ANALYSIS OF UNCERTAINTY INDUCING METHODS FOR NEURAL NETWORK CALIBRATION

SEMINAR REPORT - "BEYOND DEEP LEARNING: SELECTED TOPICS ON NOVEL CHALLENGES"

Thomas Wimmer
Department of Informatics
Technical University of Munich
wimmerth@in.tum.de

ABSTRACT

In recent years, the popularity of neural networks has increased and people have even allowed computers to perform such critical tasks as driving a car or using machine learning to detect cancer cells. Especially in these areas, it is of immense importance that the confidence of the neural network's output also matches the actual probability of an event occurring when it is predicted. In other words, it is important that the outputs of neural networks are calibrated. Starting from a fundamental work by Guo et al. (2017), more and more research has been done on post-hoc calibration methods in recent years. This report, in addition to introducing the topic of calibration, highlights three new methods that each attempt to address the issue of calibration in their own way (Gupta et al., 2020; Patel et al., 2020; Rahimi et al., 2020). Besides theoretical derivations and backgrounds of the methods that are presented, the three methods are further compared both qualitatively and quantitatively.

1 INTRODUCTION

Deep learning methods using neural networks (NNs) have revolutionised machine learning (ML) in recent years and are now state of the art in almost all areas of ML research. However, recent research suggests that while neural networks today are more accurate on the one hand, they are often not well calibrated on the other (Guo et al., 2017): The output confidence estimates do not match the actual probability of correctness of the NN's predictions. However, since reliable confidence estimates of the networks are of enormous importance, especially in safety-critical areas, there is a need for post-hoc calibration of the outputs.

Recent research has shown that previously proposed calibration methods, including histogram binning (HB), temperature scaling and other post-hoc calibration functions, have some weaknesses and may not work as well as initially thought. For this reason, several novel calibration methods have been developed, of which spline-based calibration (Gupta et al., 2020), I-max binning (Patel et al., 2020) and order-preserving functions (Rahimi et al., 2020) are reviewed in this paper.

1.1 DEFINITION OF CALIBRATION

Calibration is usually performed on supervised multi-class[1] classification tasks with samples (X, Y) where X is the input to the ML methods and Y is the corresponding output or label. We assume that X and Y are random variables that follow a ground truth joint distribution $\pi(X, Y) = \pi(Y \mid X)\pi(X)$, where $\pi(Y \mid X)$ is not necessarily 1 for a single class and 0 for the others. We further assume that all samples that are used for training, validation and testing are independent and identically distributed (iid with joint distribution π).

When considering the calibration of NNs, we assume that every model $f_\theta = \sigma \circ g_\theta$ with trainable parameters θ is a composition of a network g_θ with non-probabilistic output logits $Z \in \mathbb{R}^K$ (K

[1] Calibration can also be performed on single-class classification problems. However, in this paper we will focus on the more general case of multi-class classification.

classes) and a softmax function[2] σ that returns a probabilistic output P (i.e. class probabilities with $\sum_k P^{(k)} = 1$ where $P^{(k)}$ denotes the k-th element of the vector P). The class prediction $\hat{Y}$ (i.e. the top-1 class prediction) and the corresponding confidence score $\hat{P}$ can be derived from the network outputs by taking the maximum:

$$\hat{Y} = \arg\max_k P^{(k)} \quad \text{with the corresponding} \quad \hat{P} = \max_k P^{(k)}. \tag{1}$$

Note that X, Y, Z, P, $\hat{Y}$ and $\hat{P}$ are correlated random variables.

Using these definitions, perfect *multiclass-calibration* can be defined as

$$\mathbb{P}(Y = k \mid P = p) = p^{(k)}, \quad \forall p \in [0,1]^K \quad \forall k \in 1, ..., K. \tag{2}$$

Put into words, the probability that the class k is the true class Y, assuming that the confidence vector (probabilistic output of the model for all K classes) P is p, is exactly $p^{(k)}$ (the k-th element of the confidence vector).

A weaker notion of calibration is *classwise-calibration*:

$$\mathbb{P}\Big(Y = k \mid P^{(k)} = p^{(k)}\Big) = p^{(k)}, \quad \forall p \in [0,1]^K \quad \forall k \in 1, ..., K. \tag{3}$$

Put into words, the probability that the class k is the true class Y, assuming that the confidence for predicting this class $P^{(k)}$ is $p^{(k)}$, is exactly $p^{(k)}$.

