Quantifying the Reliability of Predictions in Detection Transformers: Object-Level Calibration and Image-Level Uncertainty

Young-Jin Park*, Carson Sobolewski*, and Navid Azizan
Massachusetts Institute of Technology
{youngp,csobo,azizan}@mit.edu

arXiv:2412.01782v3 [cs.CV] 10 Dec 2025

***Abstract*—DEtection TRansformer (DETR) and its variants have emerged as promising architectures for object detection, offering an end-to-end prediction pipeline. In practice, however, DETRs generate hundreds of predictions that far outnumber the actual objects present in an image. This raises a critical question: which of these predictions could be trusted? This is particularly important for safety-critical applications, such as in autonomous vehicles. Addressing this concern, we provide empirical and theoretical evidence that predictions within the same image play distinct roles, resulting in varying reliability levels. Our analysis reveals that DETRs employ an optimal "specialist strategy": one prediction per object is trained to be well-calibrated, while the remaining predictions are trained to suppress their foreground confidence to near zero, even when maintaining accurate localization. We show that this strategy emerges as the loss-minimizing solution to the Hungarian matching algorithm, fundamentally shaping DETRs' outputs. While selecting the well-calibrated predictions is ideal, they are unidentifiable at inference time. This means that any post-processing algorithm—used to identify trustworthy predictions—poses a risk of outputting a set of predictions with mixed calibration levels. Therefore, practical deployment necessitates a joint evaluation of both the model's calibration quality and the effectiveness of the post-processing algorithm. However, we demonstrate that existing metrics like average precision and expected calibration error are inadequate for this task. To address this issue, we further introduce Object-level Calibration Error (OCE), which evaluates calibration by aggregating predictions per ground-truth object rather than per prediction. This object-centric design penalizes both retaining suppressed predictions and missed ground truth foreground objects, making OCE suitable for both evaluating models and identifying reliable prediction subsets. Finally, we present a post hoc uncertainty quantification (UQ) framework that predicts per-image model accuracy.**

I. Introduction

Object detection is an essential task in computer vision, with applications that span various domains including autonomous driving, warehousing, and medical image analysis. Existing object detection methods predominantly utilize convolutional neural networks (CNNs) [1]–[6] to identify and locate objects within images. More recently, DEtection TRansformer (DETR) [7] has revolutionized the field by utilizing a Transformer encoder-decoder architecture to offer a scalable end-to-end prediction pipeline, where the model predicts a set of bounding boxes and class probabilities. This paradigm shift has led to the exploration of various DETR variants, positioning them as potential foundation models for object detection tasks. While notable progress has been made, the reliability of these predictions remains under-investigated.

Quantifying the reliability of these models' predictions is critical across different scenarios. For instance, when building an auto-labeling system using DETRs, it is crucial to consider that the model may not always provide accurate predictions. Images where the model's reliability is questionable may need to be reviewed by human labelers. In safety-critical applications, such as autonomous vehicles, knowledge of the reliability of a prediction in the perception pipeline for a given scene is crucial for the downstream decision-making. Without properly understanding this reliability, the system may, for example, overlook a pedestrian, leading to catastrophically wrong decisions. However, owing to DETR's unique set-prediction mechanism, quantifying its reliability is not straightforward and remains largely underexplored.

In particular, DETR outputs a fixed number of predictions, typically in the hundreds; consequently, the central concern is *which predictions can be trusted and used*. Practitioners often employ heuristic approaches to select a subset of predictions, such as by setting a user-defined threshold to retain only a small number of high-confidence (e.g., 0.7) outputs, as seen in the official demo [**?**]. Similarly, previous studies exploring model reliability, such as [8]–[13], have also applied a user-defined confidence threshold (e.g., 0.3) to retain a subset of predictions for evaluating calibration quality, rather than using the entire set. On the other hand, the published implementations of several DETR variants select the top-k outputs (e.g., 100 out of 300 for Deformable-DETR [14] and 300 out of 900 for DINO [15]) based on confidence scores. Nonetheless, how different subset-selection schemes affect model reliability, as well as the process of choosing appropriate configurations for each scheme, remains under-explored, and their significance has yet to be fully understood.

To address this gap and move beyond heuristic subset selection, we first seek a fundamental characterization of DETRs' predictions based on the training objective used. Our theoretical analysis shows that the Hungarian loss incentivizes a "specialist strategy": having a single well-calibrated prediction (which we refer to as the *primary* prediction) to represent each ground-truth object, while suppressing all remaining predictions (referred to as the *secondary* predictions). This specialist strategy presents a fundamental challenge for prac-

*Contributed equally and share co-first authorship.

tical deployment: while those primary predictions are trained to be reliable, they are unidentifiable at test time. This means that any post-processing algorithm poses a risk of outputting a set of predictions with mixed calibration levels. Consequently, evaluating model reliability requires jointly assessing the model's calibration quality and the effectiveness of the post-processing algorithm in recovering the well-calibrated subset.

To facilitate this joint evaluation, we introduce **object-level calibration error (OCE)**, which evaluates calibration along ground-truth *objects* rather than *predictions* (Section V). OCE penalizes both missing ground truth objects and retaining unreliable secondary predictions, enabling proper assessment of model + post-processing combinations, something that common existing metrics are fundamentally incapable of. For instance, average precision (AP) [16]–[18] favors retaining all predictions, while expected calibration error (ECE) [8], [19], [20] favors discarding nearly all predictions to achieve artificially low error by ignoring missed objects.

Finally, we leverage both our theoretical characterization of DETRs' specialist strategy as well as OCE's ability to identify primary and secondary predictions and develop a method for **image-level uncertainty quantification**. More specifically, we first demonstrate that the foreground confidence contrast between primary and secondary predictions is strongly correlated with image-level reliability. This motivates our novel framework for quantifying image-level reliability by measuring the confidence contrast between the "positive predictions" retained by an OCE post-processing and the remaining "negative" ones. We conduct numerical experiments across in-distribution, near out-of-distribution, and far out-of-distribution scenarios, and show the effectiveness of our approach (Section VI).

In short, our key contributions are:

- We theoretically characterize and empirically validate the optimal strategy that DETR takes to minimize the Hungarian loss: it creates a single well-calibrated prediction per object while suppressing all others to near-zero foreground confidence while maintaining reasonable bounding box accuracy.
- We introduce OCE, a metric suitable for both evaluating model calibration and identifying reliable post-processing by measuring calibration per ground-truth object, avoiding pitfalls of existing metrics.
- We propose an algorithm to quantify image-level reliability by measuring the confidence contrast between positive and negative predictions identified by OCE. We demonstrate the effectiveness of our approach with numerical experiments on the `COCO` and `Cityscapes` datasets.

II. Related Work

Calibration. Model calibration refers to how well a model's predicted confidence scores align with the actual likelihood of those predictions being correct. In other words, low-confidence samples should exhibit low accuracy, and high-confidence samples should demonstrate high accuracy. For example, if among all instances to which the model assigns an 80% probability, the event occurs in about 80% of those instances, the model is well-calibrated. Achieving good calibration is crucial because it improves trust in the model's predictions, guides more informed decision-making, and enables better risk assessment in real-world applications.

To evaluate the alignment, one can measure the expected calibration error by binning predictions based on their confidence scores and computing the mean absolute error between the average confidence and the corresponding accuracy within each bin. However, in object detection tasks, measuring accuracy is not straightforward because predictions comprise both class probabilities and bounding boxes. Additionally, due to the set-prediction nature of object detection models, it is unclear which ground-truth object corresponds to each prediction. To address these challenges, detection expected calibration error (D-ECE) [8] defines precision as the accuracy metric and matches each prediction to a ground-truth object based on an intersection over union (IoU) threshold (commonly set at 0.5 or 0.75). On the other hand, localisation-aware expected calibration error (LaECE) [19] defines accuracy as the product of precision and IoU, thereby accounting for localization errors as well.

Yet, when model calibration is evaluated across the entire set of predictions, which includes the secondary predictions, the resulting evaluation becomes highly biased and therefore unreliable. To avoid this issue, previous studies have often measured D-ECE only on predictions with confidence scores exceeding 0.3. As detailed in later sections, applying a threshold of 0.3 selects the well-calibrated predictions in some DETR variants; thus, using such a fixed threshold may occasionally be acceptable. However, the optimal threshold is not guaranteed to be always 0.3 for other models, like UP-DETR, which introduces potential risks. In response, our paper proposes OCE, which measures the model's calibration quality alongside the employed post-processing scheme and thus can adaptively identify the reliable subset.

Uncertainty Quantification. Several studies focus on out-of-distribution (OOD) identification in object detection models. For example, [22] proposes a built-in OOD detector to isolate OOD data for human review, including those of unknown and uncertain classes (i.e., epistemic but not aleatoric uncertainty), by modeling the distribution of training data and assessing whether samples belong to any of the training class distributions. [23] generates outlier data from class-conditional distribution estimations derived from in-distribution data, training the model to assign high OOD scores to this generated data and low OOD scores to the original in-distribution data. Similarly, [19] employs an auxiliary detection model capable of expressing its confidence. Other works, including [24] and [25], investigate the latent representations generated by object detection models to identify the OOD nature of the input.

To the best of our knowledge, the aforementioned existing UQ techniques primarily focus on prediction-level analysis. Moreover, they predominantly address CNN-based models and explore the methodological way to better quantify the uncertainty in object detection models. In contrast, our paper emphasizes the significance of identifying a reliable subset within the entire set of predictions for uncertainty quantification,

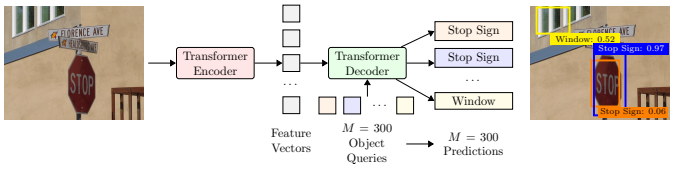


Fig. 1. A diagram of the DETR architecture. An input image is first processed through a CNN backbone to generate a 2D feature representation. This representation is then passed to the Transformer encoder, which extracts feature vectors. These feature vectors are fed to the decoder, which receives M learned object queries together. The decoder outputs M prediction sets, each containing a bounding box and corresponding class probabilities.

particularly in DETRs. Further, another novelty of our work lies in investigating an appropriate methodology to integrate different predictions' confidence estimates to quantify image-level reliability.

III. Preliminaries

A. Detection Transformer (DETR)

Consider a test image x and denote the set of ground-truth objects present in the image by $\mathcal{Y}$. Analogously, the set of predictions generated by DETR, parameterized by θ, is denoted by $\hat{\mathcal{Y}}_\theta(x)$. Each prediction $\hat{y}_i \in \hat{\mathcal{Y}}_\theta(x)$ is characterized by a bounding box $\hat{b}_i$ and an associated class probability distribution $\hat{p}_i$ over $\mathcal{C} \cup \{\emptyset\}$, where $\mathcal{C}$ is the set of foreground classes and $\emptyset$ represents the background class.

The structure of DETR is composed of two main components: the Transformer encoder, which extracts a collection of features from the given image; and the Transformer decoder, which uses these features to make predictions. In addition to the features extracted by the encoder, the decoder's input consists of M (typically several hundred) learnable embeddings, also known as *object queries*. Each decoder layer is composed of a self-attention module among object queries and a cross-attention module between each object query and the features. After processing the queries through several decoder layers, the model produces the M final representation vectors that are converted into bounding boxes and class labels via a shared feedforward network, f_ϕ. Together, these predictions form the final outputs, making DETR's predictions essentially an M-element set. See Figure 1 for an illustration.

B. Hungarian Loss in DETR

Let $\hat{\mathcal{Y}}_\theta(x) = \{\hat{y}_1, ..., \hat{y}_M\}$, where $\hat{y}_i = (\hat{p}_i, \hat{b}_i)$. This set is compared against $\mathcal{Y} = \{y_1, ..., y_N\}$, where N is the number of ground-truth objects. To evaluate the loss, a one-to-one correspondence between the two sets is first established. The ground-truth set is padded with $M - N$ background objects, and an optimal matching permutation $\hat{\sigma}$ is found by minimizing a pairwise matching cost $\mathcal{L}_{\text{match}}$ using the Hungarian algorithm [7], [32].

Once the optimal matching $\hat{\sigma}(\cdot)$ is determined, the actual training objective, the Hungarian Loss, is computed over these pairs. The loss for the entire set is the sum of the losses for each ground-truth object $y_j = (c_j, b_j)$—having the target class label c_j and bounding box b_j—and its matched prediction $\hat{y}_{\hat{\sigma}(j)}$:

$$\mathcal{L}(\hat{y}_{\hat{\sigma}(j)}, y_j) = \sum_{j=1}^{M} w_j \left[-\log \hat{p}_{\hat{\sigma}(j)}(c_j) + \mathbb{1}_{\{c_j \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_j, \hat{b}_{\hat{\sigma}(j)}) \right], \tag{1}$$

where $\hat{p}_i(c)$ is the predicted probability for class c of the i-th prediction. $\mathcal{L}_{\text{box}}$ is a linear combination of the IoU and L1 loss between the two bounding boxes. This process naturally partitions the M predictions into two groups for each image: the *optimal positive* predictions, which are matched to ground-truth foreground objects, and *optimal negative* predictions, which are matched to background objects. In practice, the log-probability term for negative predictions (where $c_j = \emptyset$) is down-weighted (e.g., by a factor of $w_\emptyset = 0.1$):

$$w_j = \begin{cases} w_\emptyset & \text{if } c_j = \emptyset \\ 1 & \text{if } c_j \neq \emptyset. \end{cases}$$

This structure, which penalizes high-confidence foreground predictions from negative predictions, forms the basis of our subsequent analysis.

