Calibration via Post-Processing Methods

Ralf Kinkel
Technical University of Munich
ralf.c.kinkel@gmail.com

Christian Tomani
Technical University of Munich
christian.tomani@tum.de

Abstract

The role of accuracy as a metric for quality of classifiers tends to overshadow other considerations. Calibration, a measure for how well the estimated probabilities of a model represent the true likelihood of classes, is one such metric which is highly important for many real-world applications. We give some historical context outlining important developments for calibration in machine learning and show different approaches for it. We then explain and compare several methods for the post-processing approach of calibration and discuss which methods are preferable under specific circumstances.

1. Introduction

Thanks to ever increasing accuracy neural networks and other machine learning models today are part of many decision-making pipelines like medicine [3], autonomous driving, and weather forecasting [7]. Well-calibrated confidence[1] is important in these applications, to be able to rely on the outputs for safety-critical decisions. In medicine low confidence can lead to a human expert being consulted or additional tests being conducted and in autonomous driving to give control back to the driver. A user of a weather forecasting app may not only be interested in the most likely weather event, but the predicted chances for events like storms, and other hazards to be as exact as possible. Interpreting the outputs and combining results of multiple models are additional tasks where calibration of confidence is important [3].

We will first define metrics and explain relevant concepts. We then lay out some of the history of classification in neural networks, the introduction of negative log likelihood and how calibration and accuracy are represented in its structure. Then we will show the problems that arise with modern neural networks regarding calibration. The main part of the report consists of descriptions, discussions and comparisons of post-processing calibration methods. It concludes with recommendations for method choice and some final remarks. The scope is limited to probabilistic classification models[2].

2. Definitions and setup

Let X be the input space and Y be the label space. Let $x \in X$ and $y \in Y$ be random variables denoting the input and label, given by an unknown joint distribution. Let f be a classification model and the confidence of the model depending on the input $p = f(x)$.

2.1. Binary setting

In the binary setting $y = \{0, 1\}$, we have a model that maps x to the confidence of the model that $y = 1$, $f : x \Rightarrow [0, 1]$. We define the expected calibration error of our model as:

$$ECE^k(f) = ||E[|f(x) - E[y|f(x)]|]||_k \qquad (1)$$

Where k is the norm of the error, if not mentioned otherwise the l1-norm is used in this report. If $ECE(f) = 0$ the model is perfectly calibrated [7]. In evaluation this can only be approximated as our validation samples are finite [13]. Another complication is that uncalibrated models[3], as well as some calibrated models have continuous confidence outputs[4] and $E[y|f(x)]$ cannot be measured for continuous $f(x)$ as there are infinitely many possible values for

[1] Estimated probability is usually called confidence in this report, or confidence scores in the multiclass setting.

[2] One could imagine the concept of calibration being used for other tasks than classification, like calibrated variance for regression.

[3] In this report "uncalibrated" means a classifier whose outputs has not been adjusted by a calibration method. If we want to convey quality of calibration, we will use terms like "well calibrated" or "poorly calibrated".

[4] Theoretically any model implemented on a computer is discrete, but the number of discrete states is so high that the output can be treated as continuous for the purposes of this analysis.

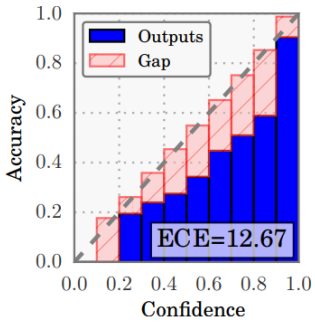


Figure 1. Reliability diagram for binary classification using 10 equal-width bins. The blue bar shows $\hat{y}$, the red bar goes up to $\hat{p}$. The gap between $\hat{y}$ and $\hat{p}$ contributes to total miscalibration, given in $ECE * 100$. Figure based on [3].

$f(x)$ and each will likely have one or no samples associated with it. This problem is similar to the difficulty of measuring mutual information between two continuous signals [7]. Empirical measurement of expected calibration error is impossible this way. To still be able to calculate expected calibration error, we use a binning scheme on the confidence output of the model first.

2.2. Evaluation of ECE with bins

We partition all samples into B bins based on the confidence of the model f. The average confidence $\hat{p}$ and the average accuracy $\hat{y}$ are computed for each bin. $\hat{p}$ and $\hat{y}$ can be used to replace $f(x)$ and $E[Y|f(X)]|]$ in (1) respectively to obtain an expected calibration error for each bin. A sample-size-adjusted mean of the bin-wise ECE forms the approximation of expected calibration error of the binned model, which is defined as:

$$ECE^k(f_{bin}) = \sum_{b=1}^{B} \frac{n_b}{N} ||\hat{y}_b - \hat{p}_b||_k \quad (2)$$

Where N is the total sample size and n_m the number of samples in the bin m.

2.3. Reliability diagrams

The bins and associated values calculated for the evaluation can be used to illustrate not only how well-calibrated a model is, but also where exactly it is overconfident or underconfident. An example is shown in Figure 1. A reliability diagram [2] for a binary classification model can be seen, where the samples are binned into 10 equal-width bins. Here the blue bar is the average accuracy $\hat{y}$ for samples in the bin, and the red bar extends upwards to the average confidence $\hat{p}$ in the associated bin. The gap between the bars represents the miscalibration, total miscalibration is given as $ECE * 100$. This model is overconfident in all bins. One can also see that no samples had an associated confidence of under 0.1 so the first bin is empty. The grey line plots the identity function and is close to perfect calibration.

2.4. Multiclass setting

In the multiclass setting $y = \{1, \ldots, C\}$ and our model $f : X \Rightarrow [0,1]^C$ outputs a confidence for each class in $[C]$. Here calibration measures are more varied. We use two metrics for multiclass calibration in this report based on [7]. We define the top-label calibration error as:

$$TCE(f) = E[|maxf(x) - P(y = argmaxf(x)|maxf(x))|] \quad (3)$$

This metric considers how well calibrated the model is for the class-prediction with the highest confidence. A model may satisfy this metric very well but be highly miscalibrated in the other outputs. To identify how well calibrated a model is for all its predictions we define the class-wise calibration error:

$$CCE(f) = \sum_{c=1}^{C} w_c E[|f(x)^{(c)} - P(Y = c|f(x)^{(c)})|] \quad (4)$$

Here we compute a weighted average of miscalibration for each class. In general, this metric is higher and more difficult to minimize than TCE. w_k could be the proportion of samples with labels of the specific class but could also be adjusted for importance. Calibration for the class "storm" may be more important than calibration for "cloudy" for example. The transition to the binned version of these measures is done similarly as in the binary case, by binning the results based on confidence and using average confidence $\hat{p}$ and average accuracy $\hat{y}$ for each bin. Which of these measures is more important depends on the specific application.