Lastly, a classifier is *confidence-calibrated*[3], if

$$\mathbb{P}\Big(Y = \hat{Y} \mid \hat{P} = p\Big) = p, \quad \forall p \in [0,1]. \tag{4}$$

Put into words, the probability that the predicted class $\hat{Y}$ is the true class Y, assuming that the confidence for this class prediction $\hat{P}$ is p, is exactly p.

In practice, it is not possible to achieve perfect calibration but it is possible to improve it using post-hoc calibration methods.

1.2 Calibration measures

Finding a measure of the quality of the calibration of a method is an important task when comparing different calibration methods. There are a number of ways to compare two distributions, as well as some measures created specifically for quantifying calibration. The currently most popular evaluation metrics for calibration will be introduced in this section, while a new binning-free measure of calibration will be introduced and explained in section 3.1.

1.2.1 Expected Calibration Error (ECE)

In order to measure the (mis-)calibration of a method, one can inspect the expected difference between accuracy and confidence of a given method (Naeini et al., 2015)

$$\mathbb{E}\Big|\mathbb{P}\Big(\hat{Y} = Y \mid \hat{P} = p\Big) - p\Big|. \tag{5}$$

In practice, this term can be approximated by first partitioning the confidence scores on different samples (X_i, Y_i) with $i \in 1, ..., N$ into equally spaced bins B_m with $m \in 1, ..., M$. After that, an estimation can be computed by taking the weighted average of the differences between the average confidence and the accuracy in all the bins.

$$\begin{aligned} \text{acc}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{Y}_i = Y_i) \\ \text{conf}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \hat{P}_i \\ \text{confidence-ECE} &= \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \end{aligned} \tag{6}$$

[2] When we consider single-class classification, the softmax is replaced by a sigmoid function

[3] The differentiation of these different calibration definitions was first introduced in Kull et al. (2019)

The ECE is the most common calibration measure used in the literature on neural network calibration. However, several recent papers have noted that this metric has certain weaknesses (Nixon et al., 2019; Ashukha et al., 2020). Among other things, the emprisic approximation to the ECE asymptotically underestimates the true ECE as Vaicenavicius et al. (2019) has been able to show. Furthermore the ECE as proposed by Naeini et al. (2015) only takes into account the top-1 class prediction but not the calibration for the other classes (only confidence- instead of classwise-calibration). Therefore, Kull et al. (2019) proposed the classwise ECE

$$\text{classwise-ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \frac{1}{K} \sum_{k=1}^{K} \left|\text{acc}_k(B_m) - \text{conf}_k(B_m)\right|, \tag{7}$$

with acc_k and conf_k being the classwise average prediction and actual proportion of the class k in the bin B_m. One can think of this measure as an averaged ECE over all classes.

1.2.2 Maximum Calibration Error (MCE)

The MCE is base on the same idea of binning as the ECE and was also proposed by Naeini et al. (2015). Formally it is defined as

$$\max_{p\in[0,1]} \left|\mathbb{P}\Big(\hat{Y} - Y|\hat{P} = p\Big) - p\right| \tag{8}$$

and can empirically be approximated with

$$\text{MCE} = \max_{m\in\{1,\dots,M\}} \left|\text{acc}(B_m) - conf(B_m)\right|. \tag{9}$$

1.2.3 Brier Score

Early work on the calibration of predictions involved the analysis of weather forecasts, and meteorologists proposed an assessment measure in Brier et al. (1950) that we now call the Brier score:

$$\text{Brier} = \frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - \mathbb{1}(Y_i = \hat{Y}_i))^2 \tag{10}$$

or in a more general manner defined for all classes $k \in \{1, ..., K\}$

$$\text{Brier} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(P_i^{(k)} - \mathbb{1}(Y_i = k))^2. \tag{11}$$

As one can see, the Brier score is also known as mean squared error (MSE) nowadays.

1.2.4 Negative Log Likelihood (NLL)

Since we are comparing the predicted probability distribution and the actual class-distribution in the test samples, we can of course also use the well-known NLL to evaluate the calibration of our methods.