IV. Optimal Prediction Strategies in DETR

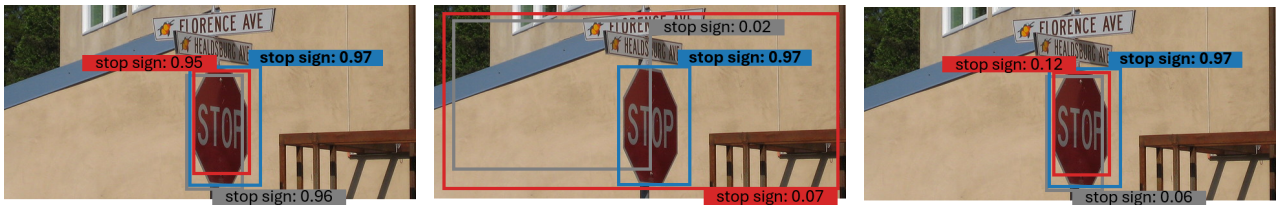
In this work, we evaluate a model's reliability using the concept of calibration. We define a prediction as perfectly calibrated if its predicted probability distribution matches the true posterior probability distribution of the object it is matched to. We formalize this by considering an object whose true, and potentially ambiguous, class identity is represented by a ground-truth probability distribution p_{gt}. A prediction is *perfectly calibrated* with respect to this object if its predicted distribution $\hat{p}$ exactly matches the true distribution p_{gt}, i.e.,

$$\hat{p}(c) = p_{\text{gt}}(c) \quad \forall c \in \mathcal{C} \cup \{\emptyset\}. \tag{2}$$

A. Motivation and Scope

During the inference stage at test-time, ground-truth annotations are unavailable, meaning the optimal positive predictions remain unknown. This raises the question of **Which of the predictions can be trusted and used?** If all of the model's predictions were generated independently and were well-calibrated (e.g., Figure 2a), the large number of predictions would not be a concern. We could simply apply well-known algorithms like NMS to remove duplicates and resolve redundancy.

However, empirical evidence suggests that such independence is unlikely to happen, as it appears to conflict with the minimization of the Hungarian loss in Equation (1). In practice, we observe that DETR tends to assign a well-calibrated confidence score to only a single prediction per object, while suppressing remaining predictions with low foreground confidence. Intriguingly, although the Hungarian loss imposes no penalty on the localization of these background-matched optimal negative predictions—implying they could potentially output random bounding boxes (Figure 2b)—we find that



(a) Accurate boxes. Calibrated. (b) Random boxes. Low Confidence. (c) Accurate boxes. Uncalibrated.

Fig. 2. DETR generates hundreds of predictions for each image, resulting in multiple predictions per object, with at least one (i.e., **blue boxes**) being well-calibrated. This figure illustrates how DETR handles the remaining predictions (i.e., **red** and **gray** boxes). DETR could (a) assign equally high confidence scores with accurate bounding boxes to all predictions; however, this strategy fails to optimize the Hungarian loss, as high-confidence unmatched predictions incur substantial penalties. Instead, DETR could assign low confidence to unmatched predictions, either with (b) random bounding boxes or (c) accurate bounding boxes. Our analysis reveals that DETR follows the third strategy, which explains the varying levels of reliability observed across predictions.

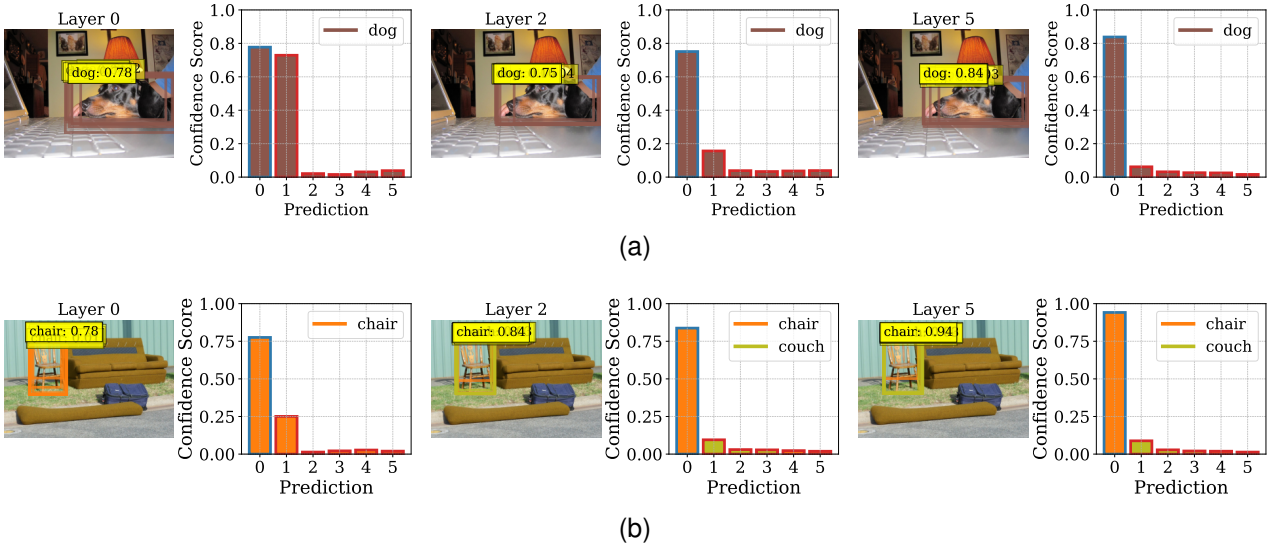


Fig. 3. Visualizations of the predictions generated by Cal-DETR. The optimal positive prediction (indexed by 0 and bordered in blue) and the five optimal negative predictions (indexed by 1-5 and bordered in red) with the largest IoU are presented. For each prediction and layer, the maximum confidence score and its corresponding label are visualized. The model's attention mechanism enables it to learn a "specialist strategy." When an object is confidently identified, a single primary query (the intended optimal positive prediction) is assigned a (high) calibrated confidence score. In parallel, all other remaining queries (the intended optimal negative predictions) have their confidence scores suppressed to near-zero, even while they maintain accurate bounding box predictions. This specialist strategy—where the model designates one query as the sole reliable predictor and the rest as background specialists—is demonstrably favored by the loss function over having multiple competing, moderately-confident predictions for the same object.

DETR exhibits a strong preference for accurate localization (Figure 2c), consistently producing precise boxes even for optimal negative predictions. In the following sections, we investigate this behavior in depth.

B. Exploring the Anatomy of DETR's Predictions

We begin our qualitative analysis by visualizing and examining the outputs generated from the DETR decoders. Since the Transformer decoder outputs only representation vectors, investigating their evolution across layers is not straightforward. We address this by reapplying the final feedforward network that operates on the last layer, f_ϕ, to the intermediate layers. This allows us to transform each representation vector into its associated bounding box and class label. This is feasible due to the alignment of intermediate representations, facilitated by residual connections between decoder layers [34]. Sample visualizations are in Figures 3 and 4.

In the first decoder layer, the model appears to explore the encoded image features, producing varied queries that result in various plausible predictions. In this early stage, the distinction between optimal positive and negative queries can be ambiguous (e.g., Figure 3a and Figure 4a). However, the attentions through the subsequent decoder layers progressively refine these predictions. By the final layer, the model selects a single prediction and assigns a confidence score based on its understanding of the image and the object. In contrast, the confidence scores for remaining queries do not increase to the same extent as the positives and even decrease. On the other hand, in images of low reliability (where the model is uncertain), the confidence score of the optimal positive prediction does not significantly increase, while the scores of the optimal negative ones are either slightly raised or unchanged. Based on this observation, we present our main claim:

DETR learns to assign multiple predictions for each object within a given image and to designate a single prediction that is calibrated, while the remaining predictions are suppressed to have low foreground confidence, even when maintaining

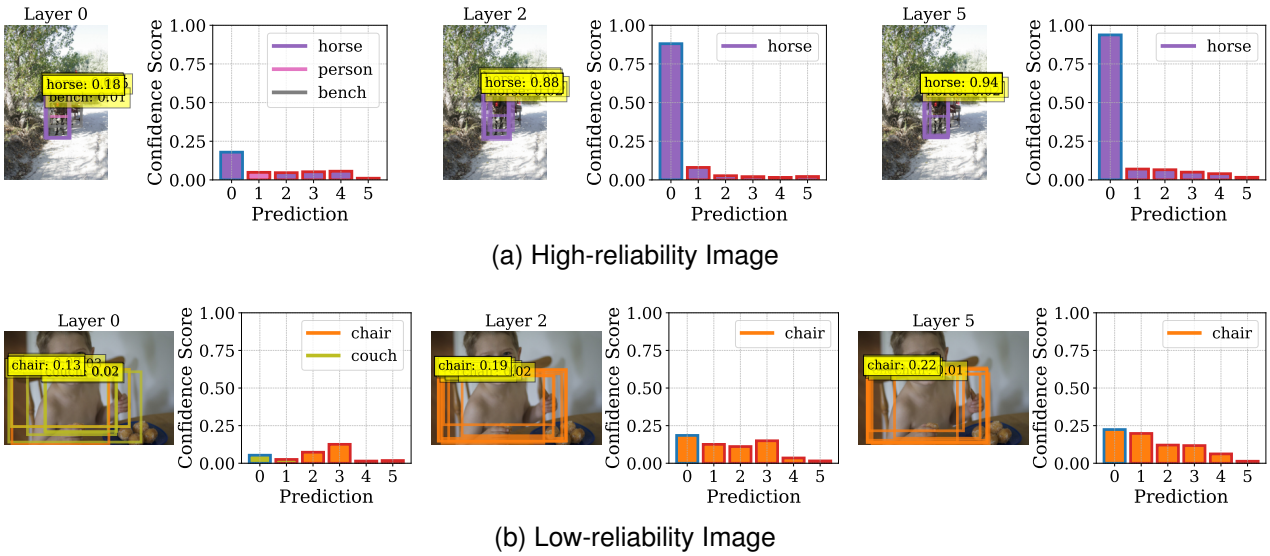


Fig. 4. When the model is confident, it assigns a high confidence score to the optimal positive prediction by either increases or remains high across the decoder layers via cross-attention mechanism—thus be calibrated. While those of remaining optimal negative queries' confidence scores are pushed to be near-zero while having accurate bounding box predictions. On the other hand, when the model is uncertain, DETR assigns a lower confidence score to the optimal positive prediction, thereby maintaining good calibration. However, **conversely**, it slightly increases or maintains the confidence scores for remaining predictions.

accurate localization. When the model's uncertainty about its prediction (and thus about the matching process) is high, the confidence separation between those predictions diminishes.

C. Theoretical Analysis of the Optimal Prediction Strategy

To provide theoretical support for the claim, we analyze the optimal prediction strategy in a stochastic setting. We define a random variable $y_{\text{gt}} = (c_{\text{gt}}, b_{\text{gt}})$ representing the ground-truth object with distribution Y, where $c_{\text{gt}} \in \mathcal{C}$ represents the class that is sampled from the class distribution p_{gt}, and b_{gt} represents the bounding box sampled from some distribution $\mathcal{B}$.

The padded set of M targets is denoted by $\{y_j\}_{j=1}^{M}$, where $y_1 = y_{\text{gt}}$ (the sampled object) and $y_j = \emptyset$ for $j \neq 1$. The model provides a set of predictions $\{\hat{y}_i\}_{i=1}^{M}$. The objective is to find predictions that minimize the expected total loss over the distribution Y:

$$\mathbb{E}\left[\mathcal{L}_{\text{total}}\right] = \mathbb{E}_{y_{\text{gt}}\sim Y}\left[\sum_{j=1}^{M} \mathcal{L}(\hat{y}_{\hat{\sigma}(j)}, y_j)\right].$$

To formalize the roles of the predictions, we define the index of the optimal positive query as $p = \hat{\sigma}(1)$, while the remaining $M-1$ indices, $\{n\}_{n\neq p}$, are the optimal negative queries. Given this definition, we can decompose the total loss into terms for the optimal positive and optimal negative indices:

$$\mathbb{E}\left[\mathcal{L}_{\text{total}}\right] = \mathbb{E}_{y_{\text{gt}}\sim Y}\left[\mathcal{L}(\hat{y}_p, y_{\text{gt}})\right] + \sum_{n\neq p} \mathcal{L}(\hat{y}_n, \emptyset).$$

The loss for the optimal positive prediction $\hat{y}_p = (\hat{p}_p, \hat{b}_p)$ matched to the y_{gt} is:

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\hat{y}_p, y_{\text{gt}})\right] &= \mathbb{E}_{y_{\text{gt}}\sim Y}\left[\mathcal{L}_{\text{box}}(\hat{b}_p, b_{\text{gt}}) - \log(\hat{p}_p(c_{\text{gt}}))\right] \\ &= \mathbb{E}_{b_{\text{gt}}\sim\mathcal{B}}\left[\mathcal{L}_{\text{box}}(\hat{b}_p, b_{\text{gt}})\right] - \sum_{c\in\mathcal{C}} p_{\text{gt}}(c)\log(\hat{p}_p(c)),\end{aligned} \tag{3}$$

where $\hat{p}_p(c)$ is the predicted probability for class c, as defined before. The loss for optimal negative prediction $\hat{y}_n$ matched to $\emptyset$ (one of the optimal negatives) is the down-weighted negative log-probability of the background class:

$$\mathcal{L}(\hat{y}_n, \emptyset) = -w_\emptyset \log(\hat{p}_n(\emptyset)). \tag{4}$$

Here, $w_\emptyset$ is the down-weighting factor (e.g., 0.1) for the background class. The following result establishes a lower bound on the expected total loss.

Proposition 1. *The minimum achievable expected total loss is lower bounded by the entropy of the ground-truth class distribution, i.e.,*

$$\mathbb{E}\left[\mathcal{L}_{total}\right] \geq \mathcal{H}(p_{gt}) + \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{box}(\hat{b}, b_{gt})],$$

where $\mathcal{H}(p_{gt}) = -\sum_{c\in\mathcal{C}} p_{gt}(c)\log(p_{gt}(c))$.

