# An Entropic Metric for Measuring Calibration of Machine Learning Models

Daniel James Sumler[1,*], Lee Devlin[1], Simon Maskell[1] and Richard Oliver Lane[2]

[1]University of Liverpool, United Kingdom
[2]QinetiQ, United Kingdom

**Abstract**

Understanding the confidence with which a machine learning (ML) model classifies an input datum is an important, and often over-looked, concept. We propose a new probability calibration metric, the Entropic Calibration Difference (ECD). Inspired by existing research in the field of state estimation, specifically target tracking, we show how ECD may be applied to binary and multi-class classification ML models. We describe the relative importance of under- and over-confidence and how they are not conflated in the tracking literature. Indeed, our metric naturally distinguishes under- from over-confidence. We consider this important given that algorithms that are under-confident are likely to be "safer" or more trustworthy than algorithms that are over-confident, albeit at the expense of also being over-cautious and so statistically inefficient. We demonstrate how this new metric performs on real data and present a theoretical analysis to prove its properties. We also compare with other metrics for ML model probability calibration, including the Expected Calibration Error (ECE).

**Keywords**

calibration, over-confidence, miscalibration, safety

## 1. Introduction

Calibration of probabilities is an important and often overlooked concept when developing machine learning (ML) models. Usually, accuracy is the main metric used to calculate how well an ML model performs in terms of predicting a class for unseen data. Generally speaking, the closer the accuracy is to 100%, the better the model is deemed to be. However, this does not take into account the probability of predictions that the model outputs, which can be just as important, if not more, than the accuracy.

In binary classification, a probability greater than a threshold, typically 0.5, is enough to decide whether an input belongs to one of two classes. While accuracy informs whether a classification is correct, a probability calibration metric informs how well the confidence probabilities match the true proportions of correct decisions. For example, a model that always outputs a probability of 0.6 for class label 1, but always classifies correctly, should produce a poor calibration score, as even though the model classifies the input correctly, it has low confidence in that decision.

Calibration has become even more important as of late, as the research of Guo et al. [1] reveals that while modern neural networks are more accurate than ever, they are also badly calibrated. This could be attributed to over-confidence of the networks due to the large amount of data they are trained on. This over-confidence can lead to a loss of trust of users in the models.

A well-calibrated model is defined as one that outputs probabilities that are representative of the real-life occurrences from the unseen data. For example, if, on average, 70% of people are correctly predicted to contract a certain disease, then one would expect the average probability outputted by a diagnosis model to be 0.7. A mathematical representation of calibration can be seen in (1), where $X$ is the true label, $k$ represents a class from the total number of classes $K$, $\hat{P}$ is the predicted probability distribution, $\mathbf{p}$ is the vector of class confidences with elements $p_k$, and $\mathbb{P}$ is the true probability.

*TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.*

*Corresponding author.
daniel.sumler@liverpool.ac.uk (D. J. Sumler); ljdevlin@liverpool.ac.uk (L. Devlin); smaskell@liverpool.ac.uk (S. Maskell); rlane1@qinetiq.com (R. O. Lane)

$$\mathbb{P}(X = k|\hat{P} = \mathbf{p}) = p_k \; \forall k \in \{1, ..., K\} \tag{1}$$

In this paper, we present a novel calibration metric that addresses some weaknesses of some of the most commonly-used existing metrics. In section 2, we discuss existing calibration metrics that are widely discussed in the literature. In section 3, we detail our motivations for 'safe' or trustworthy calibration and why we feel our metric is necessary. Section 4 defines our new metric and details how the results can be interpreted. In section 5, we use hypothesis tests for our metric on pre-trained models using calibrated and uncalibrated probabilities and make a comparison with other metrics. Results for binary classifiers are included in the main paper, while multiclass results are found in Appendix A. Finally, section 6 concludes the paper.

## 2. Existing metrics for model calibration

In this section, we explore the literature and highlight some important calibration metrics, from early pioneers of the field to widely-used modern standards.

Various methods exist that attempt to calibrate the probabilities outputted by badly calibrated models. Some attempt to do this by altering their training process, such as using a loss function that addresses neural network calibration as it is being trained [2], while other methods attempt to change a model's output probabilities [1, 3, 4, 5]. The former method is useful if we want to train a new model; however, this is not always required. Occasionally, extant trained models need their existing probabilities calibrated, which is where the latter method is used. However, before calibrating a model's output probabilities, it must first be determined whether the model is already calibrated or not, and to what extent, using a calibration metric. The remainder of this section defines and discusses various metric benefits and drawbacks.

### 2.1. Reliability Diagram

Methods have been developed to allow one visually to assess the calibration of a model. First brought to the limelight by DeGroot and Fienberg [6], and then further expanded upon by Niculescu-Mizil and Caruana [7], reliability diagrams are a widely-used method to display calibration. This visual technique works by partitioning the probabilities that a model outputs into a number of bins, where each bin represents a probability interval. It is possible to use any number of bins, but one must keep in mind the bias-variance trade-off as highlighted by Nixon et al. [8]. They specify that when the number of bins is increased, this results in a lower population per bin, and, in turn, a larger variance. This will, however, also result in a lower bias. Therefore, the number of bins should be fine-tuned for each problem.

Once every probability has been assigned to the appropriate bin, the average predicted probability and fraction of class 1 labels is calculated for each bin. For multiclass reliability diagrams, a common method is to take the top-label, also known as most confident, prediction [1]. The fraction of positives is plotted against the average predicted probability, with a reference line $fraction = probability$ showing what a perfectly calibrated model would look like. Points above the line are under-confident, while points below it are over-confident. A typical reliability diagram is shown in Figure 1.