Imagine we have a weather forecast with confidence outputs 0.5 sunny, 0.4 clouds, 0.1 rain, 0.0 storm. Even if the top-prediction is perfectly calibrated, we would still care about miscalibration in the other classes, for example if the real probabilities were 0.5 sunny, 0.1 clouds, 0.1 rain, 0.3 storm instead, one may want to postpone plans for outside activities. Requirements for calibration of models that predict the winner in winner-takes-all competitions on the other hand, would be mostly satisfied even if only the top-label is well calibrated.

Note that for TCE and CCE one can use norms like in (1). If confidence scores p are continuous, one can calculate TCE and CCE for the binned model like in Section 2.2.

Dataset	Uncal.	Cal. 1	Cal. 2
D1	0.4	0.4	0.2
D2	0.2	0.1	0.2
SMRE	0.32	0.29	0.2
ME	0.3	0.25	0.2
RMSE	0.29	0.23	0.2

Dataset	Uncal.	Cal. 3	Cal. 4
D3	0.8	0.8	0.7
D4	0.2	0.1	0.2
SMRE	0.58	0.57	0.51
ME	0.5	0.45	0.45
RMSE	0.45	0.37	0.41

Table 1. Comparisons of calibrators for the same model across different datasets with $RMSE$, ME and $SMRE$ calculated for all calibrators. **Left:** The calibrators half CE for one dataset but do not affect CE for the other dataset. **Right:** The calibrators reduce CE by a fixed amount for one dataset but do not affect CE for the other dataset.

2.5. SMRE

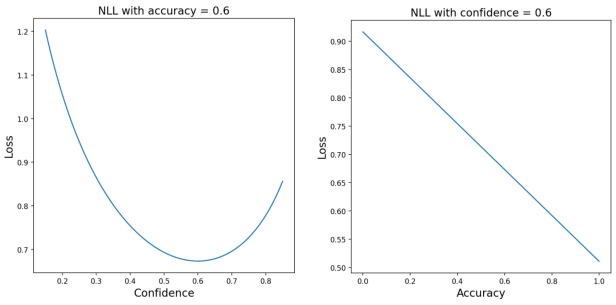
To compare performance regarding TCE and CCE of calibration methods across different datasets E in Section 5.2, we use the square-mean-root error (SMRE) for any calibration error CE.

$$SMRE = (\sum_{e=1}^{|E|} \frac{\sqrt{CE_e}}{|E|})^2 \tag{5}$$

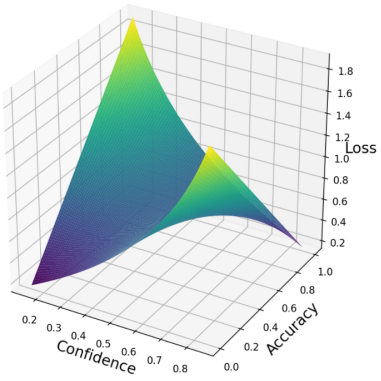
A generally more popular metric is the root-mean-square error (RMSE) which is more prone to outliers or the regular mean error (ME) which does not adjust for outliers. The use of $SMRE$ is specifically motivated by the fact that it is less prone to outliers. A justification is given here and supported by Table 1:

- Reducing CE by half for a model in a highly miscalibrated dataset is more valuable than reducing CE by half in a well calibrated dataset. This is captured by $RMSE$, ME and $SMRE$, as can be seen in Table 1 (Left), where calibrators half CE for one dataset but do not affect CE for the other dataset. Calibrator 2 which halves CE for the more miscalibrated dataset is preferred by all metrics.

- Reducing CE by a certain amount is generally more valuable in a well-calibrated dataset than in a poorly calibrated one. If one had an uncalibrated model and measured CE for two datasets as 0.8 and 0.2 one would arguably rather reduce the second by half to 0.1, than the first by 0.1 to 0.7. As shown in Table 1 (Right), calibrator 3, which leads to the higher percentage reduction with same absolute reduction is only preferred by $SMRE$.



(a) (b)



(c)

Figure 2. Visualizations for NLL in binary classification. **(a):** Accuracy is fixed $P(y) = 0.6$, loss is minimized for $p = P(y)$. **(b):** Confidence is fixed $p = 0.6$, loss is minimized for $P(y) = 1$. **(c):** NLL loss over accuracy and confidence in 3D-plot. High accuracy and well calibrated confidence leads to low loss.

3. History and related work

3.1. NLL-loss

The outputs of many machine learning models for classification like Support Vector Machines (SVM) are real numbers that do not represent probabilities and are hard to interpret [15]. The introduction of the sigmoid function allowed one to use logits for calculating confidence in the binary classification setting $p = \sigma(z)$. The SoftMax function extended this to the multiclass setting $p = \sigma_{SM}(z)$.

In the training of neural networks for classification, negative log likelihood, also known as cross entropy, became the standard loss function with which one could optimize these confidence scores. We define Negative Log Likelihood (NLL) depending on data D consisting of tuples of labels y and confidence scores p as:

$$NLL(D) = -\sum_{d=1}^{|D|}(y_d * ln(p_d)) \tag{6}$$

Its structure penalizes low accuracy and high miscalibration which is illustrated in Figure 2. The illustrations show negative log likelihood in the binary classification setting. When the probability $P(y)$ is fixed, NLL is minimized if confidence equals probability $f(x) = p = P(y)$. In this way minimization of NLL-loss increases calibration. When the confidence $p > 0.5$ is fixed, NLL is minimized when the probability of the associated label is maximized. Intuitively this means whenever the model is over 50% confident, the loss is lower when the label is 1 than when it is 0. In this way minimization of NLL-loss increases accuracy. In summary NLL is lowest when confidence and probability are as similar as possible and as far from 0.5 (being complete uncertainty) as possible. Therefore, minimization via backpropagation promised to lead to high accuracy and low miscalibration.

From this analysis one could expect models trained with NLL to be well calibrated and earlier analyses confirmed this intuition for neural networks [12].