Proof. See Section B-A. □

Next, we characterize the optimal prediction set that uniquely achieves this bound.

Proposition 2 (Optimal Prediction Set)**.** *The unique globally optimal prediction set that minimizes* $\mathbb{E}\left[\mathcal{L}_{total}\right]$ *is* $\{(\hat{p}_i^*, \hat{b}_i^*)\}_{i=1}^{M}$*, where*

1) for the optimal positive prediction ($i = p$)*:* $\hat{p}_i^* = p_{gt}$ *(i.e.,* $\hat{p}_i^*(c) = p_{gt}(c)\ \forall c \in \mathcal{C}$ *and* $\hat{p}_i^*(\emptyset) = 0$*) and the box* $\hat{b}_i^*$ *minimizes* $\mathbb{E}[\mathcal{L}_{box}(\hat{b}_i, b_{gt})]$*, and*

2) for any optimal negative prediction ($i \neq p$): $\hat{p}_i^*(c) = 0$ $\forall c \in \mathcal{C}$ and $\hat{p}_i^*(\emptyset) = 1$.

Proof. See Section B-B. □

From Optimal Set to Strategy under Ambiguity. The previous propositions formally show that the Hungarian loss incentivizes a deterministic specialization. The optimal state is not a symmetric hedge where multiple queries compete, but for one query to emerge as a specialist that becomes a perfectly calibrated predictor of the object's true class distribution (p_{gt}). All other queries are simultaneously optimized to predict the background class, minimizing their own loss to zero.

This finding implies that the model is incentivized to develop an internal strategy to achieve this state. Ideally, the model would learn to designate a specific object query (which we conceptually refer to as the *primary* query) to become the optimal positive prediction p, and the remaining ones (*secondary queries*) to become the optimal negatives. As we discussed in Section IV-B, this is architecturally feasible, as DETR can leverage its attention mechanism to orchestrate predictions: Throughout the decoder layers, the model can identify the most promising object query and progressively specialize it, while simultaneously suppressing the others.

However, in practice, a critical ambiguity arises. While the Hungarian matching algorithm $\hat{\sigma}$ is deterministic for a deterministic annotation, the ground truth itself is often ambiguous. Consequently, the assignment of queries to targets becomes a stochastic process. We formalize this by defining P_i as the probability that the i-th query is matched to the ground-truth object:

$$P_i \triangleq \mathbb{P}_{y_{\text{gt}} \sim Y}(\hat{\sigma}(1) = i). \tag{5}$$

In this setting, the extreme specialization is now suboptimal because it incurs an infinite loss if the primary prediction is matched to the background. Therefore, to minimize its expected loss under this uncertainty, each query must find an optimal prediction $\hat{p}_i^*$ that "hedges" against this matching uncertainty. The expected loss J_i for query i is decomposed as:

$$J_i = \underbrace{P_i \cdot \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}_i, b_{\text{gt}}) \mid \hat{\sigma}(1) = i]}_{\text{Expected Box Loss Term}} \underbrace{-P_i \sum_{c \in \mathcal{C}} p_{\text{gt}}(c) \log(\hat{p}_i(c)) - (1 - P_i) w_\emptyset \log(\hat{p}_i(\emptyset))}_{\text{Expected Classification Objective } J(\hat{p}_i)}.$$

Minimizing J_i with respect to $\hat{p}_i$ is equivalent to minimizing the expected classification objective $J(\hat{p}_i)$. Proposition 3 characterizes the optimal strategy to minimize the objective.

Proposition 3 (Optimal Strategy under Uncertainty)**.** *For a query with a matching probability P_i to the ground-truth object whose class distribution is p_{gt}, the class prediction that minimizes the expected loss is $\hat{p}_i^*$, where*

1) the prediction for any foreground class $c \in \mathcal{C}$ is:

$$\hat{p}_i^*(c) = p_{fg}^*(P_i) p_{gt}(c). \tag{6}$$

2) the prediction for the background class is

$$\hat{p}_i^*(\emptyset) = 1 - p_{fg}^*(P_i),$$

where $p_{fg}^*(P_i)$ is the optimal foreground probability, defined as

$$p_{fg}^*(P_i) = \frac{P_i}{P_i + w_\emptyset(1 - P_i)}. \tag{7}$$

Proof. See Section B-C. □

Based on the result from Proposition 3, we analyze two characteristics in their prediction strategy:

- **Perfect Specialization:** Consider the scenario where the model is highly certain about the image context and thus about which query will be matched. In this scenario, the matching probability for the primary—which will eventually become the optimal positive—query index is perceived as 1 (i.e., $P_p \to 1$), while it vanishes for all others (i.e., $P_i \to 0$ for $i \neq p$). In this scenario, the strategy recovers the optimal prediction set (Proposition 2), where solely the p-th query becomes a perfectly calibrated predictor of the object (p_{gt}).
- **Hedging:** When the model is uncertain about the matching query (i.e., P_i is dispersed), it optimally "hedges." The foreground probability for the primary query is suppressed ($p_{\text{fg}}^*(P_p) < 1$), balancing the risk between foreground and background assignments.

In practice, matching probabilities are skewed: P_p is close to 1 for the primary query, and P_i is close to 0 for secondary queries: $i \neq p$. Thus, DETRs prioritize specialization while retaining a hedging mechanism proportional to their uncertainty. We term this the *Specialist Strategy*. This jives with the empirical trends observed in Figures 3 and 4.

D. Numerical Analysis

Setup. To provide quantitative support for our claim, we conducted experiments using four DETR variants: UP-DETR [35], Deformable-DETR (D-DETR) [14], Cal-DETR [12], and DINO [15]. We also investigate with Faster R-CNN [2] and YOLOv3 [**?**], [3]. Each model is trained on the `COCO` (train2017) dataset and we use $1,000$ images (i.e., 20%) of `COCO` (val2017) for the validation set.

Empirical Analysis of Prediction-Level Distributions. We now empirically analyze the distributions of both confidence and localization accuracy to validate our theoretical findings. Ideally, this analysis would compare the primary prediction (the query intended to be the optimal positive) against secondary predictions. However, since the model's internal designation is latent and not directly observable, we utilize the outcomes of the Hungarian matching as effective proxies. Specifically, we analyze optimal positive predictions (those matched to the ground truth) as the observable realization of primary prediction, and optimal negative predictions (those matched to the background) as the proxy for secondary predictions.

First, we investigate the confidence scores. We plot the empirical cumulative distribution function (ECDF) of the maximum foreground probability ($\max_{c \neq \emptyset} \hat{p}_i(c)$) in Figures 5a

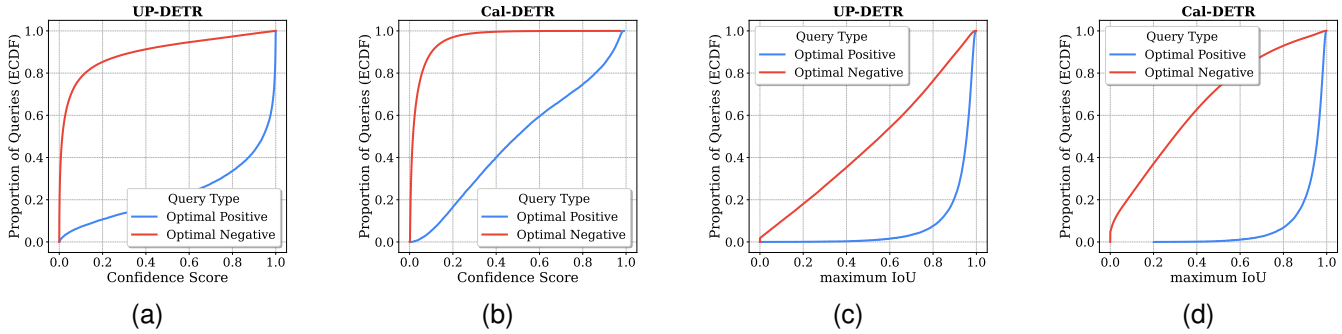


Fig. 5. Empirical cumulative distribution function (ECDF) plots for maximum confidence score over foreground classes and maximum IoU over ground truth objects, comparing optimal positive predictions (blue) and optimal negative predictions (red), in UP-DETR and Cal-DETR on the COCO dataset. (a, b) Confidence: The optimal negative prediction confidence is overwhelmingly clustered at zero. Optimal positive predictions span the full confidence range, acting as the system's calibrated predictors. (c, d) Localization: Optimal positive predictions are highly accurate (high IoU), which is the necessary condition for the model to designate a primary query. Optimal negative predictions maintain decent localization, which is not a failure but the optimal risk-averse strategy.

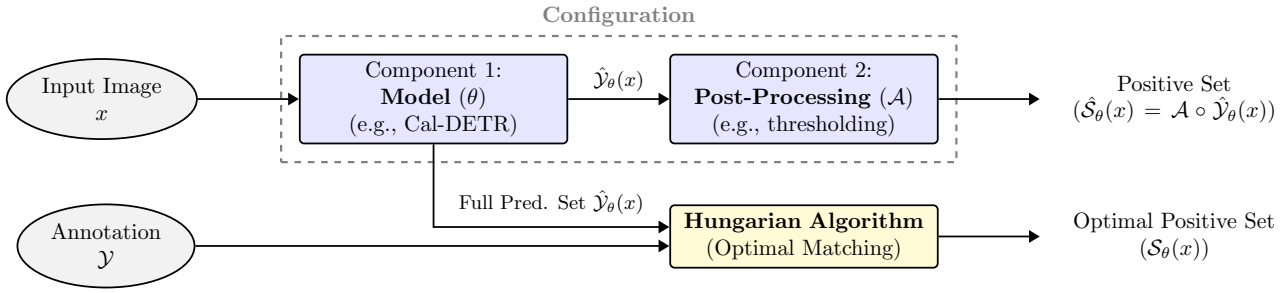


Fig. 6. Flowchart of our evaluation framework. Two components vary across configurations: (1) the model (θ), which generates a full set of predictions ($\hat{\mathcal{Y}}_\theta(x)$), and (2) the post-processing algorithm ($\mathcal{A}$), which selects a positive set ($\hat{\mathcal{S}}_\theta(x)$) from these predictions. In addition, we can obtain the optimal positive sets ($\mathcal{S}_\theta(x)$) by passing the full prediction set ($\hat{\mathcal{Y}}_\theta(x)$) and ground-truth object annotations ($\mathcal{Y}$) to the Hungarian Algorithm.

and 5b. This indicates that the optimal negative predictions' confidence scores are heavily concentrated in the near-zero regime, effectively being suppressed, in contrast to the widely dispersed confidence distribution observed for optimal positive predictions.

Next, we perform a parallel analysis on localization accuracy in Figures 5c and 5d. More specifically, for each prediction, we measure the maximum IoU with respect to the ground truth objects in the image. The results for optimal positive predictions confirm that the vast majority exhibit high accuracy. On the other hand, the analysis of optimal negative predictions reveals a critical distinction: despite their suppressed confidence, they frequently maintain substantial localization accuracy. It reflects a risk-averse strategy where the model maintains precise backup bounding boxes to mitigate the high penalty of potential matching errors, ensuring training stability. As shown in Figure 13, these observed trends are consistent across various DETR variants and other detectors.

We remark that this is not a model failure, but rather a risk-averse strategy induced by the loss landscape. In a stochastic setting where y_{gt} varies, there is always a non-zero probability that secondary queries get matched to the ground-truth object. If the secondary query predicts a random box, its matching to the foreground object elevates the expected box loss. Thus, to minimize the expected total loss, the model is incentivized to set secondary queries to predict a decent bounding box rather than a random box.

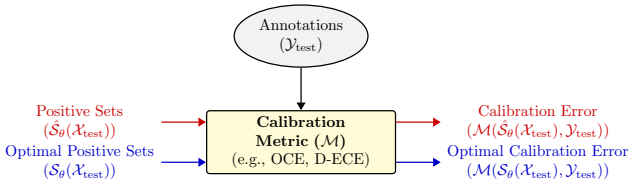


Fig. 7. Flowchart of our method for evaluating calibration error. A calibration metric ($\mathcal{M}$) compares either given positive sets ($\hat{\mathcal{S}}_\theta(\mathcal{X}_{\text{test}})$) or optimal positive sets ($\mathcal{S}_\theta(\mathcal{X}_{\text{test}})$) with ground-truth object annotations ($\mathcal{Y}_{\text{test}}$) to get calibration error ($\mathcal{M}(\hat{\mathcal{S}}_\theta(\mathcal{X}_{\text{test}}), \mathcal{Y}_{\text{test}})$) or optimal calibration error ($\mathcal{M}(\mathcal{S}_\theta(\mathcal{X}_{\text{test}}), \mathcal{Y}_{\text{test}})$), respectively.

V. Quantifying Calibration in DETR

A. Implications of the Specialist Strategy for Evaluation

Our analysis reveals that DETR's training objective incentivizes a specialist strategy. This strategy assigns two fundamentally different roles to the predictions:

1) A small subset of primary predictions (intended to be matched with foreground objects) is trained to be *calibrated predictors*, matching the true class distribution p_G.
2) The vast majority of secondary predictions are trained to be *background classifiers*, intentionally suppressing their foreground confidence to near-zero.

This specialist strategy introduces a fundamental ambiguity in downstream decision-making. When a prediction exhibits

low confidence, it becomes indistinguishable: is it a positive prediction correctly expressing high uncertainty about an ambiguous object, or is it a secondary prediction whose confidence has been artificially suppressed? This distinction is paramount, particularly in safety-critical domains.