Reliability diagrams can be a good tool for visualising a model's calibration. However, they do not, in themselves, return a calibration score, and the diagram can be misleading when some bins are sparsely populated.

### 2.2. Expected Calibration Error

Initially proposed by Naeini et al. [9], the Expected Calibration Error (ECE) is one of the most widely used calibration metrics. It is a simple, yet effective, formula that uses the same method of binning as reliability diagrams, but produces a normalised calibration score between 0 and 1, similar to methods such as the Brier Score [10].

ECE works by calculating the accuracy and confidence of each bin in a reliability diagram. The accuracy is the traditional definition, where the number of correct predictions is divided by the total number of predictions in the bin. The confidence refers to the average probability in a bin. It is not explicitly stated in the original ECE paper, but Guo et al. [1] use the maximum class probability to calculate the confidence of a model output in multiclass classification scenarios. This means that, for a multiclass classifier, the minimum probability in all bins will be $1/K$, and some of the bins may be unpopulated. In binary classification, the estimated probability, $\hat{p}_i$, of the positive class is used when calculating the mean confidence for bin $B_m$, as seen in (2), and the ratio of class 1 labels is taken instead of the accuracy [11, 12], as shown in (3), where $y_i$ is the true label. These are the same values used in the reliability diagram. The ECE of each bin is then calculated by obtaining the absolute difference between (3) and (2). This ECE value is weighted depending on how populated each bin is, as shown in (4), where $N$ is the total number of probabilities across all bins, and $M$ is the number of bins.

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{|B_m|} \hat{p}_i \tag{2}$$

$$frac_{pos}(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{|B_m|} 1(y_i = 1) \tag{3}$$

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |frac_{pos}(B_m) - conf(B_m)| \tag{4}$$

While ECE is widely-used in the literature, it has some shortcomings. One of the most noticeable is the use of bins, and the fact that the user needs to decide on an optimal binning strategy, or rely on an adaptive algorithm [13]. Due to its positive fraction and confidence terms, as shown in (3) and (2), changing the number of bins can change the final ECE score. This also evokes the bias-variance trade-off mentioned previously [8].

Additionally, the ECE equation deals with averages within bins rather than individual sample probabilities and their respective true labels. Due to this, outliers (such as a wildly incorrect prediction) may not have a large impact on the final calibration score. While it may be the case that this is a good thing, since a model should not be heavily penalised for a single mistake, it could be considered much more important in models for sensitive applications, such as predicting the probability of a medical patient having a certain disease.

### 2.3. Maximum Calibration Error

The Maximum Calibration Error (MCE) [9], seen in (5), is a metric similar to ECE; however, it finds the largest difference between the average confidence and average accuracy across all bins.

$$MCE = \max_{m \in \{1,..,M\}} |acc(B_m) - conf(B_m)| \tag{5}$$

Theoretically, it is a miscalibration metric that checks the worst-case scenario, which is especially useful in applications where the largest error should be minimised. This makes it useful in safety-critical applications. The closer the metric is to 0, the less deviation there is between accuracy and confidence. A value closer to 1 shows that there is a large discrepancy present.

This metric suffers from the same drawback as ECE due to its utilisation of bins. To ensure that a valid MCE value is received, it is good practice to use an adaptive binning practice, such as the one used by Lee et al. [13].

### 2.4. Other Works

There exist numerous other calibration metrics outside of the ones we mentioned in this section. A comprehensive review of classifier probability calibration metrics is given in [14]. Honourable mentions

include the Maximum Mean Calibration Error (MMCE), which was first proposed by Kumar, Sarawagi and Jain [15]. It is used to measure calibration error, and can also be used to train network parameters, which fixes poor calibration, while maintaining high accuracy.

Luo et al. [16] proposed the Local Calibration Error (LCE), a kernel-based metric which is used to bridge the gap between metrics that look at the average reliability across a population of probabilities (such as ECE and MCE) and reliability calculation for individual data points.

Verhaeghe et al. [17] propose the Expected Signed Calibration Error (ESCE), a signed counterpart of the ECE metric, with its main difference being the removal of the absolute value in the equation. Ao et al. [18] proposed the Miscalibration Score (MCS) which, like ESCE, attempts to present over- and under-confidence by altering the ECE equation. They also propose a modified version of temperature scaling [1], which is a post-hoc re-calibration technique.

The above metrics attempt to find how well-calibrated a model is according to the definition in (1). In the following section, we detail our proposal – the concept of 'safe' calibration. This not only looks for perfect calibration, but also strongly penalises over-confidence, which is considered unsafe in some models.

## 3. Motivation

In this section, we present background knowledge that frames the metric we propose in this paper, and what we call 'safe' calibration, a property that makes models trustworthy. In the following subsections, we talk about the target tracking (TT) literature which is the inspiration for safe calibration. We also propose our metric, the Entropic Calibration Difference, show how this fits into the TT field, and demonstrate how it can be adapted for general use in ML model calibration.

### 3.1. Target Tracking

A fundamental goal of target tracking is to derive the state (e.g. position and velocity) of an object over time through noisy measurements. This is accomplished by using algorithms called target trackers.

Tracking systems generally consist of multiple components; however, one of the core algorithms is track filtering. Commonly used methods include Kalman Filters and Particle Filters [19]. Target trackers are often used to process 2D polar measurements. When the range to a target is relatively well estimated, contours of the 2D probability distributions involved each resemble a banana. It transpires that the Taylor series used by the Extended Kalman Filter (EKF) [20] fails to represent the uncertainty caused by the curvature of the distribution adequately. This can lead an EKF to diverge over time since it attributes excessive confidence to the output of processing previous data relative to a newly acquired datum. Techniques such as the Unscented Kalman Filter (UKF) [21, 22] attempt to address this using a form of quasi monte-carlo integration. In the TT literature there is therefore a need to quantify the extent to which a technique consistently under-estimates the uncertainty.

