = \{(i, j) \mid (y_{ni}, \boldsymbol{b}_{ni}) \text{ matches } (\hat{\boldsymbol{p}}_{nj}, \hat{\boldsymbol{b}}_{nj})\}, \tag{2}$$

$$\overline{\mathcal{M}}_n^{\mathcal{T}} = \{j \mid (\hat{\boldsymbol{p}}_{nj}, \hat{\boldsymbol{b}}_{nj}) \text{ is not matched}\}, \tag{3}$$

$$\overline{\mathcal{M}}_n^{\hat{\mathcal{T}}} = \{i \mid (y_{ni}, \boldsymbol{b}_{ni}) \text{ is not matched}\}. \tag{4}$$

Consequently, for our evaluation, we define the evaluation set as

$$\mathcal{E}_n = \underbrace{\{(y_{ni}, \boldsymbol{b}_{ni}, \hat{\boldsymbol{p}}_{nj}, \hat{\boldsymbol{b}}_{nj})\}_{(i,j)\in\mathcal{M}_n}}_{\text{matched predictions (TP and FP)}} \cup \underbrace{\{(y^{\otimes}, \boldsymbol{b}^{\otimes}, \hat{\boldsymbol{p}}_{nj}, \hat{\boldsymbol{b}}_{nj})\}_{j\in\overline{\mathcal{M}}_n^{\mathcal{T}}}}_{\text{unmatched predictions (FP and TN)}} \cup \underbrace{\{(y_{ni}, \boldsymbol{b}_{ni}, \hat{\boldsymbol{p}}^{\otimes}, \boldsymbol{b}^{\otimes})\}_{i\in\overline{\mathcal{M}}_n^{\hat{\mathcal{T}}}}}_{\text{missing predictions (FN)}}, \tag{5}$$

where $y^{\otimes} = K + 1$ denotes the label for the background class, $\boldsymbol{b}^{\otimes} \in \mathcal{B}$ is an arbitrary box, and $\hat{\boldsymbol{p}}^{\otimes} \in \Delta_K$ is a probability vector assigning all its probability mass to the background class $K + 1$. With this set, our evaluation considers not only all predictions (TP, FP, and TN) but also missing ones (FN). Hence, we obtain an unbiased evaluation as we do not ignore potentially valuable predictions.

5.3 Metric Definition

The final step toward our framework consists of defining the required metrics based on the evaluation sets $\mathcal{E}_n$. More specifically, we focus on proper scoring rules (Feng, Harakeh, Waslander, and Dietmayer, 2021), calibration plots and errors (Kumar, Liang, and Ma, 2019) for the evaluation of the calibration, and on the entropy (Ovadia et al., 2019) to assess if a detector can identify distributional changes. Note that we can easily incorporate additional metrics for evaluation if necessary. To further simplify the notation, we define the evaluation set containing all predictions as $\mathcal{E} = \bigcup_{n=1}^{N} \mathcal{E}_n$, an element of that set as $d \in \mathcal{E}$ (i.e., matched, unmatched, or missing predictions), and a one-hot encoded version for y_{nt} as $\boldsymbol{y}_{nt}$.

Proper scoring rules are prevalent metrics for evaluating predictive probability distributions. Scoring rules are functions that map a probability distribution $\hat{\boldsymbol{p}}_{nt}$ and a ground truth label y_{nt} to a score. This score expresses how well distribution $\hat{\boldsymbol{p}}_{nt}$ matches the ground truth distribution of y_{nt}. Furthermore, we call it proper if only the ground truth distribution leads to a minimum score value. For our evaluation, we use the *Negative Log-Likelihood* (NLL) defined as

$$\mathrm{NLL}(\mathcal{E}) = -\frac{1}{|\mathcal{E}|} \sum_{d \in \mathcal{E}} \boldsymbol{y}_{nt}^{\mathrm{T}} \ln \hat{\boldsymbol{p}}_{nt}, \tag{6}$$

where ln is applied element-wise to the entries of $\hat{\boldsymbol{p}}_{nt}$. Furthermore, we also use the *Brier Score* (BS) defined as

$$\mathrm{BS}(\mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{d \in \mathcal{E}} \| \boldsymbol{y}_{nt} - \hat{\boldsymbol{p}}_{nt} \|^2, \tag{7}$$

where $\| \cdot \|^2$ is the Euclidean distance. Both the NLL and BS are proper scoring rules and we refer the reader to Feng et al. (2021) for a more detailed explanation.

Calibration plots or reliability diagrams are often employed to assess a model's calibration visually (Guo, Pleiss, Sun, and Weinberger, 2017). Their idea is to visualize Eq. 1 by plotting approximations of the true (but unknown) probability against the detector's predicted probability. Hence, we estimate both quantities empirically by splitting the probability space $[0, 1]$ into M adjacent bins (i.e., intervals) of equal sizes, assigning all predictions based on their probability, and computing accuracy and mean probability as defined below. On the one hand, we employ the *Top-Label Calibration Plot* (TCP), in which we consider a bin m containing all predictions $\mathcal{B}_m \subseteq \mathcal{E}$ where the maximum probability falls into the respective bin interval. The mean probability and accuracy are therefore defined by

$$\mathrm{prob}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{d \in \mathcal{B}_m} \max \hat{\boldsymbol{p}}_{nt}, \tag{8}$$

$$\mathrm{acc}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{d \in \mathcal{B}_m} \mathbb{I}(\arg\max \hat{\boldsymbol{p}}_{nt} = y_{nt}), \tag{9}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Since the TCP only considers the calibration of the most probable class, it neglects to assess all other predicted probabilities. For this reason, we also employ the *Marginal Calibration Plot* (MCP), which depicts the mismatch of all probabilities (Kumar, Liang, and Ma, 2019). In such a case, we define $\mathcal{B}_m$ for each class k individually and only consider the probabilities of the respective class when assigning predictions to a bin. Thus, the average mean probability and accuracy in bin m for class k is defined as

$$\mathrm{prob}_k(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{d \in \mathcal{B}_m} \hat{p}_{ntk}, \tag{10}$$

$$\mathrm{acc}_k(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{d \in \mathcal{B}_m} \mathbb{I}(k = y_{nt}). \tag{11}$$