Safety-Critical Implication

For an autonomous vehicle, a low-confidence "person" prediction could be a truly ambiguous object (like a person depicted on a billboard or on a vehicle), which the model is correctly uncertain about. However, it could also be a definite pedestrian identified by a secondary prediction, whose confidence was artificially suppressed to near-zero as part of the model's behavior, resulting from optimizing the Hungarian loss (as discussed in Section IV). In the latter case, relying on this suppressed score could lead the system to catastrophically misjudge the risk posed by a real person.

Ideally, at test time, we aim to filter out the secondary predictions and retain only the primary ones. However, identifying the primary predictions is non-trivial because the model's internal designation is latent. Consequently, practitioners rely on a post-processing algorithm (e.g., confidence thresholding) to select a subset of predictions serving as an estimate of the primary set.

While one might consider evaluating calibration directly on the optimal positive predictions, identifying this set requires ground-truth labels, making it infeasible at test time. Evaluating against this oracle set is useful for analysis, but it does not perfectly reflect the actual deployment conditions, where the system must rely solely on a post-processed subset.

These issues create a critical challenge for practical deployment: we must evaluate calibration quality using the same post-processing algorithm that will be applied at test time. Therefore, **practitioners should assess the entire (model + post-processing) pipeline jointly to identify configurations that produce reliable positive sets**, as illustrated in Figures 6 and 7, rather than evaluating the model in isolation. To address this, we propose a new calibration metric that takes as input any positive set produced by a post-processing algorithm, enabling such joint evaluation.

B. Proposed Metric: Object-Level Calibration Error

Notation. For each image x_l having N_l objects, we consider a set of ground-truth object annotations $\mathcal{Y}_l = \{y_{l,j} = (c_{l,j}, b_{l,j})\}_{j=1}^{N_l}$, and a set of DETR predictions: $\hat{\mathcal{Y}}_\theta(x_l) = \{\hat{y}_{l,i} = (\hat{p}_{l,i}, \hat{b}_{l,i})\}_{i=1}^{M}$ where M represents the number of object queries. Let $\mathcal{Y}_{\text{test}} = \big(\mathcal{Y}_l\big)_{l=1}^{N_{\text{test}}}$ denote the set of all ground-truth objects across all N_{test} test images $\mathcal{X}_{\text{test}} = \big(x_l\big)_{l=1}^{N_{\text{test}}}$.

Definition 1. Consider a subset of predictions, $\hat{\mathcal{S}}_\theta(x_l) \subseteq \hat{\mathcal{Y}}_\theta(x_l)$, that is generated by post-processing algorithm $\mathcal{A}$ from the full prediction set: $\hat{\mathcal{S}}_\theta = \mathcal{A} \circ \hat{\mathcal{Y}}_\theta$. We define an object-level calibration error (OCE) as the average Brier score per object:

$$\mathrm{OCE}_\tau\big(\hat{\mathcal{S}}_\theta, \mathcal{Y}_{\text{test}}\big) = \frac{1}{|\mathcal{Y}_{\text{test}}|} \sum_{y_{l,j} \in \mathcal{Y}_{\text{test}}} \mathrm{Brier}_\tau\big(\hat{\mathcal{S}}_\theta(x_l), y_{l,j}\big), \quad (8)$$

$$\mathrm{Brier}_\tau\big(\hat{\mathcal{S}}_\theta(x_l), y_{l,j}\big) = \sum_{c=1}^{C} \big(\mathbb{1}(c = c_{l,j}) - \bar{p}_{l,j}^{(\tau)}(c)\big)^2,$$

$$\bar{p}_{l,j}^{(\tau)}(c) = \frac{1}{|\mathcal{Q}_{l,j}^{(\tau)}|} \sum_{q \in \mathcal{Q}_{l,j}^{(\tau)}} \hat{p}_{l,i}(c),$$

where $\mathcal{Q}_{l,j}^{(\tau)}$ is a set of prediction indices that has bounding box overlap with the ground-truth object $y_{l,j}$:

$$\mathcal{Q}_{l,j}^{(\tau)} \triangleq \{i \mid \mathrm{IoU}(b_{l,j}, \hat{b}_{l,i}) \geq \tau\}.$$

When $|\mathcal{Q}_{l,j}^{(\tau)}| = 0$, we consider the predicted probability to be zero, thus the corresponding Brier score is estimated as 1.0. Following [18], [20], we use IoU thresholds of $\tau = 0.5, 0.75$ and report the average score as $\mathrm{OCE}(\cdot) = \mathrm{OCE}_{0.5}(\cdot) + \mathrm{OCE}_{0.75}(\cdot)$.

The introduced object-level calibration error (OCE) has two desirable characteristics derived from its object-centric design. Unlike prediction-level metrics (e.g., D-ECE) that average errors over the selected predictions, OCE averages errors over the ground-truth objects. Consequently, the metric effectively penalizes two critical failure modes in recovering the primary predictions. (1) Retaining secondary predictions: OCE penalizes subsets containing secondary predictions—those that maintain systematically low confidence despite accurate localization—as they fail to contribute correctly to the assigned objects. (2) Missing primary predictions: OCE penalizes subsets that are too aggressive and discard primary predictions, leaving ground-truth objects unmatched and incurring a maximum penalty. Thus, unlike existing prediction-level ECE metrics, OCE effectively measures how well the post-processed subset approximates the set of reliable predictions.

Remark (complexity). Computing OCE only requires IoU checks between GT boxes and predicted boxes; precomputing an $N_i \times M$ IoU matrix per image makes the additional cost negligible relative to standard evaluation.

C. Numerical Analysis

1) Validating OCE as a Calibration Metric: Our first step is to validate that OCE is a rightful measure of calibration. We first perform a sanity check to ensure OCE fundamentally measures the same concept of calibration as established metrics. To do this, we isolate the metrics' behaviors from the confounding effects of different post-processing decisions. We therefore choose the optimal positive set. This set best represents the predictions that each model was *specifically trained to produce as a calibrated predictor*.

Hypothesis. If OCE is a valid calibration metric, its ranking of models on this oracle optimal positive set should strongly correlate with the rankings produced by established metrics

Table 1. The Pearson correlation coefficient between the proposed OCE and established calibration metrics, D-ECE and LA-ECE$_0$, on the optimal positive predictions (*) across six models—UP-DETR, D-DETR, Cal-DETR, DINO, Faster R-CNN, and YOLOv3—and three data distributions. The consistently very strong correlations confirm that OCE effectively captures the same underlying concept of calibration quality as D-ECE and LA-ECE$_0$.

Predictions	Metrics	COCO (in-distribution)			Cityscapes (near OOD)			Foggy Cityscapes (OOD)			Average ± Std.
		D-ECE*	LA-ECE$_0^*$	OCE*	D-ECE*	LA-ECE$_0^*$	OCE*	D-ECE*	LA-ECE$_0^*$	OCE*	
Optimal Positives	D-ECE*	1.000	0.983	0.791	1.000	0.963	0.967	1.000	0.890	0.989	0.954 ± 0.066
	LA-ECE$_0^*$	0.983	1.000	0.859	0.963	1.000	0.873	0.890	1.000	0.854	0.936 ± 0.061
	OCE*	0.791	0.859	1.000	0.967	0.873	1.000	0.989	0.854	1.000	0.926 ± 0.076

(D-ECE, LA-ECE$_0$). A lack of correlation would suggest OCE is measuring a completely different, unrelated property.

Method. We evaluate the calibration errors on the optimal positive set from our six models—UP-DETR, D-DETR, Cal-DETR, DINO, Faster-RCNN, and YOLOv3—using OCE, D-ECE, and LA-ECE$_0$. We then evaluate the models using each metric and compute the Pearson correlation coefficient (PCC) between metrics.

As shown in the first three rows in Table 1, the correlations are all extremely high. This robust finding provides a critical validation: **OCE captures the same underlying concept of calibration quality as established metrics.** This confirms OCE is a legitimate calibration metric, providing the necessary foundation for the next section, where we will demonstrate its superiority in the practical scenario where optimal positive set is not accessible.

2) Pitfalls of Existing Metrics as a Reliable Set Selection Criteria: We now analyze the behavior of existing metrics and the proposed OCE in a practical scenario: selecting a confidence threshold (i.e., post-processing configuration) that achieves the lowest calibration error for a given model.

To this end, this section analyzes the effectiveness of existing calibration metrics (D-ECE, LA-ECE$_0$, and our OCE) as well as performance metrics (AP [16]–[18] and Localization recall precision (LRP) [21], [36]) for this task. Specifically, for each of the four different DETR models, we sweep the confidence score threshold from 0.0 to 1.0. This generates different positive subsets of predictions. We then plot the score of each metric for these subsets to determine the threshold that each metric prefers (i.e., the one that yields its optimal score). Results on Cal-DETR are illustrated in Figure 8. For definitions of metrics and additional results on other DETR models, please refer to the Appendix C. Our empirical findings reveal the following behaviors:

Pitfall 1: Average Precision Prefer No Filtering. As noted in several studies [19]–[21], [36], the optimal AP is achieved when the threshold is set close to 0.0. This is because AP does not harshly penalize low-confidence predictions. However, as discussed so far, using the full prediction set (hundreds of predictions) is practically non-sensical. It carries a high risk of including uncalibrated negative predictions, diminishes interpretability, and leads to unreliable downstream decisions.

Pitfall 2: ECEs Prefer Discarding Almost All Predictions. More importantly, the optimal ECEs (D-ECE and LA-ECE$_0$) are achieved when the threshold is set close to 1.0. This behavior, which favors retaining only a tiny subset of high-confidence predictions, is a structural pitfall of prediction-level

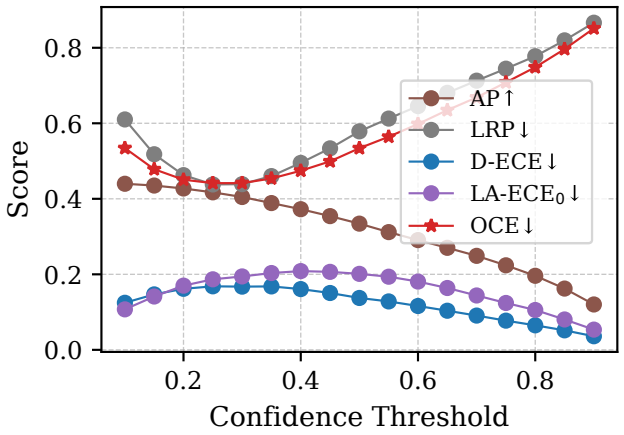


Fig. 8. Impact of confidence threshold selection on various performance and error metrics in Cal-DETR evaluated on COCO. Higher scores are preferable (↑), while lower scores are preferable (↓).

ECEs [20]. The cause is simple: ECEs do not penalize missed predictions (i.e., false negatives). As a result, they can achieve a near-zero (perfect) error by evaluating a set containing only a few highly accurate and confident predictions, while ignoring all the ground-truth objects the model missed at that high threshold.

Therefore, this pitfall makes ECEs fundamentally unsuitable for the goal we defined: finding a reliable (model + post-processing) combination. If a practitioner uses D-ECE to select a threshold, the metric will misleadingly guide them to select a threshold near 1.0, which is nonsensical as it discards almost all predictions to achieve a low error.

A Note on LRP: Good for Localization, Not Calibration. On the other hand, LRP, a localization-focused performance metric, does appear to find a sensible, non-extreme optimal threshold. This is rational: optimal positives are trained to be well-localized predictors, while optimal negatives are relatively not. LRP, by focusing purely on localization quality, thus correctly identifies a threshold that filters out the poorly-localized predictions. However, LRP is not designed to measure calibration and does not consider the confidence quality of the predictions at all. As a result, a model ranking scored by LRP reflects localization performance, which is not necessarily aligned with the model's confidence calibration qualities.

3) The effectiveness of OCE: The biggest advantage of using OCE over other metrics is that **OCE is capable of measuring the model's calibration quality alongside the post-processing scheme employed.** Resultingly, as illustrated



(a) Threshold Optimizing AP (b) Threshold Optimizing OCE (c) Threshold Optimizing D-ECE

Fig. 9. Exemplary visualization demonstrating the impact of confidence threshold (ρ) selection on the final prediction subset in Cal-DETR across different post-processing configurations. Optimal positive and negative predictions are highlighted with blue and red boxes, respectively. The threshold selected by AP (e.g., $\rho = 0.1$) retains redundant predictions, mixing calibrated positives with uncalibrated negatives. The threshold selected by D-ECE (e.g., $\rho = 0.9$) discards nearly all detections, including the optimal positive prediction with moderate confidence, resulting in a missed detection. The proposed OCE identifies a practical threshold (e.g., $\rho = 0.27$) that balances precision and recall while maintaining calibrated predictions.

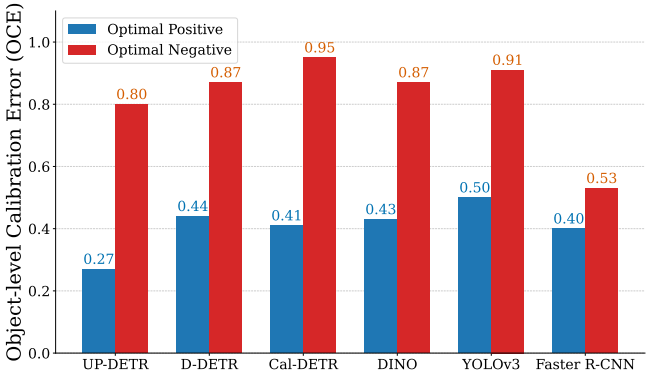


Fig. 10. A visualization of the difference in calibration between optimal positive and negative predictions on the COCO dataset. Object-level Calibration Error (OCE) value of each set is shown, where a lower score represents better-calibrated predictions. As shown, OCE varies significantly between optimal positive and negative predictions, indicating that OCE is a useful metric for assessing the reliability of post-processing schemes.