3.2. Miscalibration in modern neural networks

Neural networks with modern architectures[5] were found to be much more poorly calibrated than less recent ones as shown by [3]. Some proximate causes have been identified, but there is no ultimate explanation yet for why exactly these should lead to miscalibration. An overview of proximate causes is given here.

- Model capacity: Model capacity has increased by orders of magnitudes in the last decades. Models with hundreds of layers [3] and hundreds of millions of parameters are getting more common and the trend shows no sign of stopping. While model capacity increases accuracy, it seems to worsen calibration as shown in Figure 3 (a) and Figure 3 (b).
- Batch normalization: Batch normalization has been a recent development that leads to faster training, better accuracy and has a regularizing effect [3]. It is widely deployed in modern neural networks but also seems to worsen calibration as shown in Figure 3c.
- Weight decay: Weight decay is still an important regularization technique which is widely used but the magnitude has been decreased severely as consequence of the regularizing effect of batch normalization [3]. This development increases miscalibration because higher weight decay has a positive impact on calibration in general, as shown in Figure 3d.

All these recent developments add up to severe miscalibration in modern neural networks which must be corrected if one wants to deploy them in safety critical decision-making pipelines.

3.3. Other related work

To obtain well calibrated confidence scores one can take different general approaches: For example one can penalize overconfidence with an additional regularization term [14], use ensembles of models to obtain uncertainty estimates [8, 23], or use additional data augmentation [17, 19]. One can also take an already trained model and adjust its confidence outputs.

In this report we will focus on the last approach called calibration via post-processing or recalibration [3] and on a small selection of popular methods and extensions, many of which have been milestones in the domain of post-processing calibration. It is important to keep in mind that there are a number of more recent promising recalibration methods that are not covered, like splines [4], gaussian processes [20] and prediction specific temperature [18].

One can also choose to represent model uncertainty differently: Bayesian neural networks for example use probability distributions to represent model uncertainty [9].

4. Post-processing methods

A distinction can be made here between parametric and non-parametric methods [18]. Parametric methods adjust confidence p by applying a recalibration function r with a fixed set of parameters, where assumptions about the form of this function are necessary for this to be possible. Non-parametric methods also apply some function but make no or few assumptions about the form and are not limited to one specific set of parameters. We obtain a recalibrated confidence after application of post-processing methods $q = r(p)$.

The general approach for post-processing calibration is to take an already trained classifier and then use an unseen recalibration dataset to train r in the case of parametric methods or create r in the case of non-parametric methods. We define D as data for the recalibration dataset, containing labels y and associated confidence scores p calculated by the classifier.

[5] The term "modern" is somewhat ambiguous here, it means neural networks with features like batch normalization and skip connections. There is some recent evidence [10] that the even more "modern" neural networks, transformer like architectures, are not that poorly calibrated anymore. When this report refers to "modern neural networks" architectures like ResNet are meant, not transformers.

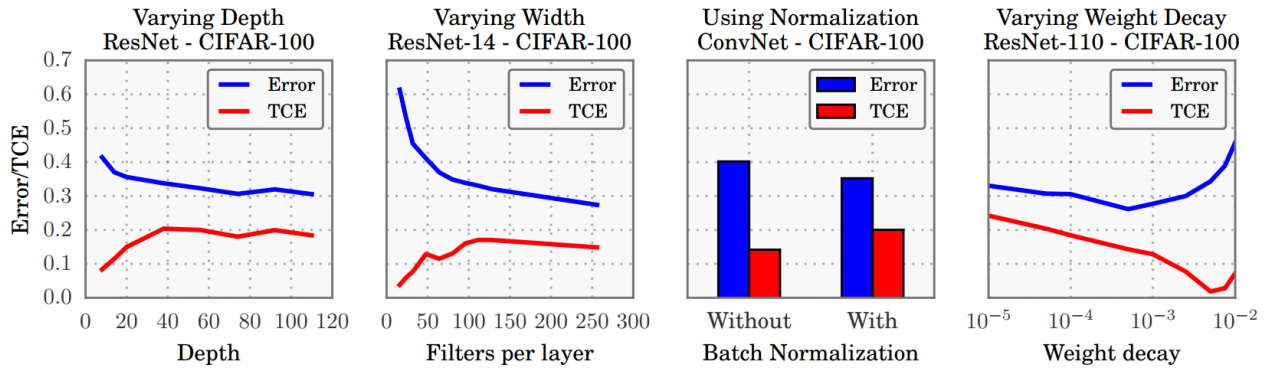


Figure 3. From left to right: a, b, c, d. **(a,b):** Miscalibration and error in respect to model capacity. **(c):** to batch normalization. **(d):** to weight decay. Measured in TCE (lower is better). Figure based on [3].

4.1. Non-parametric methods

For the sake of simplicity and conciseness the descriptions of the methods are for the binary setting first and are later extended into the multiclass setting.

Histogram binning [21] Histogram binning and its variants are non-parametric calibration methods that use the binning scheme described in Section 2.2 not only for the evaluation of calibration, but as a method of recalibration itself. In histogram binning each of the mutually exclusive bins is assigned a score θ_b. At test time any uncalibrated confidence prediction p fits in one bin designated by boundaries a_b to a_{b+1} and is assigned the score of this bin as recalibrated confidence $q = r(p) = \theta_b$. Boundaries and scores must be defined to use this method. The boundaries are mostly chosen to be of equal-width, or uniform-mass, meaning all bins contain equal numbers of samples. The scores are chosen to minimize the bin-wise squared loss:

$$\min_{\theta_1,\ldots,\theta_B} \sum_{b=1}^{B} \sum_{i=1}^{n} \mathbf{1}(a_b \leq p_i < a_{b+1})(\theta_b - y_i)^2 \qquad (7)$$

Where $\mathbf{1}$ is the indicator function, n is the number of samples and B is the number of bins. This results in the score being the average number of positive-class samples in the bin $\theta_b = \hat{y}$ [3]. An example can be seen in Figure 5 (Middle).

An advantage of binning for recalibration is that the recalibrated confidence is already discrete, this way we can approximate calibration errors directly. If we have continuous confidence outputs we still have to still bin for evaluation as described in Section 2.2. Additional problems can arise from continuous confidence scores as described in Section 5.1.