By plotting the accuracy (y-axis) against mean probability (x-axis) in a two-dimensional Cartesian coordinate system, we can observe the calibration qualities. A diagonal corresponds to a perfect calibration. In contrast, our detector is overconfident if the line is below the diagonal and underconfident if it is above. Considering the TCP, we only examine the highest predicted probabilities. This results in a single calibration line. In the case of multiple classes, we will have one line per class. As we will see in Sect. 6, it is crucial not to average the respective metrics per bin since this might bias the curves towards the diagonal.

Calibration errors are aggregated values derived from calibration plots, allowing us to simplify the comparison of different models against each other. Correspondingly, we employ Eq. 8 and Eq. 9 to define the *Top-Label Calibration Error* (TCE) as

$$\mathrm{TCE}(\mathcal{E}) = \left(\sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{|\mathcal{E}|} (\mathrm{acc}(\mathcal{B}_m) - \mathrm{prob}(\mathcal{B}_m))^2 \right)^{1/2}. \tag{12}$$

The TCE is also referred to as the *Expected Calibration Error* in literature (Guo, Pleiss, Sun, and Weinberger, 2017). Similarly, we use Eq. 10 and Eq. 11 to define the *Marginal Calibration Error* (MCE) as

$$\mathrm{MCE}(\mathcal{E}) = \left(\sum_{k=1}^{K+1} \sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{|\mathcal{E}|} (\mathrm{acc}_k(\mathcal{B}_m) - \mathrm{prob}_k(\mathcal{B}_m))^2 \right)^{1/2} \quad (13)$$

In addition, we also report the recently proposed detection variants of TCP/TCE and MCP/MCE, which we refer to as dTCP/dTCE and dTCP/dMCE (Kuppers, Kronenberger, Shantia, and Haselhoff, 2020). Instead of the accuracy, they use the precision as an estimate for the true probability. We utilize an adjusted version of those metrics ignoring box positions to be able to compare them to the other plots and errors.

Entropy is a measure which can be interpreted as the uncertainty of a distribution and is defined by

$$\mathrm{H}(\hat{\boldsymbol{p}}) = -\sum_{k=1}^{K+1} \hat{p}_k \ln \hat{p}_k. \quad (14)$$

Similar to Ovadia et al. (2019), we compute it for all predictions in $\mathcal{E}$ and plot a histogram of entropy values to visualize the uncertainty of a model on a whole dataset. This way, we expect to observe differences resulting from data distributional changes. Since we typically have many background predictions with high probability per image, we consider only matched and missing predictions as defined by Eq. 5. This is because we intend to assess the uncertainty of predictions that were made for potential objects of unknown classes. We expect that these uncertainties are higher than the ones obtained from the in-distribution objects. Note, the entropy is the only metric we can apply to out-of-distribution data, since all other metrics require the same label space $\mathcal{Y}$.

6 Experiments

In this section, we introduce our experimental setup and discuss the results of the experiments that answer our research questions.

6.1 Setup

We focus on object detectors pretrained on the COCO dataset (Lin et al., 2014) and will evaluate their capability to provide well-calibrated probabilistic predictions as well as their ability to identify distributional changes. Besides using the 80 class test split of COCO, we want to be able to represent the calibration plots in a clear and understandable way and to investigate the dependence of calibration quality on the number of classes. Therefore, we additionally construct two subsets of COCO and fine-tune the last layers of detectors by using their respective standard hyperparameter settings for training. In particular, we use a simple subset called ANIMALS consisting of images with the two classes giraffe and elephant, and a slightly more complex subset TRAFFIC consisting of the ten classes person, bicycle, car, motorcycle, bus, train, truck, traffic light, fire hydrant, and stop sign. We refer to ALL when talking about the original 80 class COCO test split. Note, as we consider the additional background class in each detector $f^{\boldsymbol{\omega}}$, we have at least a three-class classification problem that needs to be solved.

Generally, a distributional shift occurs when the test sample distribution no longer matches the training sample distribution. To simulate distributional shifts within our evaluation, we follow the works of Ovadia et al. (2019) and Harakeh and Waslander (2021) and adapt the distribution $p(\boldsymbol{x})$ while keeping the distribution $p(y|\boldsymbol{x}, \boldsymbol{b})$ approximately unchanged. More precisely, we construct a new dataset with approximately the same class distribution as our respective training dataset but with a different sample distribution. To realize this, we create two subsets of the Open-Images object detection dataset (Kuznetsova et al., 2020) with the same set of classes as in ANIMALS and TRAFFIC and evaluate them in an identical setting. Like Harakeh and Waslander (2021), we assume that these constructed datasets can be interpreted as a shifted version of the training dataset due to a different data collection process (i.e., image quality, sources, and difficulty).