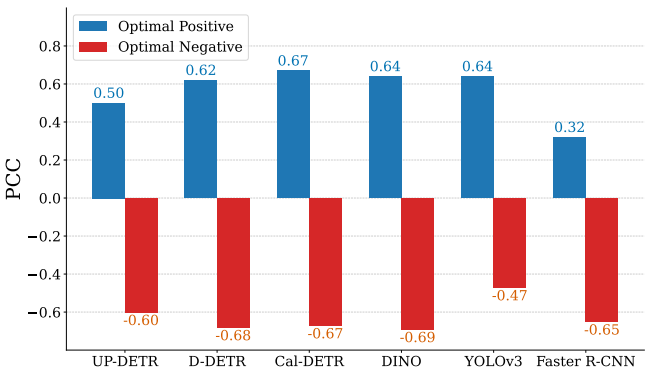


Fig. 11. Pearson Correlation Coefficient (PCC) between per-image AP scores and per-image average foreground confidence scores on optimal positive and negative predictions (higher values indicate stronger correlation). As expected, optimal positive predictions show high positive correlation between confidence and performance. More interestingly, optimal negative predictions exhibit an *inverse correlation* with image-level reliability.

in Figures 8 and 14, the confidence threshold vs. OCE curve is U-shaped and the optimal OCE is achieved around a threshold of 0.3. This is particularly notable because it reflects the practical choices in many calibration studies to date [8]–[13], confirming the reliability of using OCE.

Using a fixed separation threshold of 0.3 might serve as a reasonable approximation for distinguishing between optimal positives and negatives. However, trained or post-calibrated DETR models often have varying optimal confidence thresholds. For example, UP-DETR has an optimal threshold of approximately 0.5, and low-temperature calibrated models are likely to require even higher thresholds. Consequently, employing a fixed threshold poses potential risks.

VI. Quantifying Image-level Reliability

In this section, we revisit the importance of separation between primary and secondary predictions at test-time, now from the perspective of image-level reliability estimation.

Recall that our theoretical analysis (Proposition 3) shows that when the model is uncertain about its prediction—expressed by matching probability $P_i < 1$—it suppresses the predictive distribution. Reflecting this observation to the secondary prediction, we find that its foreground confidence score is no longer zero. Consequently, we expect the separation between these predictions to become blurred under uncertainty, and our empirical observations confirm this.

Specifically, for **reliable** images, we observe that the confidence of optimal positive predictions increases across decoder layers while optimal negative predictions remain low, resulting in a **large gap between optimal positive and negative predictions** (e.g., Figure 4a). In contrast, for **unreliable** instances, optimal positive prediction confidence fails to increase across layers, while optimal negative prediction scores are either slightly elevated or unchanged. This produces a **small gap between optimal positive and negative predictions** (e.g., Figure 4b).

To further empirically verify this, we first obtain the image-wise AP score by using the same COCO evaluator [18], passing the predictions and annotations for each image individually rather than the entire image set. We use this value as a proxy of image-level reliability (See Definition 2 for more formal interpretation.) Then, we measure the Pearson

correlation coefficient between the image-wise AP and the average confidence score of the optimal positives and the optimal negatives, respectively.

As shown in Figure 11, the average confidence scores of the optimal positive predictions exhibit a moderately strong positive correlation with image-level reliability (i.e., the model's actual average precision on that image) across different models. Notably, the average confidence scores for negative predictions are *inversely* correlated with image-level reliability. This finding reinforces our claim, highlighting the importance of identifying primary and secondary predictions, when conducting image-level uncertainty quantification (UQ) in DETR.

A. Image-Level Reliability

We introduce a formal definition of *image-level reliability* by examining the model's overall object detection performance on the image, extending [29], [33].

Definition 2. We define image-level reliability as a measure of how closely the predictions align with the ground-truth annotations for a given test image x:

$$\mathsf{ImReli}(x;\theta) \triangleq \mathsf{Perf}\big(\hat{\mathcal{Y}}_\theta(x),\ \mathcal{Y}\big),$$

where one can use any standard performance metric, such as average precision, for Perf depending on the user's requirements.

By its definition, image-level reliability directly addresses the model's applicability to a given test instance. However, since image-level reliability requires ground-truth annotations for its determination, obtaining it during inference is not feasible. Therefore, this paper's objective is to develop a method that assigns a quantitative score to each image instance, closely aligning with the model's image-level reliability.

B. Proposed Method: Quantifying Reliability by Contrasting

Based on our findings, we propose a post hoc UQ approach by contrasting the confidence scores of positives and negatives:

$$\begin{aligned}
\mathsf{ContrastiveConf}(x) &= \mathsf{Conf}^+(x) - \lambda \mathsf{Conf}^-(x),\\
\mathsf{Conf}^+(x) &= \frac{1}{|\hat{\mathcal{S}}_\theta(x)|}\sum\nolimits_{(\hat{p},\hat{b})\in\hat{\mathcal{S}}_\theta(x)} \max_{c\in\mathcal{C}} \hat{p}(c),\\
\mathsf{Conf}^-(x) &= \frac{1}{|\hat{\mathcal{S}}^-_\theta(x)|}\sum\nolimits_{(\hat{p},\hat{b})\in\hat{\mathcal{S}}^-_\theta(x)} \max_{c\in\mathcal{C}} \hat{p}(c),
\end{aligned}$$

where $\hat{\mathcal{S}}_\theta(x)$ and $\hat{\mathcal{S}}^-_\theta(x) \triangleq \hat{\mathcal{Y}}_\theta(x) \setminus \hat{\mathcal{S}}_\theta(x)$ are predicted sets, which we refer to *positive* and *negative* predictions, respectively. λ is a scaling factor that can be determined on the validation set; we include an ablation study of the scaling factor in the subsequent section. Our results demonstrate that the proposed method is robust to the choice of scaling factor and consistently outperforms baseline approaches, achieving the robust performance with a scaling factor of 5.0 - 10.0.

Selecting positives at test time. Ground-truth matches are unavailable at test time, so we approximate the optimal positive set $\mathcal{S}_\theta(x)$ by applying a post-processing operator $\mathcal{A}$ (e.g., confidence thresholding, Top-k, or NMS). We choose $\mathcal{A}^*$ on the validation set to *minimize* OCE:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}}\ \mathrm{OCE}\big(\mathcal{A}\circ\hat{\mathcal{Y}}_\theta(\mathcal{X}_{\text{val}}),\ \mathcal{Y}_{\text{val}}\big),$$

where $\mathcal{Y}_{\text{val}}$ is the set of all ground-truth objects across the images $\mathcal{X}_{\text{val}}$ in validation dataset.

C. Numerical Analysis

The model is tested on three datasets with varying levels of out-of-distribution (OOD) characteristics: `COCO` (in-distribution), `Cityscapes` (near OOD), and `Foggy Cityscapes` (OOD).

We compare different methods for quantifying per-image reliability based on different separation methods. For separation methods, we apply the optimal matching (as a reference) as well as the practical post-processing schemes such as fixed and adaptive confidence thresholding. We measure the Pearson correlation coefficient of each method with the ImReli computed using the standard AP metric.

Table 2 shows that the proposed ContrastiveConf consistently achieves the best correlation with image-level reliability. Interestingly, the absolute value of Conf^- often surpasses that of Conf^+ when a non-optimal separation scheme is applied. This observation highlights how our contrasting approach, which leverages the strengths of both Conf^+ and Conf^-, achieves robust performance across diverse settings.

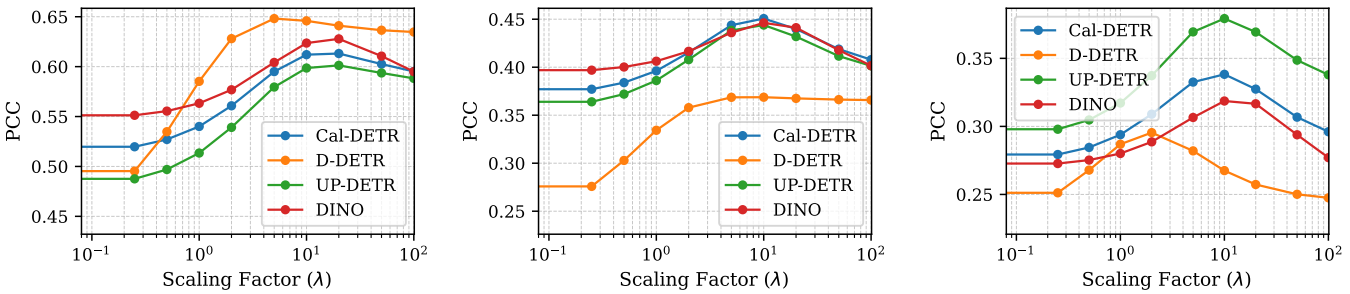
In addition, we conduct an ablation study on the scaling factor (λ), with results presented in Figure 12. The study reveals that the best performing scaling factor for `COCO` dataset lies between 5.0 and 10.0. Notably, this range remains effective even for out-of-distribution datasets such as `Cityscapes` and `Foggy Cityscapes`. Furthermore, the results demonstrate that ContrastiveConf with $1.0 \leq \lambda \leq 10.0$ consistently outperforms Conf^+ (i.e., $\lambda = 0.0$, the leftmost point on each line). That said, we also observe a rapid drop in performance when the scaling factor is excessively large, particularly on the OOD dataset, `Foggy Cityscapes`. Thus, we do not recommend using large scaling factors or relying solely on the negative predictions.

VII. Conclusion

The main contribution of our work is an in-depth analysis of the reliability of DETR frameworks. In particular, we reveal that DETR's predictions for a given image exhibit varying calibration quality, highlighting the importance of identifying well-calibrated predictions. To address this challenge, we introduced a systematic framework that leverages our object-level calibration error metric to discern these reliable predictions effectively. Furthermore, we proposed a novel uncertainty quantification method for estimating image-level reliability in DETR and conducted comparative studies of various post-processing schemes regarding their impact on DETR's reliability. We hope our efforts expand the scope of DETR applications by enabling more precise and reliable deployment, sparking further research into the area of reliable positive prediction identification.

Table 2. Comparison of the proposed method (ContrastiveConf) with baseline methods on their Pearson correlation coefficient with the image-level reliability in Equation (2), with different models. Perf is evaluated using AP. For each model, we apply the optimal matching, the standard post-processing scheme (Top-100 and confidence thresholding by 0.3), or the proposed OCE-based post-processing scheme, as indicated within the parentheses. The strongest correlations are highlighted in bold, while negative correlations appear in red. First, the proposed contrasting method, ContrastiveConf, achieves the highest correlation with image-level reliability. In addition, it is noteworthy that the average confidence of negative predictions (Conf^-) exhibits a negative correlation with image-level reliability, in contrast to the positive correlation observed when using positive predictions (Conf^+).

	Methods	COCO (in-distribution)				Cityscapes (near OOD)				Foggy Cityscapes (OOD)			
		UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO
Oracles	Conf^+ (Optimal)	0.503	0.618	0.670	0.635	0.555	0.647	0.649	0.633	0.561	0.642	0.662	0.634
	Conf^- (Optimal)	-0.608	-0.584	-0.572	-0.586	-0.293	-0.296	-0.330	-0.311	-0.177	-0.235	-0.212	-0.196
	ContrastiveConf (Optimal)	**0.700**	**0.648**	**0.684**	**0.662**	**0.601**	**0.659**	**0.656**	**0.644**	**0.612**	**0.660**	**0.667**	**0.647**
Baselines	Conf^+ (Top-100)	-0.619	-0.603	-0.581	-0.601	-0.409	-0.374	-0.372	-0.391	-0.262	-0.270	-0.239	-0.229
	Conf^+ (0.3)	0.476	0.464	0.539	0.504	0.229	0.353	0.385	0.399	0.195	0.256	0.258	0.258
Ours	Conf^+ (OCE)	0.477	0.478	0.547	0.512	0.252	0.330	0.393	0.370	0.191	0.276	0.270	0.274
	Conf^- (OCE)	-0.640	-0.575	-0.571	-0.585	-0.349	-0.382	-0.379	-0.394	-0.256	-0.322	-0.253	-0.283
	ContrastiveConf (OCE)	**0.656**	**0.588**	**0.628**	**0.613**	**0.359**	**0.407**	**0.441**	**0.440**	**0.275**	**0.350**	**0.317**	**0.327**



(a) COCO (b) Cityscapes (c) Foggy Cityscapes

Fig. 12. Impact of the scaling factor (λ) on image-level UQ performance of ContrastiveConf (OCE). Pearson correlation coefficient (PCC) using various scaling factors is reported on (a) In-distribution (COCO), (b) Near OOD (Cityscapes), and (c) Far OOD (Foggy Cityscapes). The optimal scaling factor lies within the range of 5.0 to 10.0, while this range generalizes well across out-of-distribution datasets. Furthermore, it shows the efficacy of ContrastiveConf over Conf^+ (i.e., ContrastiveConf with $\lambda = 0.0$).

Acknowledgments

The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing computing resources that have contributed to the results reported within this paper. This work was supported in part by the MIT Summer Research Program (MSRP), the MIT-Google Program for Computing Innovation, the MIT-Amazon Science Hub, the MIT-IBM Watson AI Lab, and MathWorks.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[2] S. Ren, K. He, R. Girshick, and J. Sun, " Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 06, pp. 1137–1149, Jun. 2017.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[5] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, Z. Yuan, and P. Luo, "Sparse r-cnn: An end-to-end framework for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15 650–15 664, 2023.

[6] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021.

[7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[8] F. Kuppers, J. Kronenberger, A. Shantia, and A. Haselhoff, "Multivariate confidence calibration for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 326–327.

[9] M. A. Munir, M. H. Khan, M. Sarfraz, and M. Ali, "Towards improving calibration in object detection under domain shift," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 706–38 718, 2022.