2 Related Work

Guo et al. (2017) first published possible weaknesses in the calibration of deep NNs and reviewed a number of post-hoc calibration methods that can be used to adjust the scores to represent their true confidence. While first examining the performance of single-class calibration methods, including histogram binning (Zadrozny & Elkan, 2001), isotonic regression (Zadrozny & Elkan, 2002), Bayesian binning into quantiles (BBQ) (Naeini et al., 2015) and Platt scaling (Platt et al., 1999), Guo et al. (2017) further presents a number of multi-class calibration methods: First, the paper explains the simple extension of binary binning methods to the multiclass domain by treating the calibration problem as K one-versus-all binary calibrations. This technique however has some substantial weaknesses that will be explained and elaborated on in section 3.2 and 5. Subsequently, various generalisations of Platt scaling to multiclass calibration are presented, whereby the simplest

solution, the so-called temperature scaling, results in the best calibration of the network outputs when empirically measuring the $\mathrm{confidence\text{-}ECE}$(Guo et al., 2017).

Following this pioneering work, more research has been conducted on the calibration of neural networks. Among these more recent works, Kull et al. (2019) introduced a post-hoc scaling method derived from Dirichlet-distributed likelihoods (as opposed to Platt scaling which is derived from Gaussian likelihoods). Further, Kumar et al. (2019) combined scaling and binning methods in order to achieve verifiable calibration (which is not guaranteed in scaling-only methods), while Zhang et al. (2020) suggests using an ensemble of parametric calibration methods (e.g. temperature scaling) and combining it using composition to achieve accuracy-preserving, data-efficient and expressive calibration.

Although this report is focused on post-hoc calibration methods, it is important to mention that there are several ideas on how to obtain calibrated network outputs by using specific network architectures as Bayesian DNNs (Blundell et al., 2015) and approximations thereof (Gal & Ghahramani, 2015), certain modifications to the loss function that is used during training (Kumar et al., 2018) or ensemble learning (Lakshminarayanan et al., 2017). Section 5 contains a brief comparison between these methods and post-hoc calibration.

3 METHODS

The following section will summarize and explain three recent works on the post-hoc calibration of neural networks, namely

- *Calibration of Neural Networks using Splines* (Gupta et al., 2020),
- *Multi-Class Uncertainity Calibration via Mutual Information Maximization-based Binning* (Patel et al., 2020) and
- *Intra Order-Preserving Functions for Calibration of Multi-Class Neural Networks* (Rahimi et al., 2020).

3.1 CALIBRATION OF NEURAL NETWORKS USING SPLINES

Gupta et al. (2020) propose a novel continuous (binning-free) measure for the quantification of the calibration error that is based on the KS-Error. Using the ideas of the empirical approximation of this error, they propose a novel calibration method by fitting a differentiable function to the empirically found data points from the calibration set and using its derivative as recalibration function.

3.1.1 KOLMOGOROV-SMIRNOV CALIBRATION ERROR

First, the authors introduce a new measure for the calibration of probabilistic predictions that, in contrast to the ECE and its variations is a binning-free metric resulting in a parameter-free measure (opposed to the explicit setting of bin size and boundaries for ECE). The proposed metric is inspired by the Kolmogorov-Smirnov (KS) test (Kolmogorov, 1933; Smirnov, 1939).

The authors rewrite the definition of classwise calibration (Eq. 3) using Bayes' rule:

$$\mathbb{P}\Big(Y = k, P^{(k)} = p^{(k)}\Big) = p^{(k)}\mathbb{P}\Big(P^{(k)} = p^{(k)}\Big), \tag{12}$$

which we will write as $\mathbb{P}\big(k, p^{(k)}\big) = p^{(k)}\mathbb{P}\big(p^{(k)}\big)$ in the following in order to stick to a simple and readable notation.