One of the most popular consistency metrics is the Normalised Estimation Error Squared (NEES), shown in (6), where $\hat{y}_i$ represents the estimated value and $\sigma_i^2$ is the variance associated with the $i$-th estimation error. Note that this equation assumes that the true value, $y$, is 1D. This metric works by calculating the ratio between the actual estimation error (being the difference between the predicted and true states), and the predicted error.

$$NEES = \frac{1}{N}\sum_{i=1}^{N}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2} \tag{6}$$

NEES is a good consistency metric for single-target tracking. In this paper, we propose the Entropic Calibration Difference metric, which is inspired by NEES. One of the main aspects that the metric borrows from NEES is the higher penalisation of over-confidence compared to under-confidence [23].

NEES is tested against a chi-squared distribution. It outputs a scalar value with regard to the consistency of a state estimator. If the uncertainty values are not reasonably accurate, the computed NEES value will highlight this.

For a system with well-calibrated uncertainty in its output, a NEES score of 1, or $d$ if $y$ is more than 1D, is expected on average and shows perfect consistency, as it reflects a balance between the squared prediction error and the predicted uncertainty. Values greater than 1, or $d$, show over-confidence, as the prediction error is large relative to the predicted uncertainty. This suggests that the model underestimated its uncertainty, which led it to be too confident in its predictions. A value smaller than 1 shows under-confidence as the prediction error is small relative to the predicted uncertainty. This suggests that the model overestimated its uncertainty, making it overly cautious about its predictions. The ability to find over- and under-confident predictions using only a single equation is desirable for calibration, as it gives good context in understanding the model's predictions. Treating over- and under-confidence differently would also give us an idea of whether a model is safely calibrated.

### 3.2. Safe Calibration

The aim of safe calibration is to determine whether an ML model can be deemed safe to use and is thus trustworthy. A non-calibrated model runs the risk of being over-confident in incorrect answers, or under-confident in the correct answers. We bring the TT way of thinking, where over-confidence is considered much worse than uncertainty or under-confidence, into ML. The theory behind this is that we would prefer a model be uncertain in the correct class rather than confidently choose an incorrect class. To this end, we penalise overconfidence more than under-confidence and uncertainty.

Entropic Calibration Difference can determine whether a model is well-calibrated, which can be interpreted as whether the model is safe to use. For example, if a binary classification model was determining whether it is safe for an aircraft to land, over-confidence in the incorrect class could potentially be fatal. Therefore, we would prefer to have under-confidence in the correct class or uncertainty, rather than random confident guessing [24]. ISO26262 also defines 'safety' as a lack of 'unacceptable risk' when referring to modern road vehicles [25].

## 4. Entropic Calibration Difference

### 4.1. Background and Definition

The Entropic Calibration Difference (ECD) is applicable to both TT and ML calibration, or any other probabilistic model, and does not require any parameters other than the true label and prediction. Equation (7) shows the general definition of the ECD metric.

$$ECD = \frac{1}{N}\sum_{i=1}^{N}\left[\int \log p(x|y_i)p(x|y_i)dx - \log p(x_i|y_i)\right] \tag{7}$$

where $N$ is the total number of data points, $y_i$ are the measured data for data point $i$, $p(x|y_i)$ is the algorithm's estimate of the probability density or probability mass of true state $x$ given the measurement, and $x_i$ are the known true states for a test set.

The first term in the summand in (7), containing the integral, is the negative entropy of the predicted probability distribution, or expected log likelihood, for a particular data point. This is used to represent under-confidence. The second term of the summand is the log likelihood, which is used to represent over-confidence. It should be noted that if the entropy term were zero, the overall expression would be negative log-likelihood (NLL), which is a commonly used metric to measure the calibration of classifiers. In general, ECD measures the difference between expected and actual log likelihoods. In this case, we have a metric that can produce negative scores for under-confident values and positive scores for over-confident values. However, unlike other calibration metrics such as ECE, under- and over-confidence are not treated the same, as the metric follows safe calibration scoring. Some metrics

require the use of optimal binning methods or other computationally expensive calculations. However, ECE, MCE, and ECD all run in linear time complexity, in terms of the number of data points.

## 4.2. Proving Relation with NEES

NEES can be proven to be a special case of ECD by substituting the Gaussian distribution into both equations. Equation (8) is the Gaussian formula, (9) is the Gaussian ECD, and (10) is the Gaussian NEES.

$$p(x|y_i) = \frac{1}{\sqrt{(2\pi|C_i|)}} e^{-\frac{1}{2}(x-\mu_i)^T C_i^{-1}(x-\mu_i)} \tag{8}$$

$$ECD = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}(x-\mu_i)^T C_i^{-1}(x-\mu_i) - \frac{d}{2}\right] \tag{9}$$

$$NEES = \frac{1}{N}\sum_{i=1}^{N}\left[(x-\mu_i)^T C_i^{-1}(x-\mu_i)\right] \tag{10}$$

In the above, $x$ is the true state vector, $y_i$ is the observation, and $\mu_i$ and $C_i$ are the mean vector and covariance matrix of the Gaussian uncertainty given the observation. NEES should be equal to the dimensionality $d$ in (9) for a system to be consistent and the ECD score would be zero.