[10] M. A. Munir, M. H. Khan, S. Khan, and F. S. Khan, "Bridging precision and confidence: A train-time loss for calibrating object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 474–11 483.

[11] B. Pathiraja, M. Gunawardhana, and M. H. Khan, "Multiclass confidence and localization calibration for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 734–19 743.

[12] M. A. Munir, S. H. Khan, M. H. Khan, M. Ali, and F. Shahbaz Khan, "Cal-detr: calibrated detection transformer," *Advances in neural information processing systems*, vol. 36, 2024.

[13] M. A. Munir, M. H. Khan, M. S. Sarfraz, and M. Ali, "Domain adaptive object detection via balancing between self-training and adversarial learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 353–14 365, 2023.

[14] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[15] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[16] G. Salton, "Introduction to modern information retrieval," *McGrawHill Book Co*, 1983.

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisser-

man, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.
[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
[19] K. Oksuz, T. Joy, and P. K. Dokania, "Towards building self-aware object detectors via reliable uncertainty quantification and calibration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9263–9274.
[20] S. Kuzucu, K. Oksuz, J. Sadeghi, and P. K. Dokania, "On calibration of object detectors: Pitfalls, evaluation and baselines," in *European Conference on Computer Vision*. Springer, 2025, pp. 185–204.
[21] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Localization recall precision (lrp): A new performance metric for object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 504–519.
[22] R. Li, C. Zhang, H. Zhou, C. Shi, and Y. Luo, "Out-of-distribution identification: Let detector tell which i am not sure," in *European Conference on Computer Vision*. Springer, 2022, pp. 638–654.
[23] X. Du, Z. Wang, M. Cai, and Y. Li, "Vos: Learning what you don't know by virtual outlier synthesis," *arXiv preprint arXiv:2202.01197*, 2022.
[24] X. Du, G. Gozum, Y. Ming, and Y. Li, "Siren: Shaping representations for detecting out-of-distribution objects," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 434–20 449, 2022.
[25] S. Wilson, T. Fischer, F. Dayoub, D. Miller, and N. Sünderhauf, "Safe: Sensitivity-aware features for out-of-distribution object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 565–23 576.
[26] A. Shelmanov, E. Tsymbalov, D. Puzyrev, K. Fedyanin, A. Panchenko, and M. Panov, "How certain is your transformer?" in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 1833–1840.
[27] A. Sharma, N. Azizan, and M. Pavone, "Sketching curvature for efficient out-of-distribution detection for deep neural networks," in *Uncertainty in artificial intelligence*. PMLR, 2021, pp. 1958–1967.
[28] A. Vazhentsev, G. Kuzmin, A. Shelmanov, A. Tsvigun, E. Tsymbalov, K. Fedyanin, M. Panov, A. Panchenko, G. Gusev, M. Burtsev *et al.*, "Uncertainty estimation of transformer predictions for misclassification detection," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 8237–8252.
[29] Y.-J. Park, H. Wang, S. Ardeshir, and N. Azizan, "Quantifying representation reliability in self-supervised learning models," *arXiv preprint arXiv:2306.00206*, 2023.
[30] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.
[32] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[33] S. Ardeshir and N. Azizan, "Embedding reliability: On the predictability of downstream performance," in *NeurIPS ML Safety Workshop*, 2022.
[34] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, and P. He, "Dola: Decoding by contrasting layers improves factuality in large language models," *arXiv preprint arXiv:2309.03883*, 2023.
[35] Z. Dai, B. Cai, Y. Lin, and J. Chen, "Up-detr: Unsupervised pre-training for object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1601–1610.
[36] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "One metric to measure them all: Localisation recall precision (lrp) for evaluating visual detection tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9446–9463, 2022.



Young-Jin Park is a Ph.D. candidate at the Massachusetts Institute of Technology in LIDS and MechE, specializing in uncertainty quantification and reliability assessment of deep learning models. He received his B.S. and M.S. degrees in Aerospace Engineering from KAIST in 2017 and 2019, respectively. Prior to his doctoral studies, Young-Jin worked as a research engineer at NAVER AI Lab from 2019 to 2022, where he developed production-scale machine learning systems including recommender systems and demand forecasting models. His research interests span machine learning, representation learning, uncertainty quantification, and trustworthy AI systems with a particular focus on understanding and improving the reliability of neural network predictions for safety-critical applications. Young-Jin's work has been recognized with the Wunsch Foundation Silent Hoist and Crane Award for excellence in graduate studies from the Department of MechE at MIT in 2024, the M.S. Outstanding Paper Award from the Department of Aerospace Engineering at KAIST in 2019, and the KAIST Presidential Fellowship awarded to the top ten students of the Class of 2017.



Carson Sobolewski is currently an S.M. student at the Massachusetts Institute of Technology (MIT) in the Laboratory for Information and Decision Systems (LIDS) and the Department of Aeronautics and Astronautics (AeroAstro). Previously, he received his B.S. degree in Computer Engineering from the University of Florida. In 2024, he interned at MIT, where he contributed to this project. His research interests include uncertainty quantification and safe autonomous systems.



Navid Azizan is currently the Esther & Harold E. Edgerton Career Development Assistant Professor at MIT in LIDS, MechE & IDSS. He obtained his PhD in Computing and Mathematical Sciences (CMS) from the California Institute of Technology (Caltech) in 2020, his MSc in electrical engineering from the University of Southern California in 2015, and his BSc in electrical engineering with a minor in physics from Sharif University of Technology in 2013. Prior to joining MIT, he completed a postdoc in the Autonomous Systems Laboratory (ASL) at Stanford University in 2021. Additionally, he was a research scientist intern at Google DeepMind in 2019. His research interests broadly lie in machine learning, systems and control, mathematical optimization, and network science. His work has been recognized by several awards, including research awards from Google, Amazon, and MathWorks, the 2020 Information Theory and Applications (ITA) "Sun" (Gold) Graduation Award, and the 2016 ACM GREENMETRICS Best Student Paper Award. He was named to the list of Outstanding Academic Leaders in Data by CDO Magazine in 2024 and 2023, named an Amazon Fellow in AI in 2017, and a PIMCO Fellow in Data Science in 2018. He was also the first-place winner and a gold medalist at the 2008 National Physics Olympiad in Iran.

APPENDIX A
DATASET

For our experiments, we used the `Cityscapes` and `Foggy Cityscapes` datasets, which each have 500 images of first-person driving footage in realistic environments. `Foggy Cityscapes` has the same base images as `Cityscapes`, but with fog simulated and added to create a further out of distribution set. Since the DETR models were trained on `COCO`, the `Cityscapes` and `Foggy Cityscapes` annotations were converted to correspond to the labels of `COCO`. More specifically, the person, bicycle, car, motorcycle, bus, train, and truck classes were transferred directly. In addition, the rider class of `Cityscapes` and `Foggy Cityscapes` was mapped to the person class of `COCO`. The other classes present in `Cityscapes` and `Foggy Cityscapes` are largely focused on image segmentation, and thus were omitted (e.g. building, sky, sidewalk). The pre-trained model weights were obtained from their respective official implementations.

APPENDIX B
DETAILED THEORETICAL PROOFS

A. Proof of Proposition 1

Proposition 1. *The minimum achievable expected total loss is lower bounded by the entropy of the ground-truth class distribution, i.e.,*

$$\mathbb{E}\left[\mathcal{L}_{total}\right] \geq \mathcal{H}(p_{gt}) + \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{box}(\hat{b}, b_{gt})],$$

where $\mathcal{H}(p_{gt}) = -\sum_{c\in\mathcal{C}} p_{gt}(c)\log(p_{gt}(c))$.

Proof. The expected total loss is the sum of the expected loss for the optimal positive query matched to y_{gt} (denoted as index p) and the expected losses for the $M-1$ remaining queries matched to $\emptyset$ (denoted as indices $\{n \neq p\}$):

$$\mathbb{E}[\mathcal{L}_{\text{total}}] = \mathbb{E}[\mathcal{L}(\hat{y}_p, y_{\text{gt}})] + \sum_{n\neq p} \mathcal{L}(\hat{y}_n, \emptyset).$$

From Equation 4, $\mathcal{L}(\hat{y}_n, \emptyset) = -w_\emptyset \log(\hat{p}_n(\emptyset))$; and since $\log(\hat{p}_n(\emptyset)) \leq 0$: $\mathcal{L}(\hat{y}_n, \emptyset) \geq 0$. Therefore, the total expected loss is bounded below by:

$$\mathbb{E}[\mathcal{L}_{\text{total}}] \geq \mathbb{E}[\mathcal{L}(\hat{y}_p, y_{\text{gt}})].$$

From Equation 3, we have:

$$\mathbb{E}[\mathcal{L}(\hat{y}_p, y_{\text{gt}})] = \underbrace{-\sum_{c\in\mathcal{C}} p_{\text{gt}}(c)\log(\hat{p}_p(c))}_{\mathbb{E}[\mathcal{L}_{\text{cls}}]} + \mathbb{E}_{b_{\text{gt}}\sim\mathcal{B}}[\mathcal{L}_{\text{box}}(\hat{b}_p, b_{\text{gt}})].$$

We minimize the two terms on the right-hand side independently. First, we find the minimum classification loss $\mathbb{E}[\mathcal{L}_{\text{cls}}]$. We define $S = \sum_{c\in\mathcal{C}} \hat{p}_p(c) = 1 - \hat{p}_p(\emptyset)$ and $\hat{q}_p(c) = \hat{p}_p(c)/S$. Then:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{cls}}] &= -\sum_{c\in\mathcal{C}} p_{\text{gt}}(c)(\log(S) + \log(\hat{q}_p(c))) \\ &= -\log(S) + \text{CE}(p_{\text{gt}}, \hat{q}_p),\end{aligned}$$

where $\text{CE}(p_{\text{gt}}, \hat{q}_p) = \sum_{c\in\mathcal{C}} p_{\text{gt}}(c)\log(\hat{q}_p(c))$ is the cross-entropy. By Gibbs' inequality, $\text{CE}(p_{\text{gt}}, \hat{q}_p) \geq \mathcal{H}(p_{\text{gt}})$ and equality holds if and only if $\hat{q}_p = p_{\text{gt}}$. Additionally, $-\log(S) \geq 0$ and equality holds if and only if $S = 1$ (i.e., $\hat{p}_p(\emptyset) = 0$).

Denoting the minimum box loss as $\min_{\hat{b}} \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}, b_{\text{gt}})]$, a lower bound of $\mathbb{E}_{b_{\text{gt}}\sim\mathcal{B}}[\mathcal{L}_{\text{box}}(\hat{b}_p, b_{\text{gt}})]$, we have:

$$\mathbb{E}[\mathcal{L}_{\text{total}}] \geq \mathcal{H}(p_{\text{gt}}) + \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}, b_{\text{gt}})].$$

□

B. Proof of Proposition 2

Proposition 2 (Optimal Prediction Set)**.** *The unique globally optimal prediction set that minimizes* $\mathbb{E}\left[\mathcal{L}_{total}\right]$ *is* $\{(\hat{p}_i^*, \hat{b}_i^*)\}_{i=1}^M$*, where*

1) *for the optimal positive prediction* ($i = p$)*:* $\hat{p}_i^* = p_{gt}$ *(i.e.,* $\hat{p}_i^*(c) = p_{gt}(c)\ \forall c \in \mathcal{C}$ *and* $\hat{p}_i^*(\emptyset) = 0$*) and the box* $\hat{b}_i^*$ *minimizes* $\mathbb{E}[\mathcal{L}_{box}(\hat{b}_i, b_{gt})]$*, and*
2) *for any optimal negative prediction* ($i \neq p$)*:* $\hat{p}_i^*(c) = 0$ $\forall c \in \mathcal{C}$ *and* $\hat{p}_i^*(\emptyset) = 1$.

Proof. We evaluate the expected total loss for the proposed strategy $\{(\hat{p}_i^*, \hat{b}_i^*)\}_{i=1}^M$:

1) For the optimal positive prediction: The strategy sets $\hat{p}_p^* = p_{\text{gt}}$ and $\hat{b}_p^*$ to minimize the expected box loss. The resulting expected loss is:
$$\begin{aligned}\mathbb{E}[\mathcal{L}(\hat{y}_p^*, y_{\text{gt}})] &= \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}, b_{\text{gt}})] - \sum_{c\in\mathcal{C}} p_{\text{gt}}(c)\log(p_{\text{gt}}(c)) \\ &= \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}, b_{\text{gt}})] + \mathcal{H}(p_{\text{gt}}).\end{aligned}$$
2) For any optimal negative prediction: The strategy sets $\hat{p}_n^*(\emptyset) = 1$. The resulting expected loss is:
$$\mathcal{L}(\hat{y}_n^*, \emptyset) = -w_\emptyset \log(1) = 0.$$

Summing these terms, the strategy achieves the lower bound derived in Proposition 1:

$$\mathbb{E}[\mathcal{L}_{\text{total}}] = \min_{\hat{b}} \mathbb{E}[\mathcal{L}_{\text{box}}(\hat{b}, b_{\text{gt}})] + \mathcal{H}(p_{\text{gt}}).$$

Any deviation from this strategy results in a strictly higher loss:

- If $\hat{p}_n(\emptyset) < 1$, then $-w_\emptyset \log(\hat{p}_n(\emptyset)) > 0$.
- If $\hat{b}_p$ is not the minimizer, the box loss term increases.
- If $\hat{p}_p \neq p_{\text{gt}}$, then either $-\log(S) > 0$ or $\text{CE}(p_{\text{gt}}, \hat{q}_p) > \mathcal{H}(p_{\text{gt}})$ (by Gibbs' inequality).