Isotonic regression [22] Isotonic regression is a generalization of histogram binning where number of bins B, boundaries A and scores Θ are jointly optimized. The name "isotonic" comes from an additional limitation to histogram binning that the scores must be monotonically increasing:

$$\theta_1 \leq \theta_2 \leq \ldots \leq \theta_B \qquad (8)$$

Bayesian binning by quantiles [11] Bayesian Binning into Quantiles (BBQ) is a Bayesian generalization of histogram binning where a space S of all possible binning schemes is considered. Bayesian averaging is performed on the adjusted confidence scores produced by each scheme to generate the calibrated confidence q [3].

$$q = P(y = 1|p, D) = \sum_{i=1}^{|S|} P(s_i|D) * P(y = 1|p, s_i) \qquad (9)$$

with

$$P(s_i|D) = \frac{P(s_i)*P(D|s_i)}{P(D)} = \frac{P(s_i)*P(D|s_i)}{\sum_{j=1}^{|S|} P(s_i)*P(D|s_i)}$$

Here the prior P(s) is assumed to be uniform, leading to:

$$P(s_i|D) = \frac{P(D|s_i)}{P(D)} = \frac{P(D|s_i)}{\sum_{j=1}^{|S|} P(D|s_i)}$$

$P(y = 1|p, s_i)$ in (9) can be calculated by the simple application of binning scheme s_i as described in "Histogram binning" Section 4.1. The schemes in S can be described by number of bins, boundaries and scores: $s = \{B, A, \Theta\}$. There is one scheme in S for each number of bins B, constrained to a specific range. The associated boundaries are created with uniform-mass binning. The scores are determined by assuming a prior over Θ. The prior is a beta distribution with θ_i being the midpoint of boundaries. The posterior for scores Θ is then calculated with D. This way we get the scores Θ together with the chance of data given a scheme $P(D|s_i)$.

There are multiple additional variants and extensions to these binning methods.

Binning in the multiclass setting A common way of using binning methods in the multiclass setting is treating them as C one-versus-all problems [22]. We can assign a label as $\mathbf{1}(y_i = c)$ and the uncalibrated confidence as $p^{(c)}$. At test time our C classwise calibrators return unnormalized confidence scores for each sample $[q_i'^{(1)}, \ldots, q_i'^{(C)}]$. We obtain normalized confidence scores with $q = \frac{q'}{||q'||_1}$. Note that the class prediction could change here $argmax(q) \neq argmax(p)$, if no additional restrictions are defined.

4.2. Parametric methods

The parametric methods shown in this section are Platt scaling and its extensions to the multiclass settings and Dirichlet calibration, a method created specifically for the multiclass setting.

Platt scaling [15] Platt scaling is a parametric approach applicable in the binary setting. It does not use the confidence scores p of the original classifier by default but the logits z. Those are turned into calibrated confidence scores with a simple adjustment before applying the sigmoid function: $q = \sigma(a * z_i + b)$. a and b are the only additional parameters here. They are optimized by using the NLL loss while the parameters of the original model are frozen [3]. An illustration can be seen in Figure 5 (Left).

Extensions to multiclass setting [3] In the multiclass setting the logits z are in a vector, with one entry for each class. There are 3 common variants for platt scaling in the multiclass setting: Matrix scaling, vector scaling and temperature scaling. A linear transformation is applied on the logits in all cases and then the softmax function is applied to get calibrated confidence scores q:

$$q = \sigma_{SM}(Wz + b) \tag{10}$$

- **Matrix scaling** W is not restricted in the matrix scaling variant. This way W has C^2 parameters and b has C parameters.
- **Vector scaling** Vector scaling is a simpler variant where W is restricted to being a diagonal matrix. W has C parameters and b has C parameters here.
- **Temperature scaling** Temperature scaling is a further simplification of vector scaling, where W is restricted to being a positive multiple of the identity matrix and $b = 0$. This way the whole method is determined by a single parameter. Temperature scaling can also be parametrized as:

$$q = \sigma_{SM}(z/T) \tag{11}$$

Where T is the temperature and $T > 0$. If $T = 1$ we have no change from the original confidence scores. If $T < 1$ certainty is increased[6], and if $T > 1$ certainty is decreased.

Note that the class prediction can change for matrix scaling and vector scaling but is guaranteed to stay the same for temperature scaling, as the division of logits by a positive value is order-preserving [16]. This is a feature described more closely in Section 5.1.

Dirichlet calibration [6] Dirichlet calibration is a parametric approach applicable to the multiclass setting. The parametrization is:

$$q = \sigma_{SM}(W ln(p) + b) \tag{12}$$

One can see that Dirichlet calibration is very similar to matrix scaling with the difference that the inputs are the uncalibrated confidence scores instead of the logits and that the natural logarithm is applied before the linear transformation. The use of uncalibrated confidence scores as input is motivated by the fact that all probabilistic classifiers have those as output, but not all use logits as intermediate step which is discussed in more detail in Section 5.1.

4.3. Regularization of parametric methods

Methods with a high number of parameters are prone to overfitting which must be counteracted for the method to generalize well to unseen data. Out of the parametric methods presented here, overfitting is especially a problem for matrix scaling and Dirichlet calibration [6] as their parameter count roughly scales with the square of classes. One popular approach for counteracting overfitting is regularization, where more complex hypotheses are penalized by introducing an additional loss term increasing with the magnitude of the parameters.

L2 regularization L2 regularization is one of the simplest regularization techniques. Here one additional term is defined and added to the loss function:

$$L_2 = \lambda \sum w_{ij}^2 \tag{13}$$

It appears to perform well for post-processing calibration of classifiers with the exception of neural networks [6].

ODIR regularization [6] ODIR (Off-Diagonal and Intercept Regularization) is a slightly more complex regularization technique, specifically designed to perform well for

[6] "Higher certainty" meaning that confidence scores are closer to 0 or 1.