## 4.3. ECD for Discrete Variables

To apply the ECD formula to classification problems, it is necessary to allow it to be used with discrete variables. This is a simple change, as noted in (11).

$$ECD = \frac{1}{N}\sum_{i=1}^{N}\left[\left[\sum_{k=1}^{K} p(x=k|y_i)\log p(x=k|y_i)\right] - \log p(x_i|y_i)\right] \tag{11}$$

In this modified equation, we predict the probability that $x$ is of class $k$ given a piece of observed data $y_i$, where $N$ and $K$ are the number of data points and number of classes, respectively.

### 4.3.1. ECD for Binary Classification

Equation (11) allows for an easy transition to an ECD formula for binary calibration. By substituting (12) and (13) into (11), we create the binary classification ECD formula in (14).

$$\hat{p_i} = p(x=1|y_i) \tag{12}$$

$$\log p(x_i|y_i) = x_i \log \hat{p_i} + (1-x_i)\log(1-\hat{p_i}) \tag{13}$$

$$ECD = \frac{1}{N}\sum_{i=1}^{N}(\hat{p_i} - x_i)\log\left[\frac{\hat{p_i}}{1-\hat{p_i}}\right] \tag{14}$$

With the formula in (14), we are able to determine whether a binary classification model is safe to use, based on its calibration score, as described in the following section.

### 4.3.2. Simplified ECD for Multi-class Classification

When attempting to transition the binary logic of ECD to a multiclass scenario, to preserve the original logic behind the equation, we suggest using a 'True Class vs. Rest' approach, which converts the problem into a binary one.

This, as outlined in (15), uses the probability of the true class, $p_x$, with the true class label permanently set to 1 in this binary representation.

$$ECD = \frac{1}{N}\sum_{x=1}^{N}(p_x - 1)\log\left(\frac{p_x}{1-p_x}\right) \tag{15}$$

The logic remains the same as the binary case, where the ECD calculates whether the model is under-confident in the correct class or over-confident in all other classes.

#### 4.3.3. ECD Score Interpretation

Due to the fact that the ECD score is theoretically unbounded, it is not as easily interpretable as ECE which is bound between $[0, 1]$. Therefore, we advocate for the use of a null hypothesis testing framework, which is highlighted in Section 5.

### 4.4. Theoretical Analysis

To test whether our metric works correctly, we present a theoretical analysis of its properties. The definition of calibration is in (1), while the definition of ECD is in (7). The expected population metric for ECD can be re-written as in (16). Here $\hat{P}_X$ refers to the probability assigned by the true class by the model, while $\hat{P}_k$ is the probability assigned to class *k*.

$$ECD_{pop} = \mathbb{E}\left[\left(\sum_{k=1}^{K}\hat{P}_k \log \hat{P}_k\right) - \log \hat{P}_X\right] \tag{16}$$

By using the law of total expectation, conditioning on the predicted probability $\hat{P} = p$, and the linearity of expectation, we obtain (17).

$$ECD_{pop} = \mathbb{E}_{\hat{P}}\left[\left(\sum_{k=1}^{K} p_k \log p_k\right) - \mathbb{E}[\log p_X | \hat{P} = p]\right] \tag{17}$$

When evaluating the second term of the equation, since $X$ is the true class, we obtain (18).

$$\mathbb{E}[\log p_X | \hat{P} = p] = \sum_{j=1}^{K} P(X = j | \hat{P} = p) \log p_j \tag{18}$$

Under the perfect calibration assumption in (1), this results in (19).

$$\mathbb{E}[\log p_X | \hat{P} = p] = \sum_{j=1}^{K} p_j \log p_j \tag{19}$$

When (19) is substituted back into (17), this results in the cancellation of values. Therefore, under the assumption of perfect calibration, the ECD metric will return a value of zero. Similarly, if the probability of the most likely class is $0.5$, the ECD will compute to zero, indicating safely calibrated probabilities that exhibit neither under- nor over-confidence.

## 5. Evaluation

To measure and compare ECD values with other metrics, we utilise a hypothesis testing model. Widmann, Lindsten, and Zachariah [26] argue that calibration scores, on their own, lack meaning and are difficult to interpret. Vaicenavicius et al. [27] support this claim by stating that miscalibration results may lack much meaning when compared directly with other results. Lee et al. [13] created *T-Cal*, a framework that uses an adaptive binning technique to determine whether ECE accepts or rejects the null hypothesis of perfect calibration. In this scenario, accepting the null hypothesis refers to perfect calibration, and

rejecting it means that there was a statistically significant amount of miscalibration in the probabilities. For the following experiments, this adaptive binning strategy is also adapted for use with MCE.

Consistency resampling, a type of bootstrapping, is used for all metrics. This method helps combat the unfairness of judging a model's calibration score based on a single set of predictions. Rather, the model's confidence scores are re-sampled with replacement to create multiple new datasets, so that calibration metrics provide a range of scores over the multiple datasets. A threshold is set based on the distribution of metric values. If the original score is statistically larger than the re-sampled scores, the null hypothesis is rejected [27].

### 5.1. Hypothesis Test Setup

The Null Hypothesis, $H_0$, is the case of perfect calibration for ECE and MCE. However, since ECD can reach a value of zero due to uncertainty, $H_0$ for ECD is the case of safe calibration.

The alternative Hypothesis, $H_a$, is the case of statistically significant miscalibration for ECE and MCE. For ECD, $H_\alpha$ refers to statistically significant unsafe calibration. As Lee et al. [13] point out, it is not fair to expect a model trained on finite data to be perfectly calibrated. Therefore, accepting the Null Hypothesis does not mean that a model is perfectly calibrated or perfectly safe, but rather that no statistically significant miscalibration or unsafety is detected.