Thus, the proposed prediction set is the unique global minimizer. □

C. Proof of Proposition 3

Proposition 3 (Optimal Strategy under Uncertainty)**.** *For a query with a matching probability* P_i *to the ground-truth object whose class distribution is* p_{gt}*, the class prediction that minimizes the expected loss is* $\hat{p}_i^*$*, where*

1) *the prediction for any foreground class* $c \in \mathcal{C}$ *is:*
$$\hat{p}_i^*(c) = p_{fg}^*(P_i)p_{gt}(c). \tag{9}$$
2) *the prediction for the background class is*
$$\hat{p}_i^*(\emptyset) = 1 - p_{fg}^*(P_i),$$

where $p^*_{fg}(P_i)$ is the optimal foreground probability, defined as

$$p^*_{fg}(P_i) = \frac{P_i}{P_i + w_\emptyset(1-P_i)}. \tag{10}$$

Proof. We find the optimal probability distribution $\hat{p}_i$ by minimizing the expected classification objective:

$$J(\hat{p}_i) = -P_i \sum_{c \in \mathcal{C}} p_{\text{gt}}(c) \log(\hat{p}_i(c)) - w_\emptyset(1-P_i)\log(\hat{p}_i(\emptyset)).$$

This is a constrained optimization problem, as we must satisfy the constraint that:

$$g(\hat{p}_i) = \sum_{c \in \mathcal{C}} \hat{p}_i(c) + \hat{p}_i(\emptyset) - 1 = 0.$$

We form the Lagrangian $\mathcal{J}$ with a Lagrange multiplier λ:

$$\mathcal{J} = J(\hat{p}_i) - \lambda g(\hat{p}_i).$$

We find the optimum by setting the partial derivatives of $\mathcal{J}$ with respect to each probability $\hat{p}_i(c)$ to zero.

For any foreground class $c \in \mathcal{C}$:

$$\frac{\partial \mathcal{J}}{\partial \hat{p}_i(c)} = 0 \implies \hat{p}_i(c) = -\frac{P_i \cdot p_{\text{gt}}(c)}{\lambda}.$$

For the background class $\emptyset$:

$$\frac{\partial \mathcal{J}}{\partial \hat{p}_i(\emptyset)} = 0 \implies \hat{p}_i(\emptyset) = -\frac{w_\emptyset(1-P_i)}{\lambda}.$$

We solve for λ by substituting these back into the constraint $g(\hat{p}_i) = 0$:

$$\sum_{c \in \mathcal{C}} \left(-\frac{P_i \cdot p_{\text{gt}}(c)}{\lambda} \right) + \left(-\frac{w_\emptyset(1-P_i)}{\lambda} \right) = 1.$$

Since $\sum_{c \in \mathcal{C}} p_{\text{gt}}(c) = 1$ by definition:

$$-\frac{P_i}{\lambda} - \frac{w_\emptyset(1-P_i)}{\lambda} = 1 \implies \lambda = -(P_i + w_\emptyset(1-P_i)).$$

Finally, we substitute this λ back to find the optimal predictions $\hat{p}^*_i$:

$$\hat{p}^*_i(c) = -\frac{P_i \cdot p_{\text{gt}}(c)}{-(P_i + w_\emptyset(1-P_i))} = \left(\frac{P_i}{P_i + w_\emptyset(1-P_i)} \right) p_{\text{gt}}(c),$$

and

$$\hat{p}^*_i(\emptyset) = -\frac{w_\emptyset(1-P_i)}{-(P_i + w_\emptyset(1-P_i))} = 1 - \left(\frac{P_i}{P_i + w_\emptyset(1-P_i)} \right).$$

□

APPENDIX C
OMITTED EXPERIMENTAL RESULTS

This section provides the omitted experimental results. Figure 14 extends Figure 8 across different DETR variants and datasets. Figure 13 extends Figure 5 across different DETR variants and datasets.

APPENDIX D
COMPARATIVE STUDY ON POST-PROCESSING

In this section, we provide a comprehensive analysis regarding the impact of post-processing on the model's overall calibration quality and UQ performance. For the post-processing (i.e., separation) methods, we compare the following approaches: (1) applying a threshold on the confidence score, (2) selecting the Top-k predictions, and (3) utilizing NMS.

Primarily, we evaluate OCE by varying the hyperparameter for each method (i.e., confidence threshold, k, and IoU threshold, respectively) and compare the resulting performance. In this first analysis, to exclude the dependency of the final performance on OCE, we choose the best hyperparameter on the validation set for each setting and compare their best possible performances. Table 3 shows that the confidence thresholding approach outperforms the top-k and NMS approaches. The top-k approach underperforms because the number of objects in an image varies significantly, making it impractical to determine a single optimal value for k. Top-k becomes optimal when $k = 20$. Given that the average and 95th percentile number of objects per image in the COCO dataset are 7 and 22, respectively, these results appear reasonable. Therefore, we confirm that using an excessively large number (e.g., 100) for top-k is inadequate for achieving well-calibrated predictions. On the other hand, with NMS (when applied without preceding confidence thresholding), we empirically observe that although most of the optimal positive predictions are included, a substantial number of negative predictions are retained, often outnumbering the positive ones by several times.

Similarly, we compare the image-level uncertainty quantification performance, and the results are shown in Table 4. In this experiment, we fixed the scaling factor λ at 1.0 to minimize its influence on selecting the optimal hyperparameter. We reconfirm the effectiveness of confidence thresholding; for Deformable-DETR, Cal-DETR, and DINO, the optimal thresholds are approximately 0.3, aligning well with values commonly employed in previous studies. For UP-DETR, the optimal thresholds exceed 0.5, highlighting the potential limitation of using a fixed threshold in image-level UQ applications. Moreover, the correlation is often negative when NMS is applied along with the Conf$^+$ framework. This is because, while most of the optimal positive queries are likely to be included with NMS, a substantial number of optimal negative queries remain within the final subset, often outnumbering the positive ones by several times. As a result, applying NMS leads to an inaccurate reliability assessment. However, many schemes, including NMS, achieve significant improvement when used with the proposed contrastive framework; even if the post-processing scheme is inaccurate, the framework can robustly perform by leveraging the negative predictions.

Lastly, we plot the OCE and image-level UQ performance with different hyperparameter selections to show the sensitivity of each scheme to the choice of hyperparameter. Figure 15 highlight that *carefully identifying a reliable subset is crucial for achieving both high object-level calibration quality and ef-*

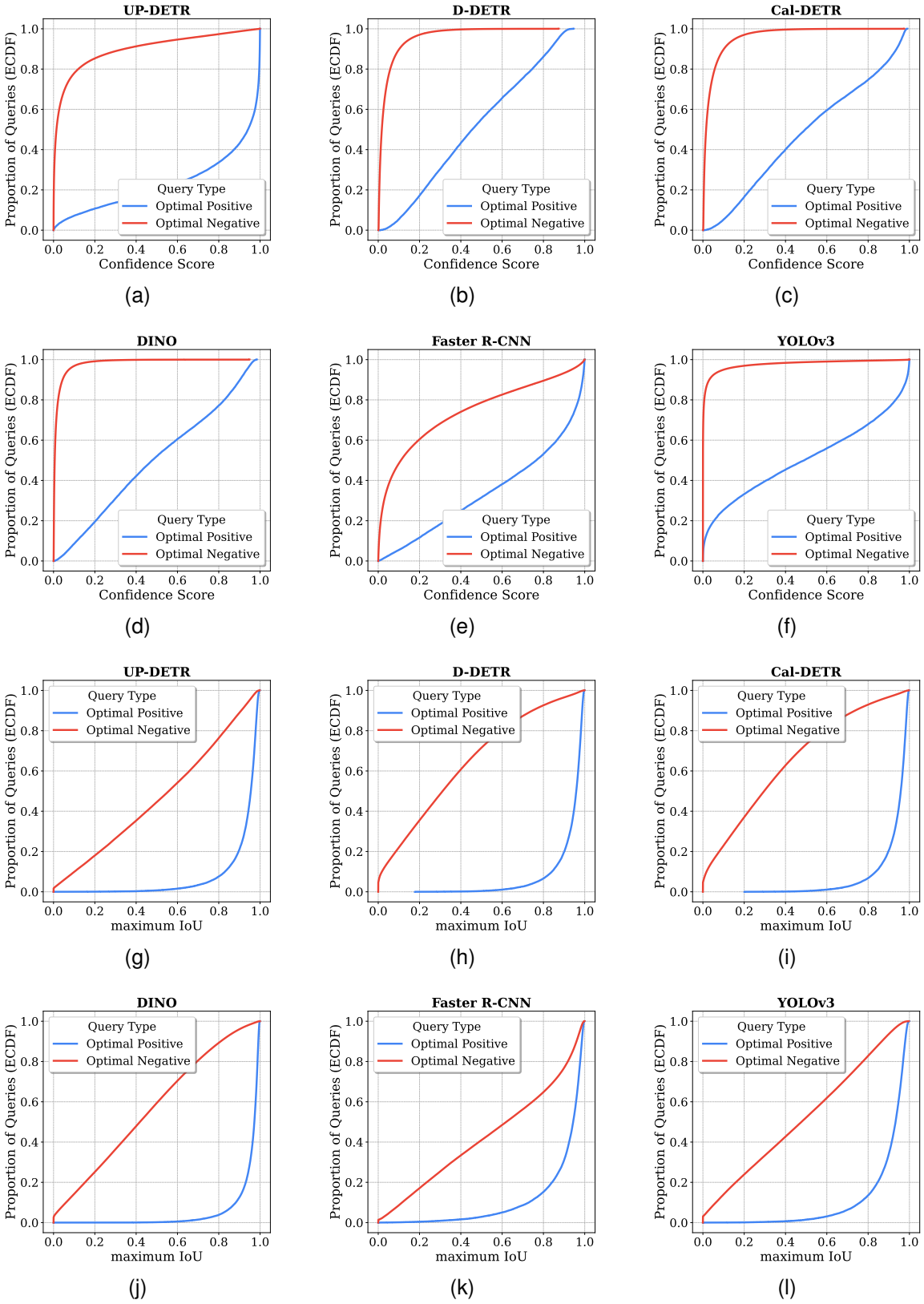


Fig. 13. Empirical cumulative distribution function (ECDF) plots for maximum confidence score and maximum IoU, comparing optimal positive predictions (blue) and optimal negative predictions (red) on the COCO dataset.

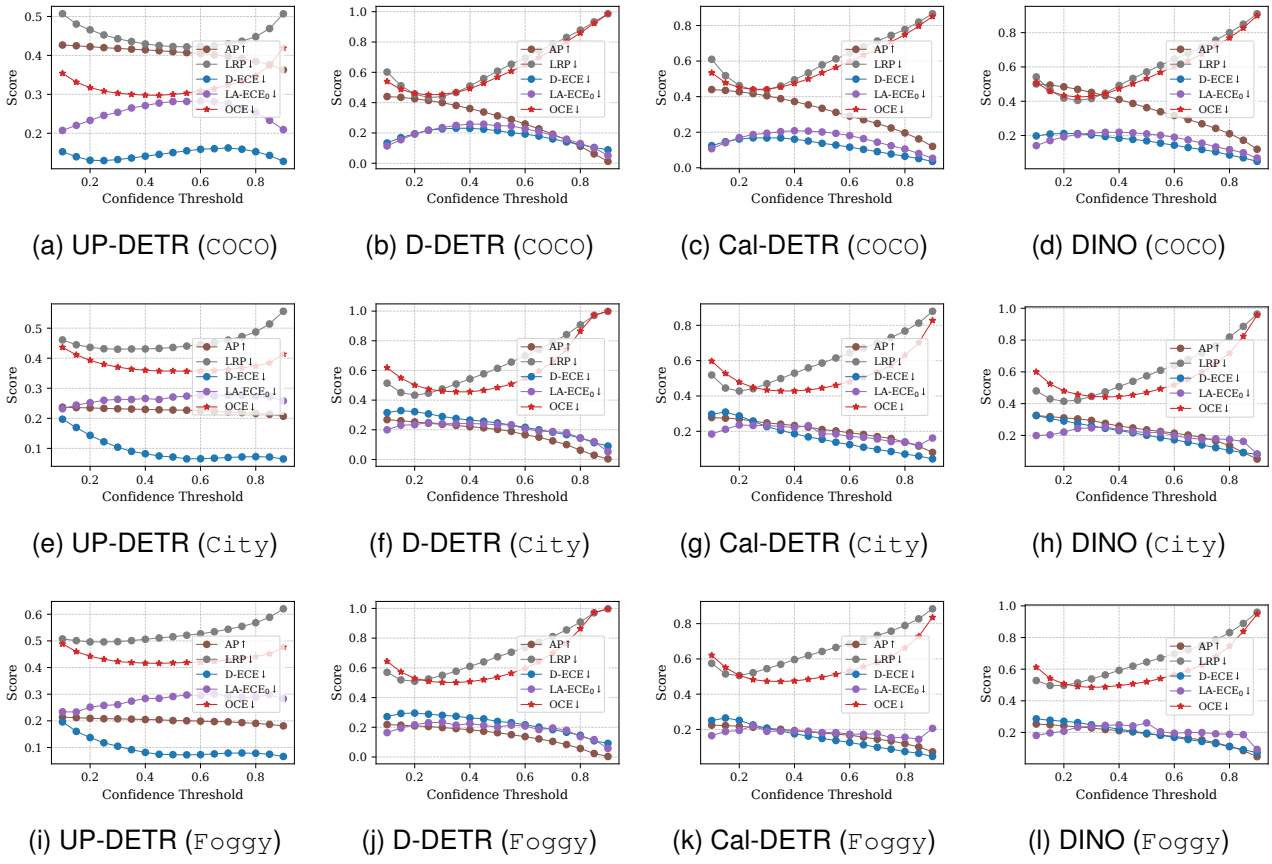
Table 3. Comparison of post-processing schemes based on calibration quality (i.e., OCE). We evaluated three standard methods: confidence thresholding, top-k, and NMS (without confidence thresholding). For each scheme, the optimal hyperparameter selected from the validation set is shown in parentheses, and is applied on the test set. the strongest correlations are highlighted in bold. Confidence thresholding achieves the lowest OCE, demonstrating its better efficacy compared to the other schemes. The optimal thresholds are approximately 0.3, aligning well with values commonly employed in previous studies.