The evaluation on OOD data is commonly done in classification (Huseljic, Sick, Herde, and Kottke, 2021) as it gives insights about the quality of uncertainty estimates of a model (e.g., probabilistic outputs or derived measures). The idea behind an OOD evaluation in an object detection setting, however, is fairly uncommon (Du, Wang, Cai, and Li, 2022). We argue that a detector should be able to identify object-like entities in OOD images while returning high uncertainty for them. For example, consider a flock of sheep crossing a road and a detector in an autonomous vehicle trained on traffic classes such as cars or pedestrians. Without returning highly uncertain predictions for the unknown objects (i.e., sheep), we would not be able to identify this situation and only predict the background class with a high probability. Optimally, in such a scenario, our detector should return high uncertainty for all predicted boxes. We evaluate the detectors on out-of-distribution data similar to Harakeh and Waslander (2021). Thus, we require a dataset that has not only a different sample distribution $p(\boldsymbol{x})$ but also an unknown class distribution $p(y|\boldsymbol{x}, \boldsymbol{b})$. Accordingly, we take a subset from Open Images containing images in which none of the 80 classes from COCO appear. More details to the corresponding datasets, experimental setup, experiments, additional results as well as the implementation of our framework can be found in our implementation.

6.2 Results

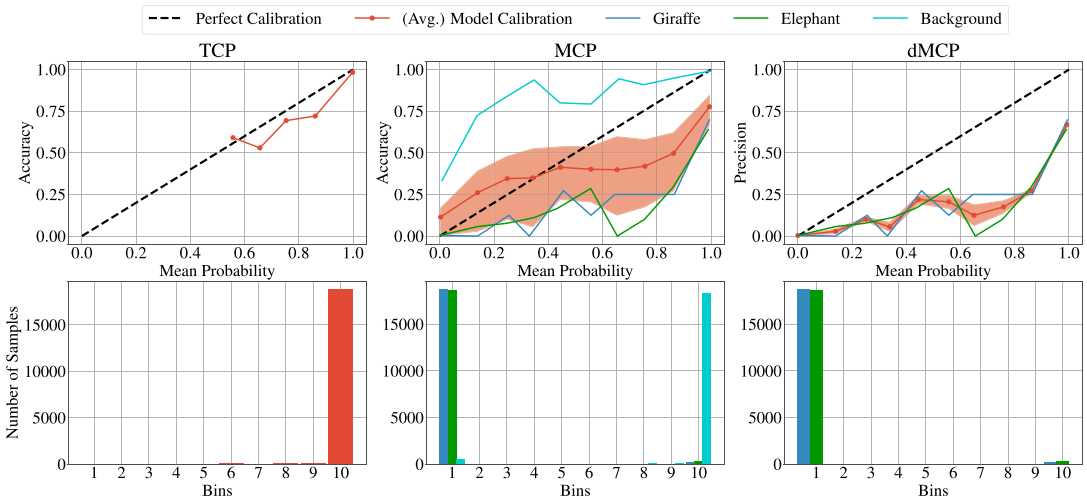
In the following, we present the results from numerous conducted experiments to sequentially address our research questions. We report values of all numerical metrics (i.e., generalization and calibration) for ANIMALS, TRAFFIC, and ALL, and the corresponding shifted versions in Table 1. Additionally, we also show calibration plots (Fig. 3) for the ANIMALS subset and entropy histograms (Fig. 4) for the OOD subset. We will utilize these to underline the answers to research questions with intuitive explanations. For additional calibration plots, we refer to our implementation.

Which metrics should be used to evaluate multi-class object detection architectures regarding their calibration and how should these metrics be applied? *When evaluating object detection architectures in a multi-class environment, it is essential to assess all predicted probabilities (instead of the maximum one) of our detector by using the MCP and MCE.* An example of this is given in the calibration plots shown in Fig. 3 on the ANIMALS subset of COCO. The comparison between TCP and MCP demonstrates for all detectors that although their highest predicted probability seems to be fairly well-calibrated, the rest of the predicted probabilities is not. For example, the predictions for the two object classes, giraffe and elephant, are overconfident for DETR and FASTER- RCNN with post-processing as their predicted probability lies below the diagonal. Vice versa, when looking at the background class, we can observe underconfidence. Such an interdependency between the object classes and the background class seems reasonable. In object detection, the neural network needs to learn an extremely imbalanced classification problem between background and objects. Therefore, it makes sense that this potentially biases the quality of our probabilistic predictions either towards under- or overconfidence. Furthermore, as shown by the red curves in Fig. 3, it is important to not average the bin metrics across classes as this might lead to biased results improving the calibration performance. To further highlight the importance of a suitable metric, we also report the dMCP, which we can see on the right in Fig. 3. Using precision instead of accuracy, we cannot capture the interdependency between background and objects. This is because the background predictions are ignored by only considering TPs and FPs for evaluation. Accordingly, we observe in Table 1 that the MCE reports the highest error compared to TCE and dMCE. Note the importance of the number of predictions in a bin when examining the MCP. Comparing only the upper plots of the MCP, FASTER- RCNN with post-processing appears to be better calibrated than DETR. However, it also has many predictions in bins between 0 and 1, leading to a higher error, as seen in Table 1. In summary, it is essential to consider the MCP and MCE to evaluate the multi-class calibration qualities of detectors as they consider TP, FP, TN, and FN.