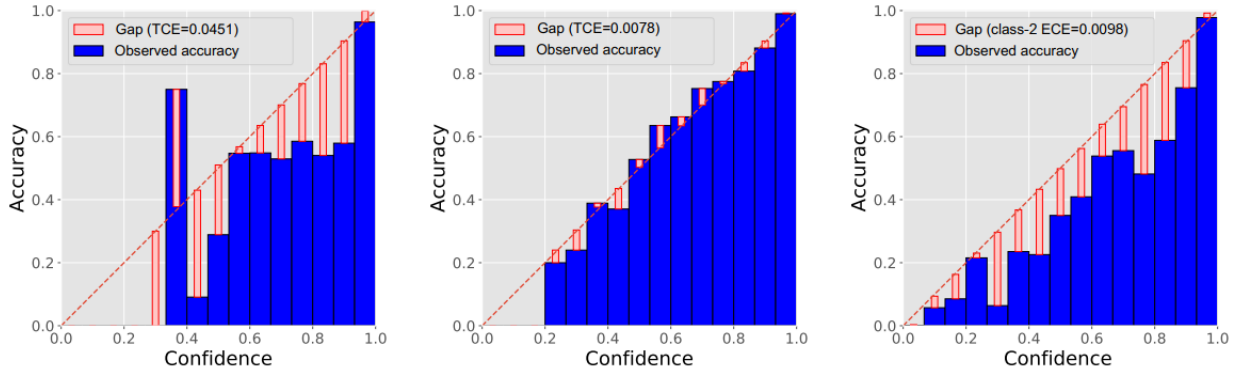


Figure 4. Reliability diagrams for a ResNet trained on CIFAR-10, figure based on [6]. **Left:** Uncalibrated classifier top-label confidence. **Middle:** Top-label confidence after temperature scaling. **Right:** Class 2 calibration after temperature scaling.

calibration of neural networks.

$$ODIR = \lambda(\frac{1}{k(k-1)} \sum_{i \neq j} w_{ij}^2) + \mu(\frac{1}{k} \sum b_j^2) \qquad (14)$$

It can be viewed as a variant or extension of L2 regularization, where the diagonals of the weight matrix W are excluded from the regularization term and another term is added for penalizing bias b. The fractions containing k are not strictly necessary for implementation but make good values for λ and μ easier to find across classication problems with varying number of classes.

4.4. Other methods

Scaling-binning calibrator [7] The scaling-binning calibrator is a combination of parametric and non-parametric methods. It is used by first applying a parametric scaling method $q' = g(z)$ and then a non-parametric binning scheme on the adjusted confidence scores to get the calibrated confidence scores $q = s(q')$. It is trained in 3 steps where recalibration dataset T is split into 3 sets: T_1, T_2, T_3.

- The parametric scaling method is trained on T_1 by minimizing the squared loss:

$$g = \underset{g \in G}{argmin} \sum_{(z,y) \in T_1} (y - g(z))^2 \qquad (15)$$

  With G being the space of possible scaling methods, for example all vector scalings.

- Bin boundaries A are then formed by applying uniform-mass binning on the adjusted confidence $q' = g(z)$ of samples in T_2.

- Finally, samples in T_3 are used to define scores theta Θ for the bins, by finding the mean of all q' falling in the corresponding bins.

An illustration of the scaling-binning calibrator can be seen in Figure 5 (Right). The main advantage of this method is that one can benefit from the efficiency and performance of scaling methods while still generating discrete outputs, making evaluation much easier as described in Section 2.2.

5. Comparison of methods

There are different strengths, weaknesses, features, and application areas for the methods presented in Section 4, which we will explain here. The comparison and concepts relate to the multiclass setting, as that is the more common, general and complex setting. Evaluation of methods for the binary setting is trivial once the intricacies of the multiclass setting are understood.

5.1. Additional concepts

We define some new metrics for comparisons, some of which have already been alluded to.

Accuracy-preservation The application of many methods presented here can lead to changes in class-predictions if no additional constraints are set. In general, the original model was likely trained to maximize accuracy, so that changes of class-predictions often lead to lower accuracy. Therefore accuracy-preservation is a useful attribute of post-processing calibration methods.

Applicability The possible inputs of the methods presented are confidence scores p and logits z. If the inputs are confidence scores p the method can be applied to any probabilistic classifier. Some probabilistic models like random forests [1] do not use logits z as intermediate step in the classification process, therefore methods that need logits as input are unsuited for calibrating them.

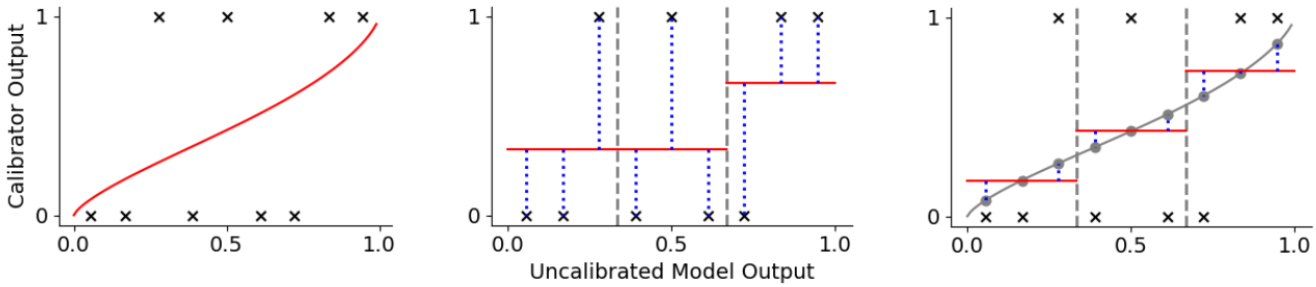


Figure 5. Calibration with 9 samples. Black crosses are labels, red line is $q(p)$. **Left:** Calibration with parametric scaling method. **Middle:** Calibration with non-parametric binning method. **Right:** Calibration with scaling-binning calibrator. Figure taken from [7].

Measurable miscalibration As shown in Section 2.2, we cannot use empirical measurements directly to evaluate miscalibration if our calibrated confidence scores q are continuous, we must bin them first. This binning is only done for evaluation here, not for recalibration as in histogram binning which is an important distinction as binning for evaluation can lead to an underestimation of miscalibration [7]. The reason for possible underestimation is that errors at each side of the bin can cancel each other out [7]. This gap between calibration errors $CE(f)$ and $CE(f_b)$ should be less pronounced the more bins one creates, which is illustrated in Figure 6. One cannot use an arbitrary high number of bins B though, as the fewer samples there are in each bin the higher the variance. This way even a perfectly calibrated model could appear highly miscalibrated if one picks too many bins and a severely miscalibrated model could appear well calibrated if one evaluates on too few bins. Therefore, methods that output discrete confidence scores are preferable all else being equal, as the miscalibration can be measured for them directly.