The KS test works by comparing cumulative distributions, resulting in the following property that we want to test for a given k:

$$\int_0^\sigma \mathbb{P}\Big(k, p^{(k)}\Big)dp^{(k)} = \int_0^\sigma p^{(k)}\mathbb{P}\Big(p^{(k)}\Big)dp^{(k)} \tag{13}$$

Writing the two sides of the equation as $\psi_1(\sigma), \psi_2(\sigma)$ respectively, the KS-distance is defined as

$$\mathrm{KS} = \max_\sigma \left|\psi_1(\sigma) - \psi_2(\sigma)\right|. \tag{14}$$

Gupta et al. (2020) argue that $p^{(k)}$ usually consistently over- or underestimates $\mathbb{P}\big(k|p^{(k)}\big)$, and therefore the sign of $\mathbb{P}\big(k|p^{(k)}\big) - p^{(k)}$ is constant for all $p^{(k)}$ (as well as the sign of $\mathbb{P}\big(k, p^{(k)}\big) - p^{(k)}\mathbb{P}\big(p^{(k)}\big)$) which results in a maximum value of the KS-distance when σ is set to 1. Using this, the *expected* difference between $p^{(k)}$ and $\mathbb{P}\big(k|p^{(k)}\big)$ is defined as

$$\mathrm{KS} = \int_0^1 \left|\mathbb{P}\Big(k, p^{(k)}\Big) - p^{(k)}\mathbb{P}\Big(p^{(k)}\Big)\right| dp^{(k)} = \int_0^1 \left|\mathbb{P}\Big(k|p^{(k)}\Big) - p^{(k)}\right| \mathbb{P}\Big(p^{(k)}\Big) dp^{(k)}, \tag{15}$$

which can also be referred to as the expected calibration error for the class k.

In practice, the cumulative distributions in Eq. 13 can be approximated as follows:

$$\begin{gathered}\psi_1(\sigma) = \int_0^\sigma \mathbb{P}\Big(k, p^{(k)}\Big) dp^{(k)} \approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(P_i^{(k)} \leq \sigma) \cdot \mathbb{1}(Y_i = k) \\ \text{and} \\ \psi_2(\sigma) = \int_0^\sigma p^{(k)}\mathbb{P}\Big(p^{(k)}\Big) dp^{(k)} \approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(P_i^{(k)} \leq \sigma) P_i^{(k)}.\end{gathered} \tag{16}$$

Sorting the data according to the values $P_i^{(k)}$, we can define two sequences h and $\tilde{h}$ with

$$\begin{aligned}\tilde{h}_0 = h_0 &= 0, \\ \tilde{h}_i &= \tilde{h}_{i-1} + \mathbb{1}(Y_i = k)/N, \\ h_i &= h_{i-1} + P_i^{(k)}/N\end{aligned} \tag{17}$$

and compute the empirical estimation of the KS-calibration error using these sequences as

$$\mathrm{KS}(P^{(k)}) = \max_i \left|h_i - \tilde{h}_i\right|. \tag{18}$$

3.1.2 Recalibration using Splines

While the h_i from Eq. 17 is an empirical estimation for

$$h_i \approx \mathbb{P}\Big(Y = k, P^{(k)} \leq P_i^{(k)}\Big), \tag{19}$$

we can define a continuous function $h(t)$ that is also empirically approximated by h_i (by setting $t = i/N$):

$$\begin{aligned}h(t) &= \mathbb{P}\Big(Y = k, P^{(k)} \leq s(t)\Big) \\ h'(t) &= \mathbb{P}\Big(Y = k \mid P^{(k)} = s(t)\Big),\end{aligned} \tag{20}$$

where $s(t)$ is the t-th fractile score and $h'(t) = dh/dt$ the derivative of the function.

The idea behind the method proposed in Gupta et al. (2020) is to fit a continuous, differentiable function to the sampled sequence h_i (that is sampled using a held out calibration set, e.g. the validation set). According to Eq. 20, the derivative of this fitted function is also an estimate for the conditional probability $\mathbb{P}\big(Y = k \mid P^{(k)} = s(t)\big)$ and can therefore be used to recalibrate the predicted probabilities that are the output of the NN.