Methods	COCO (in-distribution)				Cityscapes (near OOD)				Foggy Cityscapes (OOD)			
	UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO
Thresholding	**0.276 (0.35)**	**0.450 (0.20)**	**0.416 (0.25)**	**0.436 (0.25)**	**0.313 (0.45)**	**0.430 (0.30)**	**0.413 (0.30)**	**0.402 (0.35)**	**0.387 (0.40)**	**0.488 (0.30)**	**0.469 (0.30)**	**0.458 (0.30)**
Top-k	0.358 (20.00)	0.485 (20.00)	0.457 (20.00)	0.479 (20.00)	0.352 (20.00)	0.451 (20.00)	0.436 (20.00)	0.422 (20.00)	0.416 (20.00)	0.500 (20.00)	0.490 (20.00)	0.473 (20.00)
NMS	0.358 (0.90)	0.535 (0.90)	0.624 (0.90)	0.521 (0.90)	0.426 (0.90)	0.625 (0.90)	0.784 (0.90)	0.583 (0.90)	0.495 (0.90)	0.650 (0.90)	0.794 (0.90)	0.609 (0.90)

Table 4. Comparison of post-processing schemes and baseline methods based on their Pearson correlation coefficients with image-level reliability when Conf^{+} and $\mathsf{ContrastiveConf}(\lambda = 1.0)$ are applied, respectively. We evaluated three standard methods: confidence thresholding, top-k, and NMS (without confidence thresholding). For each scheme, the optimal hyperparameter selected from the validation set is shown in parentheses and is applied on the test set. The strongest correlations are highlighted in bold. Confidence thresholding achieves the lowest OCE, demonstrating its better efficacy compared to the other schemes. For Deformable-DETR, Cal-DETR, and DINO, the optimal thresholds are approximately 0.3, aligning well with values commonly employed in previous studies. For UP-DETR, the optimal thresholds often exceed 0.5, highlighting the potential limitation of using a fixed threshold for image-level UQ.

UQ	Methods	COCO (in-distribution)				Cityscapes (near OOD)				Foggy Cityscapes (OOD)			
		UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO	UP-DETR	D-DETR	Cal-DETR	DINO
Conf^{+}	Thresholding	**0.479 (0.25)**	**0.478 (0.20)**	**0.547 (0.25)**	**0.512 (0.25)**	**0.333 (0.80)**	**0.355 (0.25)**	**0.411 (0.20)**	**0.403 (0.35)**	**0.229 (0.50)**	**0.291 (0.25)**	**0.281 (0.15)**	**0.282 (0.35)**
	Top-k	0.067 (1.00)	0.132 (1.00)	0.252 (1.00)	0.155 (1.00)	-0.037 (2.00)	0.019 (1.00)	0.078 (1.00)	0.021 (1.00)	0.080 (1.00)	-0.006 (1.00)	0.026 (1.00)	0.090 (1.00)
	NMS	-0.524 (0.10)	-0.508 (0.10)	-0.488 (0.10)	-0.510 (0.10)	-0.370 (0.10)	-0.324 (0.10)	-0.273 (0.10)	-0.338 (0.10)	-0.205 (0.10)	-0.251 (0.10)	-0.202 (0.15)	-0.204 (0.10)
$\mathsf{ContrastiveConf}$	Thresholding	0.597 (0.55)	**0.513 (0.25)**	**0.563 (0.25)**	**0.540 (0.25)**	0.408 (0.80)	**0.389 (0.30)**	**0.422 (0.20)**	**0.437 (0.35)**	**0.312 (0.75)**	**0.317 (0.25)**	**0.288 (0.15)**	**0.311 (0.35)**
	Top-k	**0.629 (1.00)**	0.268 (1.00)	0.320 (1.00)	0.290 (1.00)	**0.409 (1.00)**	0.169 (1.00)	0.151 (1.00)	0.220 (1.00)	0.284 (1.00)	0.077 (1.00)	0.060 (1.00)	0.174 (1.00)
	NMS	-0.227 (0.10)	-0.157 (0.90)	-0.402 (0.10)	-0.140 (0.90)	-0.044 (0.80)	-0.034 (0.75)	-0.203 (0.10)	-0.080 (0.80)	0.035 (0.75)	-0.035 (0.75)	-0.149 (0.90)	-0.075 (0.70)

fective image-level uncertainty quantification performance. We also present this with exemplary visualizations in Figure 16.

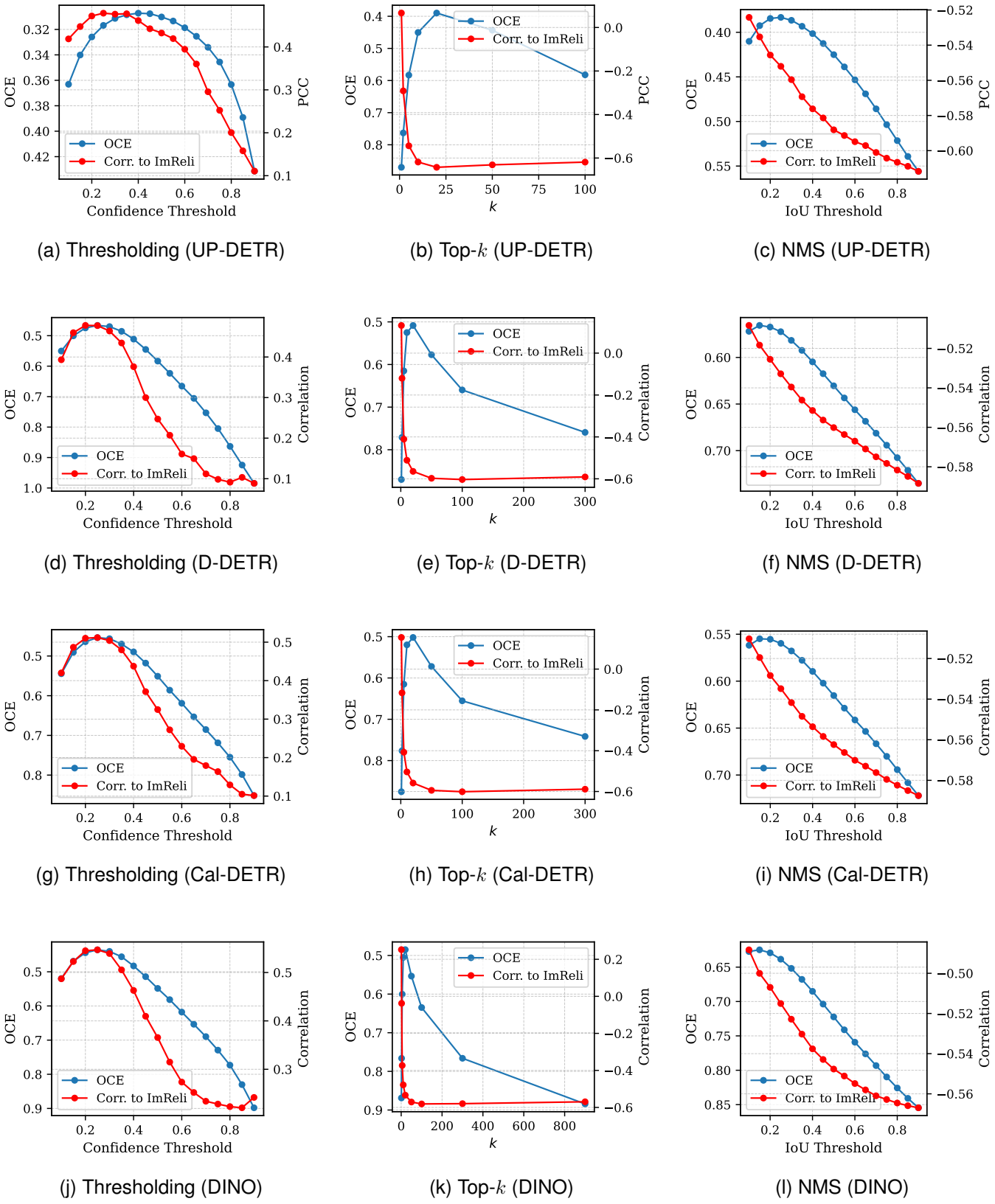


(a) UP-DETR (COCO) (b) D-DETR (COCO) (c) Cal-DETR (COCO) (d) DINO (COCO)

(e) UP-DETR (City) (f) D-DETR (City) (g) Cal-DETR (City) (h) DINO (City)

(i) UP-DETR (Foggy) (j) D-DETR (Foggy) (k) Cal-DETR (Foggy) (l) DINO (Foggy)

Fig. 14. Impact of confidence threshold selection on various performance metrics in UP-DETR, Deformable-DETR, Cal-DETR, and DINO on COCO, Cityscapes, and Foggy Cityscapes. Higher scores are preferable (↑), while lower scores are preferable (↓).



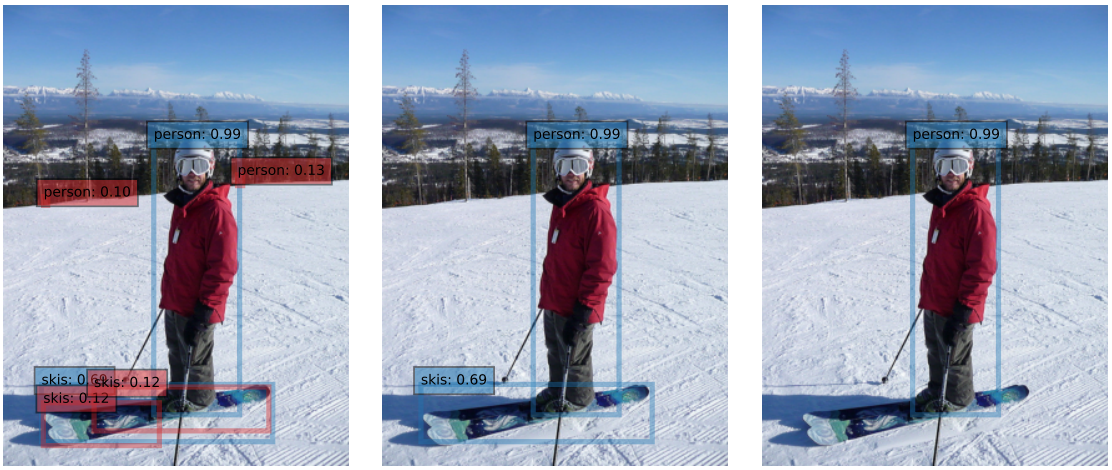
(a) Thresholding (UP-DETR) (b) Top-k (UP-DETR) (c) NMS (UP-DETR)

(d) Thresholding (D-DETR) (e) Top-k (D-DETR) (f) NMS (D-DETR)

(g) Thresholding (Cal-DETR) (h) Top-k (Cal-DETR) (i) NMS (Cal-DETR)

(j) Thresholding (DINO) (k) Top-k (DINO) (l) NMS (DINO)

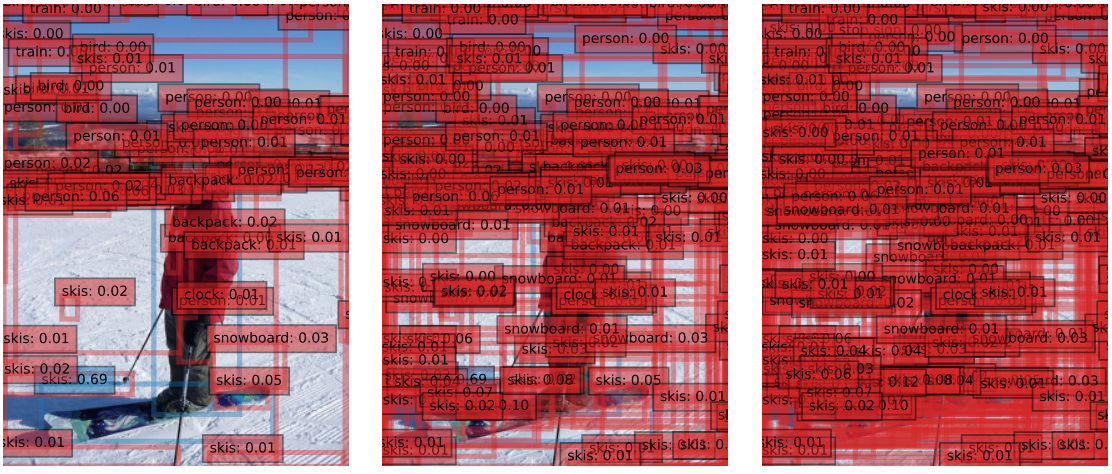
Fig. 15. Impact of parameter selection on OCE (y-axis inverted) and the Pearson correlation coefficient (PCC) between Conf^{+} and image-level reliability across different DETR models on COCO for different post-processing schemes.



(a) Thresholding (0.1) (b) Thresholding (0.27) (c) Thresholding (0.9)



(d) Top-1 (e) Top-20 (f) Top-100



(g) NMS ($\tau = 0.1$) (h) NMS ($\tau = 0.5$) (i) NMS ($\tau = 0.9$)

Fig. 16. Exemplary visualization demonstrating the impact of parameter selection on the final subset of predictions in Cal-DETR for different post-processing schemes. Optimal positive and negative predictions are highlighted with green and red boxes, respectively. As shown, the top-k and NMS approaches could include too many negative predictions, degrading the calibration quality. Confidence thresholding with too low of a threshold faces a similar issue, while too high of a threshold risks omitting positive predictions. Therefore, accurately identifying a reliable set of predictions significantly affects the reliability of DETR for downstream applications.