### 5.2. Binary Tests

Binary tests were conducted on two datasets: the Adult Income dataset [28] and the Telco Customer Churn dataset [29]. These were chosen due to their class imbalances; therefore, models trained on these datasets may be less calibrated. Separate simple neural net models consisting of two fully connected hidden layers containing 64 and 32 nodes respectively were trained on each of these datasets. Each model was trained for 100 epochs using the Adam optimiser. A model's probabilities are fed into a metric in three different ways – uncalibrated, with Platt Scaling [4], and with Isotonic Regression [5]. This gives an idea of how the metrics change after post-hoc calibration. Table 1 shows these results. The 'Dataset' row shows which dataset is being used. This is split into three columns under the 'Calibration' row which details which post-hoc calibration technique was used on the model's probabilities before they were analysed by each calibration metric. Each other row shows the calibration metric's final score, and whether the null hypothesis is accepted or rejected for each test. The 'Threshold' row shows the threshold generated during consistency resampling. When a metric's score is larger than this threshold, the $H_0$ is rejected, due to evidence of miscalibration. Figure 1 shows the reliability diagrams for both binary models. This reliability diagram uses the standard binary method of plotting the reliability diagram. The *T-Cal* [13] adaptive binning strategy is used for both ECE and MCE scores.

Table 1 shows the metrics scores and hypothesis test results for each dataset and calibration technique. Interestingly, the ECD score is considered safe in the Adult Income's Platt Scaling results, whereas the ECE does not consider it to be within range of perfect calibration. Looking at Figure 1a, it shows that there is some under-confidence present, and a well-calibrated uncertainty. The MCE metric hypothesis accepts a higher score than the other metrics; this is because it is below the expected value from the consistency resampling technique.

The Telco Customer Churn values show the most interesting results. The ECE metric accepts the null hypothesis for all calibration techniques; however, the ECD rejects them.

This is evidence that, while a model may be well-calibrated according to ECE, it does not necessarily mean that the model does not suffer from over-confident predictions.

Figure 1b shows that although most bins are well calibrated, there is some over-confidence in the higher bins, reinforcing the reason for the ECD rejection.

Results for multiclass classification experiments are included in the Appendix A, with tables 2 and 3.

| Dataset | Adult Income [28] | | | Telco Customer Churn | | |
|---|---|---|---|---|---|---|
| **Calibration** | **None** | **Platt Scaling** | **Isotonic Reg.** | **None** | **Platt Scaling** | **Isotonic Reg.** |
| **ECE Score** | 0.0695 | 0.0247 | 0.0081 | 0.0352 | 0.0304 | 0.04168 |
| **Threshold** | 0.0088 | 0.0114 | 0.0159 | 0.0665 | 0.0674 | 0.05 |
| $H_0$ | Reject $H_0$ | Reject $H_0$ | Accept $H_0$ | Accept $H_0$ | Accept $H_0$ | Accept $H_0$ |
| **MCE Score** | 0.9410 | 0.9644 | 0.9486 | 0.9449 | 0.9406 | 0.9364 |
| **Threshold** | 0.9996 | 0.8504 | 0.9908 | 0.9975 | 0.9974 | 1.0 |
| $H_0$ | Accept $H_0$ | Reject $H_0$ | Accept $H_0$ | Accept $H_0$ | Accept $H_0$ | Accept $H_0$ |
| **ECD Score** | 0.2307 | -0.0057 | -0.0028 | 0.0419 | 0.0253 | 0.04759 |
| **Threshold** | 0.0067 | 0.0081 | 0.0083 | 0.0236 | 0.0238 | 0.0233 |
| $H_0$ | Reject $H_0$ | Accept $H_0$ | Accept $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ |

**Table 1**
Binary Classification Hypothesis Test Results

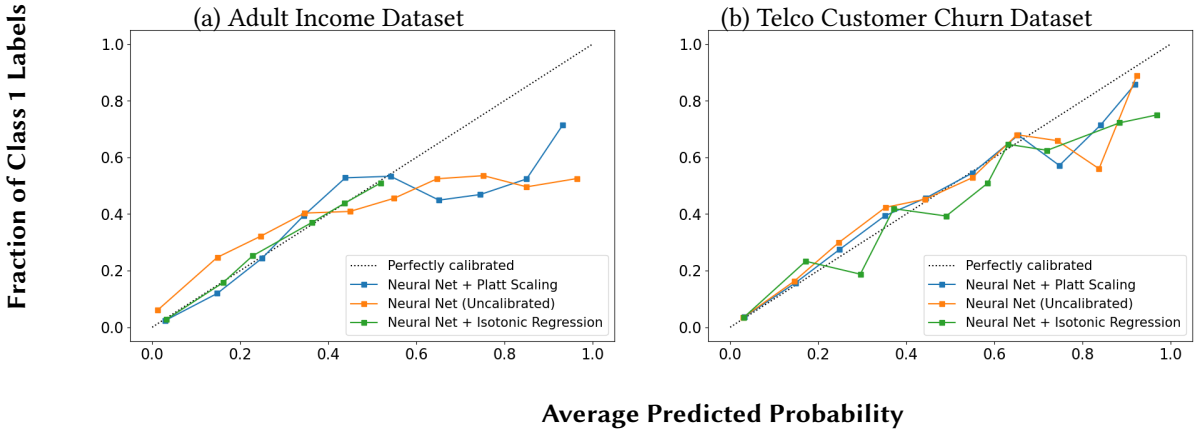


**Figure 1:** Binary Dataset Reliability Diagrams using the probability of class 1. The $x$-axis is the average predicted probability. The $y$-axis is the fraction of class 1 labels.