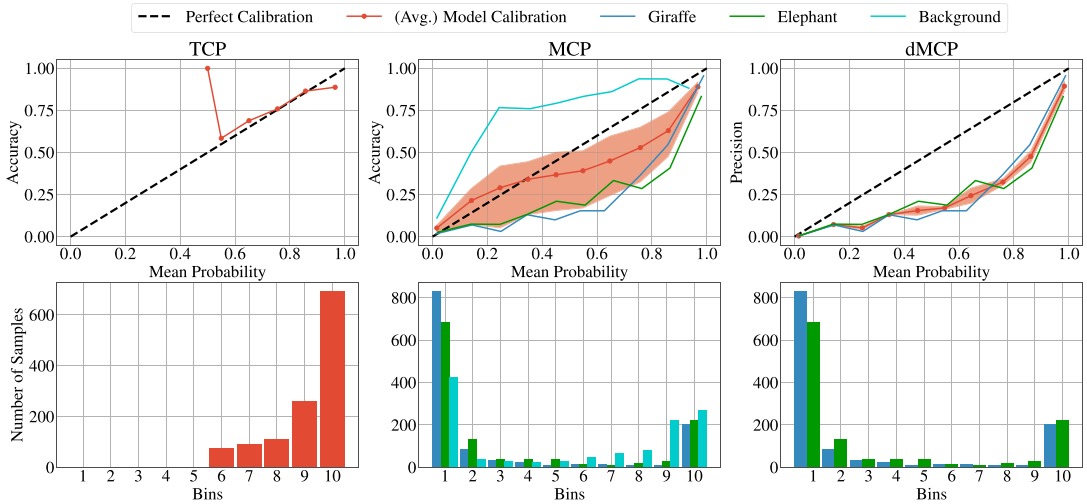
How do post-processing steps in a detection pipeline influence the calibration and do architectures that avoid them deliver better calibrated predictions? *The post-processing steps in a detection pipeline significantly influence a model's calibration quality.* We can see an example of this by comparing the calibration plots of FASTER-RCNN with (Fig. 3b) and without (Fig. 3c) post-processing. When using post-processing steps, the probabilities for object classes (i.e., giraffe and elephant) are overconfident, and the background class probabilities are underconfident. Without post-processing, however, this property no longer holds. There, the probabilities for object classes become underconfident, whereas the probabilities for the background class become overconfident. This means that post-processing steps of FASTER- RCNN modify the neural network's probabilistic predictions such that underconfidence of object classes changes to overconfidence. Table 1 also shows that we get much better calibration properties on all datasets from our neural network when avoiding post-processing. Again, we see that the averages across classes (red curves) would not be sufficient to identify this property.

Furthermore, we see that architectures avoiding post-processing steps, such as DETR, provide better calibrated probabilistic predictions. Table 1 demonstrates for all datasets that the proper scoring rules (NLL and BS) and calibration errors (TCE, MCE, and dMCE) of DETR are below the ones of FASTER- RCNN with post-processing. When looking at the calibration plots, this looks surprising as FASTER- RCNN's mean probabilities seem to be closer to the diagonal. However, its predictions are more spread across bins, weighing calibration errors in the middle higher. DETR's predictions, on the other hand, are concentrated near the edges, which results in overall lower errors. These properties are often referred to in the literature as sharpness and reliability (Ovadia et al., 2019).

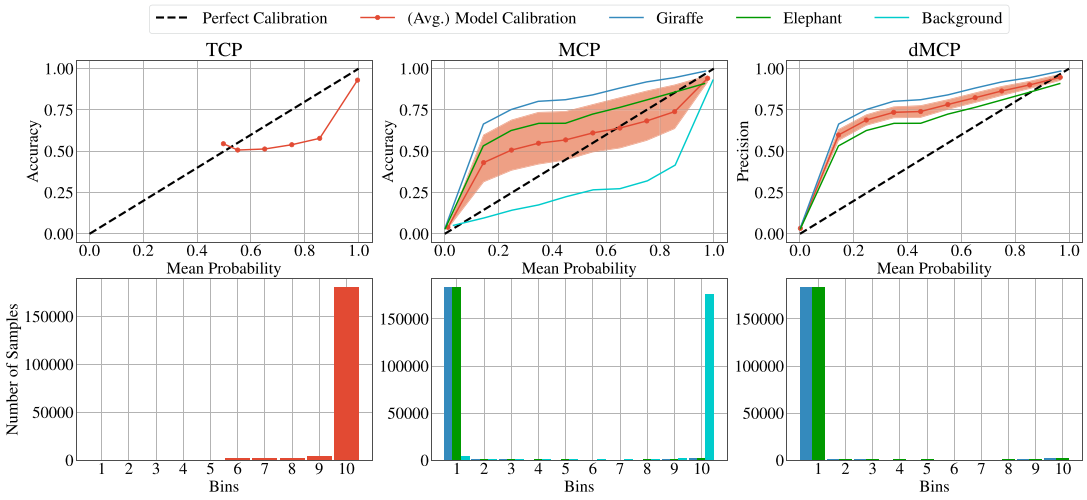
How well are the class probabilities of pretrained object detectors calibrated when the detectors are applied on samples from shifted versions of the training distribution? *Based on our experiments, the calibration qualities of the pretrained detectors on a dataset based on a shifted version of original the training sample distribution seem to depend on the difficulty of the object detection problem.* Generally, we would expect that the calibration quality worsens as we shift. However, it is also possible that it improves the detector calibration. For example, consider the three-class scenario of the ANIMALS subset (COCO vs. Open Images) in Table 1. Here, we observe that both DETR and FASTER-RCNN achieve a slightly better mAP, i.e., their performance is better on the shifted test dataset. We believe this quite surprising result is due to the fact that the shifted version of the ANIMALS subset contains images on which objects are easier to detect (approx. 50% bigger objects and fewer objects per image). Similarly, we also see that the calibra-



(a) DETR on ANIMALS subset.



(b) FASTER-RCNN on ANIMALS subset with post-processing.



(c) FASTER-RCNN on ANIMALS subset without post-processing.

Fig. 3 Calibration plots for the object detection architectures. The red shaded area shows the bin quartiles calculated across classes

Table 1 Results from DETR and FASTER- RCNN on in-distribution datasets and their respective shifted versions.