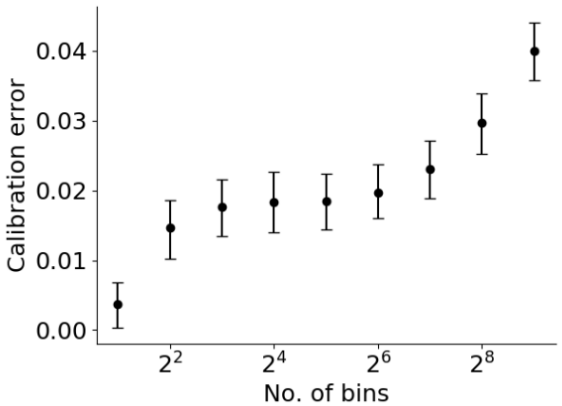


Figure 6. Estimated calibration error depending on number of bins for a recalibrated ImageNet classifier. Figure taken from [7].

Complexity We add another term in the comparison that tries to capture a general sense of complexity of the methods. Factors that may influence this measure are how complex a method is conceptually, how much time is needed to train and run the method, number of parameters and need for regularization, and steps needed in the implementation. Weighting and combining these criteria is very subjective of course and not done explicitly here. Lower complexity is preferable, all else being equal.

5.2. Summary of comparison

A rough comparison of the methods is shown in Table 2, which is based on the discussions in Section 5.3 and Section 5.4. We compare metrics that are binary: accuracy-preservation, applicability, and measurability of miscalibration. We also compare metrics that exist on a scale: complexity, TCE and CCE. TCE and CCE are sometimes hard to compare along different papers and experiments or there is no measure available, in this case we will point out assumptions that we make for the comparisons. If the methods are applicable to all probabilistic classifiers, TCE and CCE for neural networks and other classifiers are discussed separately. Otherwise TCE and CCE are compared only for neural networks.

Note that for many methods there are extensions or additional variants that change these metrics. We only compare the standard methods described in Section 4.

5.3. Metrics for non-parametric methods

Histogram binning Histogram binning has measurable miscalibration as its outputs are the discrete scores Θ. It is usually not accuracy preserving but is applicable to all probabilistic classifiers as its inputs are the confidence scores of the original model p. It is the least complex non-parametric model presented, but still relatively complex compared to many parametric methods. TCE and CCE are comparatively high in classifiers other than neural networks (Table 4 and Table 5). TCE is moderately high in neural net-

Method	Acc. Pres.	Applicability	Meas. Misc.	Complexity	TCE-neural	CCE-neural	TCE-else	CCE-else
Hist. bin	No	Yes	Yes	Medium	Medium	*Medium*	High	High
Isot	No	Yes	Yes	High	Medium	*Medium*	Low	Low
BBQ	No	Yes	Yes	Very high	Medium	*Medium*	*Low*	*Low*
TempS	Yes	Yes	No	Very Low	Low	Medium	High	High
VecS	No	No	No	Low	Low	Low	-	-
MS-(ODIR)	No	No	No	Medium	Medium	Low	-	-
Dir-(ODIR/L2)	No	Yes	No	Medium	Medium	Low	Low	Low
Scaling-Bin	No	*	Yes*	+	*	*	*	*

Table 2. Comparison of methods presented in section 4. Non-parametric methods based on binning are shown in blue, parametric scaling methods are shown in yellow. Scaling-binning calibrator is shown in green. Entries in cursive are based only on assumptions as no data was found. Dirichlet calibration is L2-regularized for the neural network recalibration and ODIR-regularized for recalibration of other models. Metrics for the scaling-binning calibrator depend mostly on the scaling function used. It can be described as a tradeoff where complexity is increased but miscalibration becomes measurable.

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	0.0919	0.0434	0.0522	0.0412	**0.0185**	0.0300	0.2113
Cars	ResNet 50	0.0430	**0.0174**	0.0429	0.0184	0.0235	0.0237	0.1050
CIFAR-10	ResNet 110	0.0460	0.0058	0.0081	**0.0054**	0.0083	0.0088	0.0100
CIFAR-10	ResNet 110 (SD)	0.0412	0.0067	0.0111	0.0090	**0.0060**	0.0064	0.0072
CIFAR-10	Wide ResNet 32	0.0452	0.0072	0.0108	0.0074	**0.0054**	0.0060	0.0072
CIFAR-10	DenseNet 40	0.0328	0.0044	0.0061	0.0081	**0.0033**	0.0041	0.0041
CIFAR-10	LeNet 5	0.0302	0.0156	0.0185	0.0159	**0.0093**	0.0115	0.0116
CIFAR-100	ResNet 110	0.1653	0.0266	0.0499	0.0546	**0.0126**	0.0132	0.2549
CIFAR-100	ResNet 110 (SD)	0.1267	0.0246	0.0416	0.0358	0.0096	**0.0090**	0.2009
CIFAR-100	Wide ResNet 32	0.1500	0.0301	0.0585	0.0577	**0.0232**	0.0257	0.2444
CIFAR-100	DenseNet 40	0.1037	0.0268	0.0451	0.0359	0.0118	**0.0109**	0.2187
CIFAR-100	LeNet 5	0.0485	0.0648	0.0235	0.0377	**0.0202**	0.0209	0.1324
ImageNet	DenseNet 161	0.0628	0.0452	0.0518	0.0351	**0.0199**	0.0224	-
ImageNet	ResNet 152	0.0548	0.0436	0.0477	0.0356	**0.0186**	0.0223	-
SVHN	ResNet 152 (SD)	0.0044	**0.0014**	0.0028	0.0022	0.0017	0.0027	0.0017
20 News	DAN 3	0.0802	**0.0360**	0.0552	0.0498	0.0411	0.0461	0.0910
Reuters	DAN 3	0.0085	0.0175	0.0115	0.0097	0.0091	**0.0066**	0.0158
SST Binary	TreeLSTM	0.0663	0.0193	**0.0165**	0.0227	0.0184	0.0184	0.0184
SST Fine Grained	TreeLSTM	0.0671	0.0209	**0.0165**	0.0261	0.0256	0.0298	0.0239
	SMRE	0.0594	0.0210	0.0263	0.0235	0.0135	0.0150	0.0629

Table 3. Comparison of TCE for recalibration methods using different classifiers and datasets. Lower rank is better. Table based on [3].

work classifiers (Table 3). We have not found comparisons of CCE for neural network classifiers. We assume it is moderately high as well because the binning is done class-wise, therefore there should not be a significant difference between the top-label calibration and the calibration of the other classes. Note that equal-width binning seems to be superior to uniform-mass binning, at least for classifiers other than neural networks (Table 4 and Table 5).