## 6. Conclusion & Future Work

We have introduced a novel calibration metric, named the Entropic Calibration Difference (ECD) due to its consideration of the entropy of probabilities. The metric is influenced by the Normalised Estimation Error Squared (NEES) metric, which is used to determine the consistency of a state estimator within the target-tracking field of research. The ECD metric is equivalent to applying a generalised version of NEES to the ML problem domain. This new metric also brings a new perspective to the probability calibration literature, namely the concept of safe calibration, which is commonly found in the target tracking literature. We define safe calibration as a metric that prefers under-confidence to over-confidence, due to the belief that an under-confident score in the correct class is safer than an overconfident score in the incorrect class. Therefore, over-confidence is penalised more than under- confidence, rather than equal penalties, which are present in other metrics. In terms of future directions, ECD could be implemented as an objective function as well as integration into re-calibration techniques, with the hopes of making probabilities safer as well as more well-calibrated.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References


[1] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: International conference on machine learning, PMLR, 2017, pp. 1321–1330.

[2] D. Neo, S. Winkler, T. Chen, Maxent loss: Constrained maximum entropy for calibration under out-of-distribution shift, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21463–21472.

[3] B. Böken, On the appropriateness of Platt scaling in classifier calibration, Information Systems 95 (2020) 101641.

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

[5] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002, pp. 694–699.

[6] M. H. DeGroot, S. E. Fienberg, The comparison and evaluation of forecasters, Journal of the Royal Statistical Society: Series D (The Statistician) 32 (1983) 12–22.

[7] A. Niculescu-Mizil, R. Caruana, Obtaining calibrated probabilities from boosting., in: UAI, volume 5, 2005, pp. 413–20.

[8] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, D. Tran, Measuring calibration in deep learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[9] M. P. Naeini, G. Cooper, M. Hauskrecht, Obtaining well calibrated probabilities using Bayesian binning, in: Proceedings of the AAAI conference on artificial intelligence, volume 29, 2015.

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

[11] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, P. Flach, Classifier calibration: a survey on how to assess and improve predicted class probabilities, Mach. Learn. 112 (2023) 3211–3260.

[12] T. Guilbert, O. Caelen, A. Chirita, M. Saerens, Calibration methods in imbalanced binary classification, Ann. Math. Artif. Intell. 92 (2024) 1319–1352.

[13] D. Lee, X. Huang, H. Hassani, E. Dobriban, T-cal: An optimal test for the calibration of predictive models, Journal of Machine Learning Research 24 (2023) 1–72.

[14] R. O. Lane, A comprehensive review of classifier probability calibration metrics, arXiv preprint arXiv:2504.18278 (2025).

[15] A. Kumar, S. Sarawagi, U. Jain, Trainable calibration measures for neural networks from kernel mean embeddings, in: International Conference on Machine Learning, PMLR, 2018, pp. 2805–2814.

[16] R. Luo, A. Bhatnagar, Y. Bai, S. Zhao, H. Wang, C. Xiong, S. Savarese, S. Ermon, E. Schmerling, M. Pavone, Local calibration: metrics and recalibration, in: Uncertainty in Artificial Intelligence, PMLR, 2022, pp. 1286–1295.

[17] J. Verhaeghe, T. De Corte, C. M. Sauer, T. Hendriks, O. W. Thijssens, F. Ongenae, P. Elbers, J. De Waele, S. Van Hoecke, Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in icu patients, International Journal of Medical Informatics 175 (2023) 105086.

[18] S. Ao, S. Rueger, A. Siddharthan, Two sides of miscalibration: identifying over and under-confidence prediction for network calibration, in: Uncertainty in artificial intelligence, PMLR, 2023, pp. 77–87.

[19] X. Li, K. Wang, W. Wang, Y. Li, A multiple object tracking method using kalman filter, in: The 2010 IEEE International Conference on Information and Automation, 2010, pp. 1862–1866. doi:10.1109/ICINFA.2010.5512258.

[20] G. A. Einicke, L. B. White, Robust extended Kalman filtering, IEEE transactions on signal processing 47 (1999) 2596–2599.

[21] S. J. Julier, J. K. Uhlmann, New extension of the Kalman filter to nonlinear systems, in: I. Kadar (Ed.), Signal Processing, Sensor Fusion, and Target Recognition VI, volume 3068, International

Society for Optics and Photonics, SPIE, 1997, pp. 182 – 193. URL: https://doi.org/10.1117/12.280797. doi:10.1117/12.280797.
[22] E. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), 2000, pp. 153–158. doi:10.1109/ASSPCC.2000.882463.
[23] X. R. Li, Z. Zhao, V. P. Jilkov, Estimator's credibility and its measures, in: Proc. IFAC 15th World Congress, 2002.
[24] M. Pitale, A. Abbaspour, D. Upadhyay, Inherent diverse redundant safety mechanisms for ai-based software elements in automotive applications, arXiv preprint arXiv:2402.08208 (2024).
[25] ROHM, Iso 26262: Functional safety standard for modern road vehicles (2020).
[26] D. Widmann, F. Lindsten, D. Zachariah, Calibration tests in multi-class classification: A unifying framework, Advances in neural information processing systems 32 (2019).
[27] J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, T. Schön, Evaluating model calibration in classification, in: The 22nd international conference on artificial intelligence and statistics, PMLR, 2019, pp. 3459–3467.
[28] B. Becker, R. Kohavi, Adult, UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.
[29] IBM, Telco Customer Churn, https://www.kaggle.com/datasets/blastchar/telco-customer-churn, 2018.
[30] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.
[34] B. Recht, R. Roelofs, L. Schmidt, V. Shankar, Do imagenet classifiers generalize to imagenet?, in: International conference on machine learning, PMLR, 2019, pp. 5389–5400.