Dataset	Architecture	mAP(↑)	NLL(↓)	BS(↓)	TCE(↓)	MCE(↓)	dMCE(↓)
ANIMALS (In-Distribution)	DETR	**0.701**	**0.120**	**0.036**	**0.019**	**0.053**	**0.046**
	F- RCNN	0.639	0.549	0.268	0.061	0.140	0.122
	F- RCNN W.O. PO.	0.639	0.403	0.158	0.080	0.077	0.063
ANIMALS (Shifted)	DETR	**0.718**	**0.105**	**0.027**	**0.019**	**0.046**	**0.040**
	F- RCNN	0.648	0.534	0.255	0.058	0.112	0.095
	F- RCNN W.O. PO.	0.648	0.327	0.121	0.067	0.070	0.058
TRAFFIC (In-Distribution)	DETR	**0.488**	**0.209**	**0.095**	**0.036**	0.059	0.044
	F- RCNN	0.473	0.637	0.312	0.057	0.085	0.061
	F- RCNN W.O. PO.	0.473	0.360	0.155	0.065	**0.030**	**0.021**
TRAFFIC (Shifted)	DETR	0.381	0.371	0.163	0.067	0.099	0.073
	F- RCNN	0.387	0.878	0.434	0.135	0.150	0.108
	F- RCNN W.O. PO.	**0.387**	**0.335**	**0.156**	**0.056**	**0.046**	**0.035**
ALL (In-Distribution)	DETR	**0.420**	**0.380**	**0.172**	**0.046**	0.030	0.021
	F- RCNN	0.392	0.792	0.364	0.087	0.036	0.023
	F- RCNN W.O. PO.	0.392	0.510	0.213	0.077	**0.011**	**0.006**

Bold entries highlight the best performance for each dataset and metric
Arrows next to metrics indicate the direction of the optimal value. We abbreviate FASTER- RCNN as F- RCNN and "without post-processing" as W.O. PO

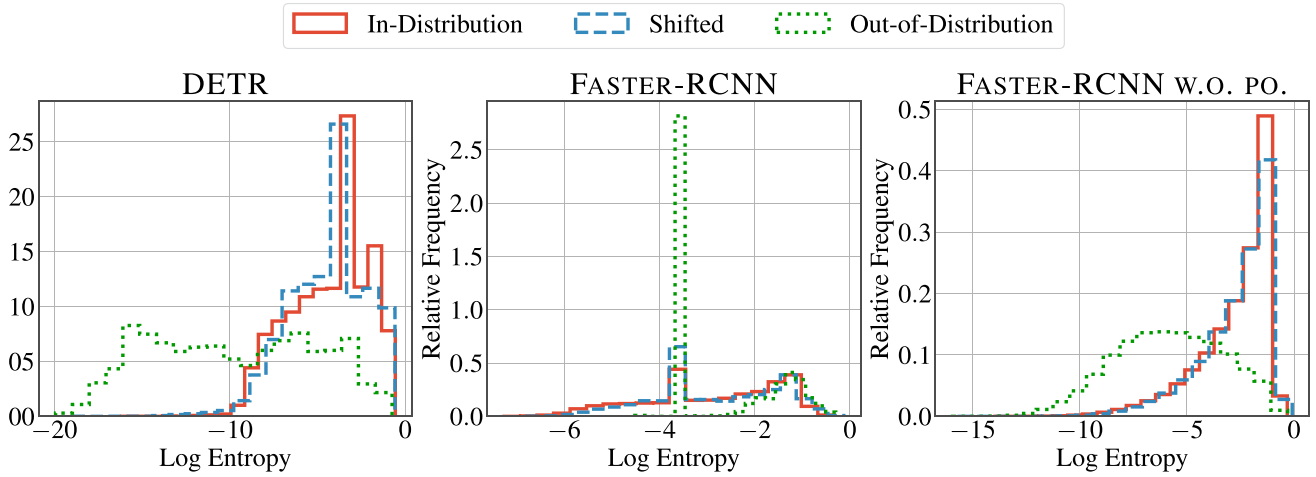


Fig. 4 Entropy histograms for in-distribution, shifted, and OOD datasets on the TRAFFIC subset

tion errors and proper scoring rules report better results for all detectors. Hence, we conclude that a shift of the sample distribution, which simplifies the problem, can lead to an improved calibration quality of our detections. Looking at the more complex problem TRAFFIC, we see that the mAP and the calibration metrics worsen when evaluating them under the shifted version. This is expected and matches the results from literature (Ovadia et al., 2019; Harakeh and Waslander, 2021). The entropy histograms in Fig. 4 show no noticeable difference between the in-distribution dataset of TRAFFIC and its shifted version for all architectures. Although the calibration qualities deteriorate, the detectors cannot identify this shift. Optimally, the worse calibration should be reflected by the dissimilarity of these histograms.

How well is a detector able to identify new objects in the case of out-of-distribution data by means of its probabilistic outputs? *Based on our experiments, pretrained object detectors cannot identify new objects through their probabilistic outputs as they only predict the background class with high probability.* To evaluate this, we visualize log entropy histograms for in-distribution, shifted, and out-of-distribution datasets in Fig. 4. Instead of the entropy, we use its logarithm to better demonstrate the differences in distributions better. We can see for DETR and FASTER- RCNN without post-processing that there are some distributional