Isotonic regression Isotonic regression also has measurable miscalibration, is usually not accuracy preserving and applicable to all probabilistic classifiers. It is more complex than histogram binning as not only the scores, but also the number of bins B and the bin boundaries A are part of the optimization. TCE and CCE are low in classifiers other than neural networks (Table 4 and Table 5). TCE is moderately high in neural networks (Table 3). We have not found CCE for neural networks, but we assume it is moderately high as well.

Bayesian binning into quantiles BBQ is not accuracy preserving but applicable to all probabilistic classifiers and has measurable miscalibration. It is a very complex method, as it is hard to implement [3], conceptually the most difficult method described in this report and takes roughly two orders of magnitude more time to run compared to histogram

	DirL2	Beta	FreqB	Isot	WidB	TempS	Uncal
adas	**1.7**	2.7	4.3	2.7	4.2	6.0	6.4
forest	4.2	2.2	5.7	**1.4**	4.4	5.1	5.1
knn	3.0	**3.0**	6.1	3.5	5.8	3.3	3.3
lda	**2.0**	2.9	5.9	2.1	4.0	5.7	5.5
logistic	2.2	3.0	6.3	**1.9**	4.7	3.8	6.1
mlp	3.5	2.7	6.6	**1.4**	5.7	4.0	4.2
nbayes	**2.1**	2.8	5.2	2.4	4.3	5.3	5.9
qda	3.1	2.3	6.5	**1.7**	4.7	4.6	5.1
svc-linear	2.7	2.8	6.7	**2.0**	4.9	3.4	5.5
svc-rbf	3.7	3.4	6.5	2.9	4.5	**2.7**	4.3
tree	2.6	3.6	6.8	4.8	5.7	**2.2**	2.3
avg rank	2.80	2.86	6.05	**2.42**	4.81	4.17	4.89

Table 4. Comparison of TCE for recalibration methods across different machine learning classfiers other than neural networks. "FreqB" and "WidB" refers to histogram binning with uniform-mass or equal-width. Lower rank is better. Table based on [6].

	DirL2	Beta	FreqB	Isot	WidB	TempS	Uncal
adas	**1.9**	3.2	4.3	4.3	4.1	5.0	5.1
forest	4.0	2.1	5.8	**1.1**	4.0	5.5	5.4
knn	4.0	3.9	6.0	3.6	5.6	2.5	2.5
lda	2.4	2.8	5.8	**2.0**	4.1	5.3	5.7
logistic	2.2	2.5	6.2	**2.0**	4.4	4.5	6.1
mlp	3.0	2.3	6.6	**1.7**	5.5	4.3	4.5
nbayes	**1.9**	3.5	5.0	2.5	4.0	5.4	5.7
qda	2.7	2.6	6.4	**1.8**	4.6	5.0	4.9
svc-linear	2.5	2.6	6.7	**2.5**	4.6	3.6	5.5
svc-rbf	**2.7**	2.9	6.5	3.1	4.5	3.6	4.7
tree	3.1	4.1	6.5	4.7	5.5	**1.9**	2.0
avg rank	2.76	2.95	5.97	**2.67**	4.65	4.25	4.75

Table 5. Comparison of CCE for recalibration methods across different machine learning classfiers other than neural networks. "FreqB" and "WidB" refers to histogram binning with uniform-mass or equal-width. Lower rank is better. Table based on [6].

binning [3]. TCE is moderately high for neural networks (Table 3), CCE is assumed to be moderately high as well. Measures for other models are hard to compare as BBQ has not been a part of the experiments in Table 4 and Table 5. The original paper [11] suggests a performance advantage over isotonic regression but is only comparing 3 models and only in the binary setting. We will assume TCE and CCE to be roughly similar to isotonic regression, but this is a precarious assumption.

5.4. Metrics for parametric methods

Temperature scaling Temperature scaling is accuracy preserving as the logits are scaled linearly with a single parameter, this way the certainty may change, but not the order of class predictions. At first TS appears to not be applicable to all probabilistic classifiers as it uses logits z as inputs, but one can show equivalence between TS applied on z and $ln(p)$ [6], this way it is usable as a general-purpose calibration method. TS does not have measurable miscalibration, having continuous outputs. It is the least complex calibration method as it only introduces a single parameter used for division. For neural networks it has low TCE, but moderately high CCE as shown in Table 6 and Table 7 and illustrated in Figure 4. All other parametric methods introduced in Section 4 are superior in regard to classwise calibration. The proximate cause for this circumstance could be the limited capacity of temperature scaling and the order preserving nature of the scaling operation. An ultimate explanation has not been revealed yet, as far as our literature search could ascertain. TCE and CCE for other classifiers are both high (Table 4 and Table 5).

Vector scaling. Vector scaling is not accuracy preserving, does not have measurable miscalibration and is not applicable to all probabilistic classifiers as its inputs are logits z. VS is not very complex as the parameter count only scales linearly with classes, it is also usable without regularization [6]. TCE and CCE for neural networks are low as shown in Table 6 and Table 7.

Matrix scaling. Matrix scaling is not accuracy preserving, not applicable to all probabilistic classifiers and does not have measurable miscalibration. It is moderately complex as the parameter count scales quadratically with classes and regularization is necessary for good performance (On calibr). TCE is moderately high and CCE is low for neural networks as shown in Table 6 and Table 7.

Dirichlet calibration. Dirichlet calibration is applicable to all probabilistic classifiers but is not accuracy preserving and does not have measurable miscalibration. It is only slightly more complex than matrix scaling. For neural networks TCE is moderately high and CCE is low as shown in Table 6 and Table 7. TCE is and CCE for other classifiers are low[7] as shown in Table 4 and Table 5.

5.5. Metrics for other methods

Scaling-binning calibrator. The applicability depends on the used scaling function. Using temperature scaling or Dirichlet calibration in the first step, one can apply the scaling-binning calibrator to any probabilistic classifier for example. In general, it is not accuracy preserving as the order of class predictions can change. The outputs are

[7]The data of Tables 4,5,6,7 is from [6] where Dirichlet calibration has first been proposed and original papers tend to be slightly biased towards methods they propose [5]. An example can be seen in the differences between vector scaling and temperature scaling for TCE in neural network classifiers. The paper that proposed temperature scaling [3] finds moderate improvement of temperature scaling over vector scaling (Table 3), while [6] finds almost none (Table 6). Therefore, relative performance for Dirichlet calibration may slightly decrease in a larger meta-analysis.