## A. Multiclass Tests

Multiclass classification hypothesis tests were carried out on the CIFAR-10, CIFAR-100 [30], and ImageNet [31] datasets. Pre-trained models were used with ResNet32, ResNet50 [32] and VGG-19 [33] models.

Unlike the previous binary tests, one calibration method was used on the data. This is due to the consistency resampling technique in the ECD tests requiring the full probability vector, whereas the ECE requires the maximum probability only. To make the tests fair, a calibration method was used that takes the whole vector into account, Temperature Scaling [1]. To calibrate the CIFAR-100 and CIFAR-10 probabilities, 20% of the test set was partitioned into a validation set, due to the unavailability of a dedicated set. For ImageNet, ImageNetV2 [34] was used as a validation set for Temperature Scaling. The T-Cal [13] framework was used for the adaptive binning strategy for both ECE and MCE. Consistency Resampling is used for all metrics, generating a threshold within which the null hypothesis would be accepted.

The results tables are split into two: Table 2 results use uncalibrated model probabilities, and Table 3 results are based on post-hoc temperature scaling. In both tables, the 'Dataset' row details which dataset the tests are conducted with. The 'Calibration' row either specifies 'None' or 'Temperature Scaling'. The 'Model' row specifies which model's probabilities are being used with the results in its respective column. Finally, each other row specifies the calibration metric and whether the null hypothesis is accepted or rejected based on the calibration score and consistency resampling simulations. Figures 2, 3,

| Dataset | CIFAR-10 | | CIFAR-100 | | ImageNet | |
|---|---|---|---|---|---|---|
| Calibration | None | | None | | None | |
| Model | ResNet32 | VGG-19 | ResNet32 | VGG-19 | ResNet50 | VGG-19 |
| ECE Score | 0.0490 | 0.0520 | 0.1444 | 0.2152 | 0.4067 | 0.0274 |
| Threshold | 0.0056 | 0.0032 | 0.0112 | 0.0073 | 0.0066 | 0.0049 |
| $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ |
| MCE Score | 0.3476 | 0.4136 | 0.2725 | 0.4951 | 0.4932 | 0.0527 |
| Threshold | 0.8631 | 0.6555 | 0.0342 | 0.0846 | 0.0126 | 0.0125 |
| $H_0$ | Accept $H_0$ | Accept $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ |
| ECD Score | 0.2830 | 0.3706 | 1.4324 | 1.9696 | 0.8754 | 0.8553 |
| Threshold | 0.0452 | 0.0135 | 0.5181 | 0.1123 | 4.1264 | 0.6844 |
| $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Accept $H_0$ | Reject $H_0$ |

**Table 2**
Multiclass Classification Hypothesis Test Results without post-hoc calibration.

| Dataset | CIFAR-10 | | CIFAR-100 | | ImageNet | |
|---|---|---|---|---|---|---|
| Calibration | Temp. Scaling | | Temp. Scaling | | Temp. Scaling | |
| Model | ResNet32 | VGG-19 | ResNet32 | VGG-19 | ResNet50 | VGG-19 |
| ECE Score | 0.0173 | 0.0323 | 0.0310 | 0.1376 | 0.2290 | 0.0763 |
| Threshold | 0.0077 | 0.0052 | 0.0144 | 0.0101 | 0.0068 | 0.0055 |
| $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ |
| MCE Score | 0.2884 | 0.1874 | 0.0776 | 0.3403 | 0.2959 | 0.1578 |
| Threshold | 0.2180 | 0.7257 | 0.0246 | 0.0473 | 0.0107 | 0.0106 |
| $H_0$ | Reject $H_0$ | Accept $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ |
| ECD Score | 0.1485 | 0.1893 | 1.1259 | 1.2198 | 0.5549 | 0.8256 |
| Threshold | 0.1231 | 0.0656 | 1.0507 | 0.4244 | 2.6640 | 1.2763 |
| $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Reject $H_0$ | Accept $H_0$ | Accept $H_0$ |

**Table 3**
Multiclass Classification Hypothesis Test Results calibrated with Temperature Scaling.

and 4 show the reliability diagrams for each model, both calibrated and uncalibrated. These reliability diagrams use a top-label prediction approach, where the most confident prediction is chosen.

Table 2 shows the results for the uncalibrated probabilities for the CIFAR-10, CIFAR-100 and ImageNet datasets. The ECE metric rejected the null hypothesis on every test, even though some of the scores were quite low. On the other hand, the MCE and ECD metrics accepted some null hypotheses, with the most surprising being the ECD ImageNet ResNet50 result, which has a high threshold value. However, Figure 4a shows that the majority of probabilities are actually under-confident, and therefore the acceptance of the null hypothesis makes sense. Similarly, VGG-19 ImageNet was rejected because of its over-confidence as seen in Figure 4b.

Table 3 shows the results for the datasets that have undergone a temperature scaling post-hoc re-calibration. The results remain relatively similar, except for the fact that ECD has accepted the null hypothesis for both ImageNet scores, while ECE has continued to reject them. While this shows that there is statistically significant miscalibration in the model's probabilities, it also does not seem to show that there is enough evidence for the ECD to deem that the probabilities are over-confident. In fact, Figure 4b shows under-confidence, explaining why the null hypothesis was accepted. This furthers the theory that safe calibration and perfect calibration should not be assumed to be one and the same.