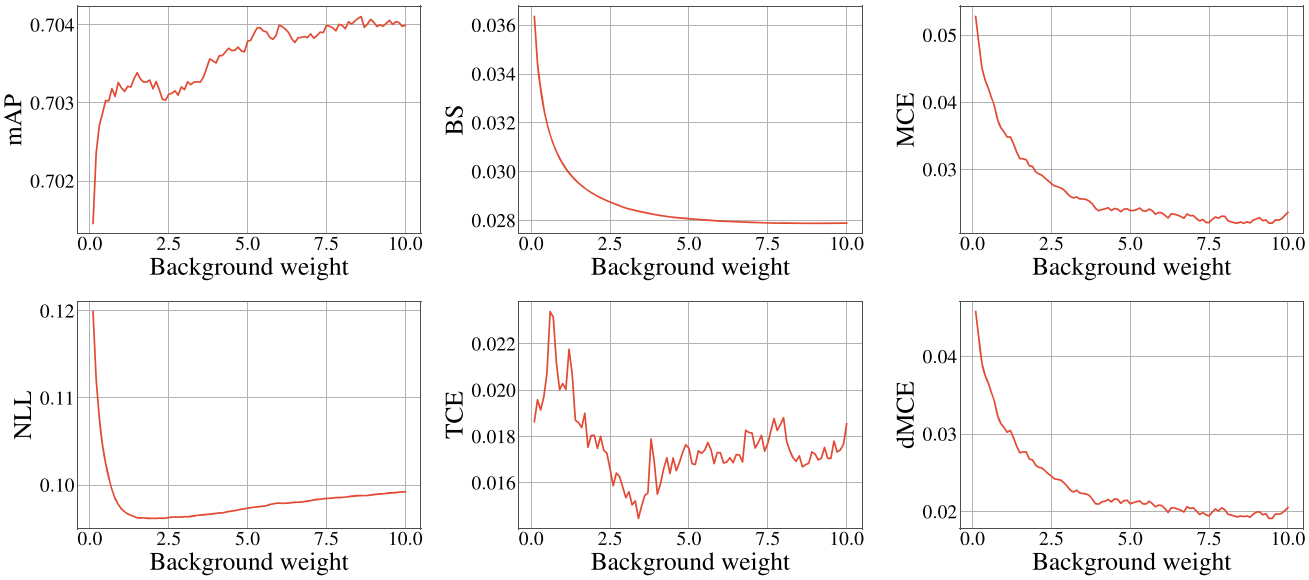


Fig. 5 Generalization and calibration metrics with respect to the background weight for the ANIMALS subset. Note the different scaling range of the respective metrics. The default background weight (0.1) is the minimum value on the x-axis

differences. However, these are due to the fact that the pre-trained detectors predict actual object classes (e.g., giraffe or elephant) for the in-distribution and shifted dataset. In contrast, for the OOD dataset, the detector solely predicts the background class with high probability. As a result, we notice that the predictions on the OOD dataset are even more overconfident when compared to the other histograms. Unfortunately, we cannot leverage this distributional difference to simply detect out-of-distribution samples. For instance, consider an autonomous vehicle in front of a flock of sheep. Since our detectors only predict the background class, we can not distinguish this situation from an empty road. Thus, we cannot identify out-of-distribution samples as detectors will just predict the background class.

7 Simple Recalibration

As mentioned in Sect. 6, there seems to exist some interdependency between object classes and the background class in a trained detector. While the predictions for the background class are underconfident, the ones for the object classes are overconfident. To investigate this and potentially improve a model's calibration, we conduct a case study on DETR with the ANIMALS and TRAFFIC subset in which we adjust its hyperparameter for the importance of the background class by changing its classification weight during fine-tuning. Specifically, as the background predictions were underconfident, we finetuned 100 object detectors with higher background weights and plot the results in Fig. 5 and Fig. 7. We can see that the model's calibration errors and proper scoring rules are improving by increasing the importance of the background class. Furthermore, we also note a slightly increasing

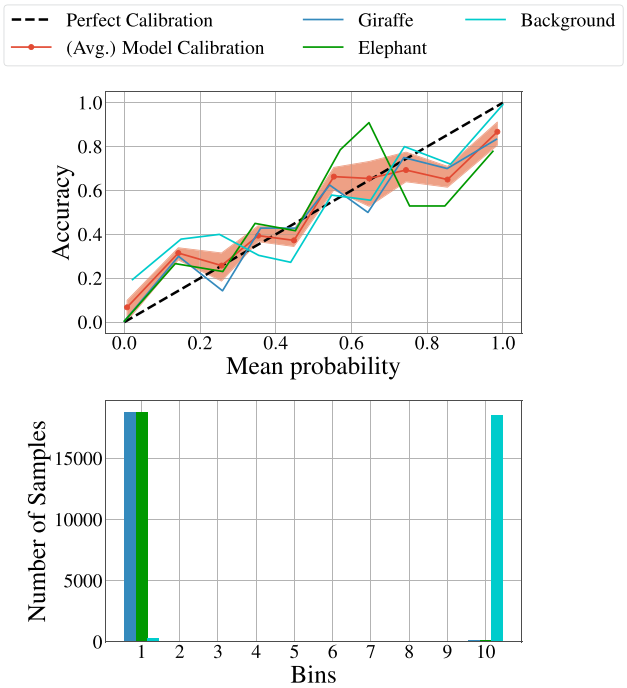


Fig. 6 The MCP for the detector that achieved the best MCE (value of 8.6 for the background weight) on the ANIMALS subset

generalization performance (mAP). Thus, it seems to be promising for calibration and generalization to raise the background class importance during fine-tuning of a specific detector. In contrast to the other metrics, the TCE does not capture this improvement, highlighting its inappropriateness for the calibration evaluation of multi-class problems again. Figure 6 shows the MCP from the detector trained on ANIMALS that achieved the best MCE in this study. Both object

and background predictions are closer to the diagonal compared to Fig. 3a while their interdependencies are no longer recognizable. We conclude that there seems to be a correlation between calibration quality and the imbalanced problem of object detection (i.e., background vs. objects) and leave further analyses regarding this for future work.

Limitations: Despite the promising results of increasing the importance of the background class to improve calibration, our case study and method have several limitations. First, our case study primarily considers a single object detector (DETR), which already includes a background weight that can be tuned. It might not be straightforward for other architectures (e.g., Faster-RCNN) as they typically do not include a background weight during optimization. Hence, the background class may not play an essential role. Additionally, our results on TRAFFIC indicate that there is an optimal background weight leading to the lowest calibration errors. This implies that we require post-hoc recalibration to obtain an optimal value. Consequently, in such a scenario, validating this using an additional calibration dataset is necessary. Finally, as we only consider a case study, we only employ two datasets. It is necessary to evaluate the effectiveness of our recalibration method on more datasets and investigate the influence of the background weight in a broader range of scenarios (e.g., in an OOD context).