	Uncal	general-purpose calibrators TempS	Dir-L2	Dir-ODIR	calibrators using logits VecS	MS-ODIR
c10_convnet	0.04760	0.01065	0.00769	0.00960	**0.00740**	0.00782
c10_densenet40	0.05500	0.00946	**0.00568**	0.01097	0.01018	0.00988
c10_lenet5	0.05180	0.01665	0.01383	0.01367	**0.01310**	0.01468
c10_resnet110	0.04750	0.01132	**0.00680**	0.01086	0.01130	0.01059
c10_resnet110_SD	0.04113	**0.00555**	0.00646	0.00815	0.00579	0.00566
c10_resnet_wide32	0.04505	0.00784	**0.00524**	0.00837	0.00769	0.00727
c100_convnet	0.17614	**0.01367**	0.14347	0.02069	0.01965	0.02660
c100_densenet40	0.21156	**0.00902**	0.12380	0.01138	0.01224	0.02197
c100_lenet5	0.12125	0.01499	0.01369	0.02003	**0.01294**	0.01407
c100_resnet110	0.18480	**0.02380**	0.14535	0.02822	0.02693	0.02735
c100_resnet110_SD	0.15861	**0.01214**	0.15920	0.02283	0.01296	0.02246
c100_resnet_wide32	0.18784	**0.01472**	0.13509	0.01891	0.01718	0.02581
SVHN_convnet	0.07755	0.01179	0.01910	0.00997	**0.00934**	0.01037
SVHN_resnet152_SD	0.00862	0.00607	0.00691	**0.00582**	0.00595	0.00604
SMRE	0.08894	0.01156	0.03810	0.01356	0.01177	0.01402

Table 6. Comparison of TCE for parametric recalibration methods across different classifiers and datasets. Table based on [6].

	Uncal	general-purpose calibrators TempS	Dir-L2	Dir-ODIR	calibrators using logits VecS	MS-ODIR
c10_convnet	0.10375	0.04423	0.04262	0.04507	**0.04259**	0.04352
c10_densenet40	0.11430	0.03977	**0.03412**	0.03687	0.03609	0.03678
c10_lenet5	0.19849	0.17141	**0.05185**	0.05891	0.05705	0.05862
c10_resnet110	0.09846	0.04344	**0.03206**	0.03950	0.03653	0.03615
c10_resnet110_SD	0.08647	0.03071	0.03148	0.02937	0.02713	**0.02681**
c10_resnet_wide32	0.09530	0.04775	0.03153	0.02947	0.03164	**0.02921**
c100_convnet	0.42414	**0.22683**	0.40185	0.24041	0.24063	0.23958
c100_densenet40	0.47026	0.18664	0.32985	**0.18630**	0.18879	0.19112
c100_lenet5	0.47264	0.38481	0.21865	0.21348	**0.20293**	0.21379
c100_resnet110	0.41644	0.20095	0.35885	**0.18639**	0.19442	0.20270
c100_resnet110_SD	0.37518	0.20310	0.37346	0.18895	**0.17015**	0.18552
c100_resnet_wide32	0.42027	0.18573	0.33258	0.17951	**0.17082**	0.17966
SVHN_convnet	0.15935	0.03830	0.04276	0.02638	**0.02480**	0.02750
SVHN_resnet152_SD	0.01940	**0.01849**	0.02184	0.01988	0.02120	0.02088
SMRE	0.19932	0.10872	0.12603	0.08856	0.08636	0.08878

Table 7. Comparison of CCE for parametric recalibration methods across different classifiers and datasets. Table based on [6].

discrete, so it has measurable miscalibration. TCE and CCE will vary based on the scaling function and should be strongly correlated to the performance of this scaling function.

5.6. Recommendations for recalibrator choice

Which recalibration method is most sensible to use depends on circumstances and priorities.

Neural network classifiers. If one wants to recalibrate neural networks and is only interested in top-label calibration, temperature scaling is least complex, has best performance and is even accuracy preserving. If one is interested in calibration across all classes, the choice is between temperature scaling with lowest complexity and accuracy preservation and vector scaling with better performance but slightly higher complexity and potentially changing class predictions. In the case that miscalibration needs to be measurable one can incorporate either temperature scaling or vector scaling in the scaling-binning calibrator.

Other classifiers. If one wants to recalibrate other classifiers, either isotonic regression or dirichlet calibration with L2 regularization is most sensible. Isotonic regression has slightly better performance and measurable miscalibration but higher complexity. If measurable miscalibration is important, isotonic regression is arguably a better choice than incorporating dirichlet calibration with L2 regularization into the scaling-binning calibrator. If measurable miscalibration is not necessary Dirichlet calibration is arguably preferable.

In the case that accuracy preservation is very important or complexity should be avoided, one might also use temper-

ature scaling but it does not seem to improve a lot on the uncalibrated case (Table 4 and Table 5), so one might also choose to stick with uncalibrated results in that situation, or look into additional recalibration methods not discussed in this report.

6. Conclusion

In this report we explained the concept of calibration and outlined some recent developments regarding calibration in machine learning and deep learning. We described different parametric and non-parametric methods for calibration via post-processing and compared those methods on a variety of different metrics.

We conclude that calibration is an important consideration for probabilistic classifiers, that post-processing is an effective way to achieve calibration and that there is no strictly superior method for recalibration, but that the method must be chosen according to the requirements of the specific application.

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 7

[2] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983. 2

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 1, 2, 4, 5, 6, 9, 10

[4] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. *arXiv preprint arXiv:2006.12800*, 2020. 4

[5] Matthew Hutson. Artificial intelligence faces reproducibility crisis, 2018. 10

[6] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*, 2019. 6, 7, 10, 11

[7] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. *arXiv preprint arXiv:1909.10155*, 2019. 1, 2, 7, 8

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016. 4

[9] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 4

[10] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *arXiv preprint arXiv:2106.07998*, 2021. 4

[11] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 5, 10

[12] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. 4

[13] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003. 1

[14] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 4

[15] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 3, 6

[16] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467, 2020. 6

[17] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019. 4

[18] Christian Tomani, Daniel Cremers, and Florian Buettner. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. *arXiv preprint arXiv:2102.12182*, 2021. 4

[19] Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. *arXiv preprint arXiv:2010.09875*, 2020. 4

[20] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR, 2020. 4

[21] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001. 5

[22] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002. 5, 6

[23] Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning*, pages 11117–11128. PMLR, 2020. 4