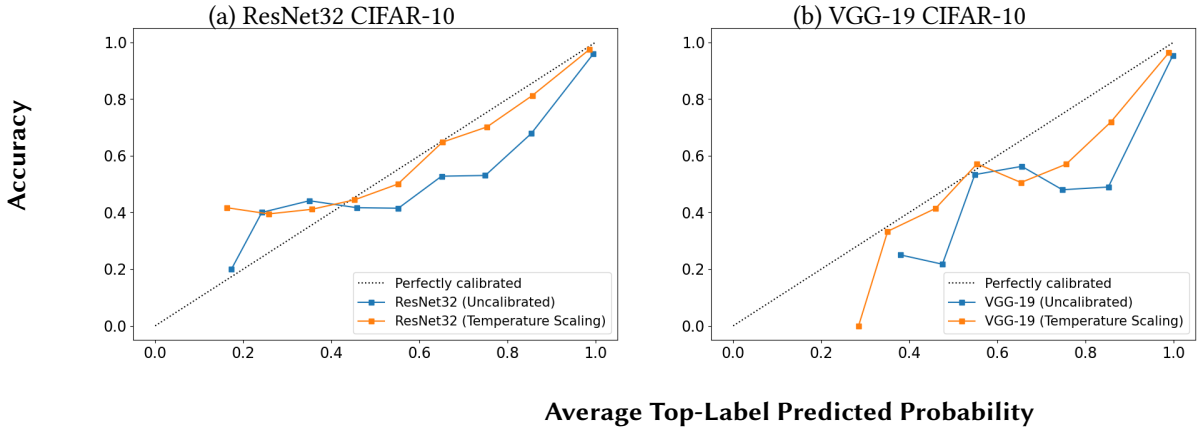


**Figure 2:** Multiclass dataset reliability diagrams for the CIFAR-10 dataset models. (a) is the ResNet32 model and (b) is the VGG-19 model. Top-Label (most confident) probability is used. $x$-axis is the average Top-Label probability. $y$-axis is the fraction of correct predictions.

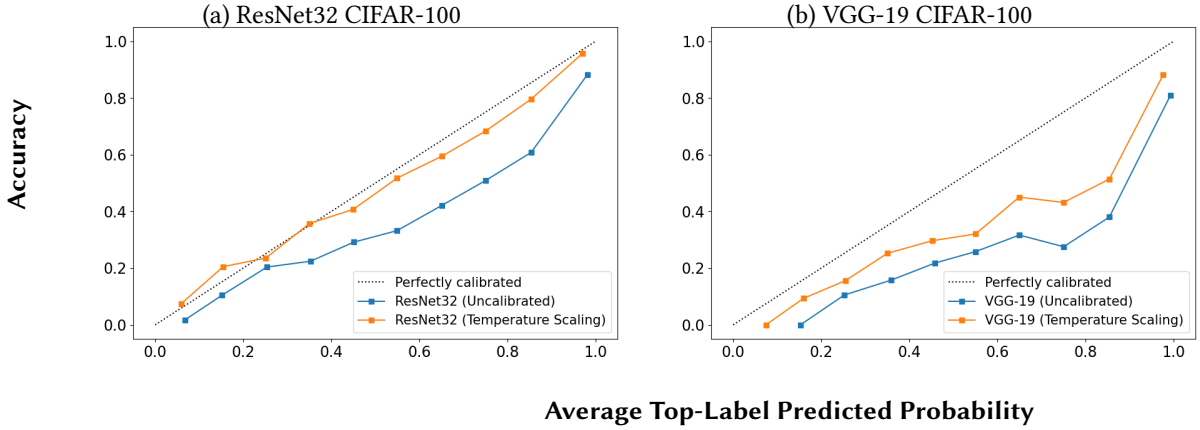


**Figure 3:** Multiclass dataset reliability diagrams for the CIFAR-100 dataset models. (a) is the ResNet32 model and (b) is the VGG-19 model. Top-Label (most confident) probability is used. $x$-axis is the average Top-Label probability. $y$-axis is the fraction of correct predictions.

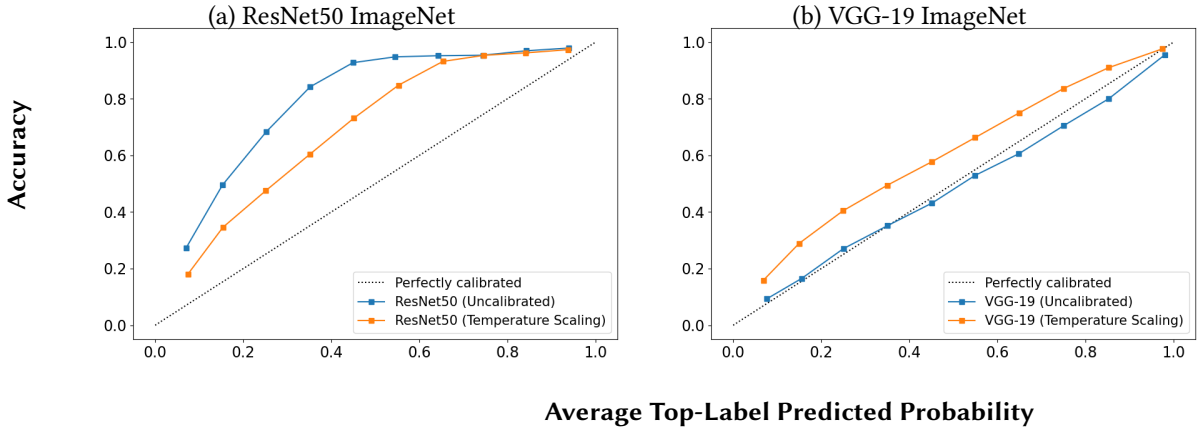


**Figure 4:** Multiclass dataset reliability diagrams for the ImageNet dataset models. (a) is the ResNet50 model and (b) is the VGG-19 model. Top-Label (most confident) probability is used. $x$-axis is the average Top-Label probability. $y$-axis is the fraction of correct predictions.