8 Future Directions

For future work, we need to **extend the presented framework** such that it includes additional metrics (box-dependent calibration errors (Kuppers, Kronenberger, Shantia, and Haselhoff, 2020)) or we can evaluate additional object detection paradigms. Currently, we assume that the neural network predictions in the classification task describe the parameters of a categorical distribution. This assumption, however, does not always hold. For example, the architecture Retinanet, which is a one-stage paradigm, assumes that every prediction describes a two-class problem between object and background. Hence, the network uses the binary cross-entropy loss function during training. To evaluate such an architecture, we need to extend our framework.

The role of label assignment in our evaluation framework, while significant, has been approached with a simplified assumption. Our localization relies on a basic IOU metric with a fixed 0.5 threshold. While functional for this baseline evaluations, this presents limitations, potentially failing to capture the full scope of an object detectors performance. Therefore, future studies should **explore more nuanced label assignment processes**, perhaps integrating advanced metrics like generalized IOU or L1 distance. Aligning with this topic, there is a compelling need to investigate how the uncertainty of object detectors varies with different IOU thresholds and metrics. Such explorations could unveil intricate dynamics of object detectors, leading to more robust and accurate systems.

As seen in our experiments, post-processing steps such as NMS deteriorate the calibration quality of detectors. Therefore, it is vital to **research the dependency between post-processing and uncertainty modeling** if we want to use these architectures in tasks such as active learning. For example, it is hard to tell whether it makes sense to enhance an architecture with uncertainty modeling techniques such as Monte-Carlo Dropout Gal and Ghahramani (2016) if the outputs are discarded anyway. Therefore, we need to examine whether we can use the direct predictions of a neural network instead of the ones from the detection pipeline or develop more suitable post-processing steps. Additionally, it seems reasonable to just avoid post-processing architectures and use ones such as DETR. They seem promising as we can intuitively recalibrate the predictions of neural networks instead of a detection pipeline that might ignore half of the predictions.

Additionally, further research is needed that focuses on the **class-imbalance problem in object detection** (i.e., interdependency between calibration of object classes and background class) and its **influence on the uncertainty modeling** of a detector. In our experiments, we solely employed pre-trained detectors and finetuned them on specific subsets. However, it would be interesting to investigate proper scoring rules and calibration errors when training a detector from scratch with different hyperparameter settings. We assume that training the entire detector with a very high background weight would probably not lead to good generalization.

At last, we want to address the **missing ability of detectors to identify OOD objects**. We believe that it is vital for many tasks to distinguish between potential unknown objects and a natural background. With that addition, it would be possible, for instance, to improve the responses to different autonomous driving scenarios or perform better exploration during active learning (Herde, Huseljic, Sick, and Calma, 2021). The training of such an object detection architecture might be realized by using out-of-distribution data (Huseljic, Sick, Herde, and Kottke, 2021).

9 Conclusion

This work assessed different pretrained object detectors regarding their capability of modeling uncertainty considering various factors. First, we proposed a modular framework for evaluating calibration properties of object detection architectures in a multi-class setting while considering all actual and missing predictions of a detector. We analyzed the importance of the choice of metrics and concluded that, besides proper scoring rules, MCP and MCE are the most appro-

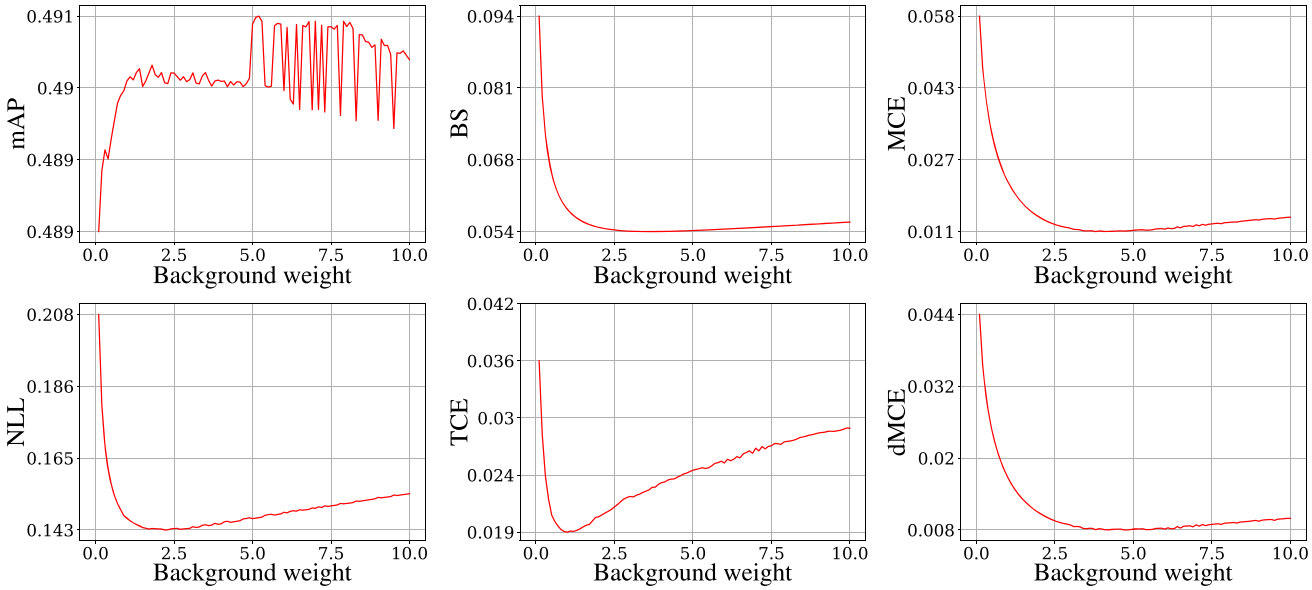


Fig. 7 Generalization and calibration metrics with respect to the background weight for the TRAFFIC subset. Note the different scaling ranges of the respective metrics. The default background weight (0.1) is the minimum value on the x-axis

priate metrics for evaluation. Subsequently, we investigated the influence of post-processing steps (e.g., NMS), which worsened the detector's calibration and revealed an interdependency between the confidences of the object class and background class predictions. We also evaluated the detectors' calibration on datasets with changed distributions. When considering a shift in the sample distribution, we noticed that the calibration does not necessarily worsen as the detection problem may get easier. Furthermore, we saw that detectors could not identify OOD objects when considering an OOD dataset as they only predicted the background class with high probability. Finally, based on the interdependency insights, we conducted a case study in which we demonstrated that DETR's calibration can be improved by simply increasing the importance weight of the background class in the objective function during training.

Author Contributions Denis Huseljic: Conceptualization Writing Investigation Visualizations Evaluation Methodology Experimental Evaluation Implementation Literature Research. Marek Herde: Conceptualization Review and Editing Investigation Evaluation Methodology Experimental Evaluation. Paul Hahn: Literature Research Implementation Experimental Evaluation. Mehmet Müjd: Literature Research Implementation Experimental Evaluation. Bernhard Sick: Conceptualization Review and Editing Evaluation Methodology

Funding Open Access funding enabled and organized by Projekt DEAL.

Data and Code Availability The datasets analyzed during the current study are all publicly available in the repositories reported in the references.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Bieshaar, M., Zernetsch, S., Hubert, A., Sick, B., & Doll, K. (2018). Cooperative starting movement detection of cyclists using convolutional neural networks and a boosted stacking ensemble. *IEEE Transactions on Intelligent Vehicles, 3*(4), 534–44.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European conference on computer vision* (pp. 213–229).

Du, X., Gozum, G., Ming, Y., & Li, Y. (2022). Siren: Shaping representations for detecting out-of-distribution objects. *Advances in Neural Information Processing Systems, 35*, 20434–20449.

Du, X., Wang, X., Gozum, G., & Li, Y. (2022). Unknown-aware object detection: Learning what you don't know from videos in the wild. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13678–13688).

Du, X., Wang, Z., Cai, M., & Li, Y. (2022). Vos: Learning what you don't know by virtual outlier synthesis. *International Conference on Learning Representations*,

Feng, D., Harakeh, A., Waslander, S. L., & Dietmayer, K. (2021). A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems, 23*(8), 9961–80.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *International conference on machine learning* (pp. 1050–1059).

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K.Q. (2017). On calibration of modern neural networks. *International conference on machine learning* (pp. 1321–1330).

Harakeh, A., & Waslander, S.L. (2021). Estimating and evaluating regression predictive uncertainty in deep object detectors. *International conference on learning representations.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 770–778).

Herde, M., Huseljic, D., Sick, B., & Calma, A. (2021). A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification. *IEEE Access, 9*, 166970–166989.

Huseljic, D., Sick, B., Herde, M., & Kottke, D. (2021). Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks. *International conference on pattern recognition* (pp. 9172–9179).

Kull M, Perello Nieto M, Kängsepp M, Silva Filho T, Song H, Flach P. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. Advances in neural information processing systems. 2019;32.

Kumar, A., Liang, P.S., & Ma, T. (2019). Verified uncertainty calibration. *Advances in Neural Information Processing Systems*, *32*

Küppers, F., Haselhoff, A., Kronenberger, J., & Schneider, J. (2022). Confidence calibration for object detection and segmentation. *Deep neural networks and data for automated driving: Robustness, uncertainty quantification, and insights towards safety* (pp. 225–250).

Kuppers, F., Kronenberger, J., Shantia, A., & Haselhoff, A. (2020). Multivariate confidence calibration for object detection. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops* (pp. 326–327).

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., et al. (2020). The open images dataset v4. *International Journal of Computer Vision, 128*(7), 1956–1981.

Liang, W., Xue, F., Liu, Y., Zhong, G., & Ming, A. (2023). Unknown sniffer for object detection: Don't turn a blind eye to unknown objects. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3230–3239).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, & D. Zitnick, C.L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision* (pp. 740–755).

Liu, Y.-C., Ma, C.-Y., Dai, X., Tian, J., Vajda, P., He, Z., & Kira, Z. (2022). Open-set semi-supervised object detection. *European conference on computer vision* (pp. 143–159).

Munir, M. A., Khan, M. H., Sarfraz, M., & Ali, M. (2022). Towards improving calibration in object detection under domain shift. *Advances in Neural Information Processing Systems, 35*, 38706–38718.

Neumann, L., Zisserman, A., & Vedaldi, A. (2018). Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. *Machine learning for intelligent transportation systems workshop at neurips.*

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, & S. Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, *32*, ,

Pathiraja, B., Gunawardhana, M., & Khan, M.H. (2023). Multiclass confidence and localization calibration for object detection. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 19734–19743).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*

Schwaiger, F., Henne, M., Küppers, F., Roza, F.S., Roscher, K., & Haselhoff, A. (2021). From black-box to white-box: Examining confidence calibration under different conditions. arXiv preprint arXiv:2101.02971,

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*

Wilson, S., Fischer, T., Dayoub, F., Miller, D., & Sünderhauf, N. (2023). Safe: Sensitivity-aware features for out-of-distribution object detection. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 23565–23576).

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*. https://github.com/facebookresearch/detectron2