Gupta et al. (2020) propose to fit a cubic spline to the sampled points h_i (McKinley & Levine, 1998). Moreover, least-squares spline fitting is used to fit a function to a small number of knot points $(u_i, v_i) = (i/N, h_i)$ (i.e. the sampled h_i plotted against the fractile score). Since the fitted cubic spline is twice continuously differentiable, the derivative of the fitted function can serve as an estimate for $\mathbb{P}\big(Y = k \mid P^{(k)} = s(t)\big)$, which can be used directly to recalibrate the output of the network: The output values of the network $P^{(k)}$ can be mapped to $\mathbb{P}\big(Y = k \mid P^{(k)} = s(t)\big)$ by computing the fractile of the output score and using this value $s^{-1}(t)$ as input for the derivative of the fitted spline $h'(t)$:

$$\gamma(\sigma) = h'(s^{-1}(\sigma)) \tag{21}$$

with the recalibration function γ, the output of the network σ, the analytically known (through differentiating the fitted spline) derivative of the function h and the function s^{-1} which maps an output score σ to its fractile.

3.2 Multi-Class Uncertainity Calibration via Mutual Information Maximization-based Binning

The second work analyzed in this report paper uses a substantially different idea than the paper presented in section 3.1. Patel et al. (2020) identifies the problem that the use of equal size or equal mass binning in common binning methods (Zadrozny & Elkan, 2001; 2002; Naeini et al., 2015) for recalibration is sample-inefficient for multi-class calibration and often leads to a decrease in classification accuracy of the models. The proposed solutions to these problems will be introduced in the following sections.

3.2.1 Sample Efficiency for Multi-Class Calibration

The general training of binning methods work by splitting the predicted logits[4] Z_i of the network for a number of samples (X_i, Y_i) in M bins (B_m with bin intervals $[g_{m-1}, g_m)$) and assigning each of these bins a representative R_m which is the logit corresponding to the actual share of positive samples in the bin:

$$\begin{aligned} R_m &\triangleq \log(q) - \log(1-q) \\ \text{with} \qquad q &= \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(Y_i = \hat{Y}_i) \end{aligned} \tag{22}$$

As indicated in Section 2, binning methods were originally designed for binary decisions and the most common attempt to extend these methods to the multiclass case is to treat the calibration problem as K independent one-vs-rest problems. This method however results in a severe sample inefficiency for classes k with small prior p_k. Since for these classes, large calibration sets are needed to have enough class-k samples for setting accurate bin representatives.

Patel et al. (2020) propose a novel shared class-wise binning in which similar classes are merged into the same calibration set, resulting in more positive samples. This shared calibration scheme over a number of classes can be extended to optimizing a single binning scheme for all classes in case the class priors are balanced. This method can of course also be used to optimize the sample efficiency of already existing calibration methods that use the one-vs-rest strategy to extend from binary to multi-class calibration (e.g. the well-known Histogram binning, BBQ or one-vs-rest extension of Platt scaling).

3.2.2 Bin Optimization via Mutual Information (MI) Maximization

Eq. mass binning struggles especially with small class priors and can result in bins that barely cover areas with high uncertainty in the predicted class (where calibration would be the most important). Eq. size binning on the other hand can results in bin edges that are too narrow and therefore in higher uncertainty for the bin representatives R_m. This is visualized in Figure 1.

In order to fix these problems in binning methods, Patel et al. (2020) proposes bin optimization with the goal to maximize the mutual information I between the quantized logits $Q(z)$ and the true label Y, where the binning calibrator is seen as quantizer Q:

$$\{g_m^*\} = \arg\max_{Q:\{g_m\}} I(Y; m = Q(Z)) = \arg\max_{Q:\{g_m\}} H(m) - H(m \mid Y), \tag{23}$$

where m is seen as a discrete random variable with $\mathbb{P}(m|Y) = \int_{g_{m-1}}^{g_m} \mathbb{P}(z|Y)dz$ and $\mathbb{P}(m) = \int_{g_{m-1}}^{g_m} \mathbb{P}(z)dz$ and the entropy $H(m)$ and the conditional entropy $H(m|Y)$ are used in the second equality.

When analyzing this equation, it is interesting to see that $H(m)$ is maximal if $\mathbb{P}(m)$ is uniform (corresponding to Eq. mass binning) and an additional part $H(m|Y)$ which considers the available label information Y to result in maximally preserved label information for accuracy.

[4]The corresponding predicted scores P_i could be used as well for binning, but finding the right binning edges is usually easier in the logit space $\mathbb{R}$ instead of in the interval $[0, 1]$.

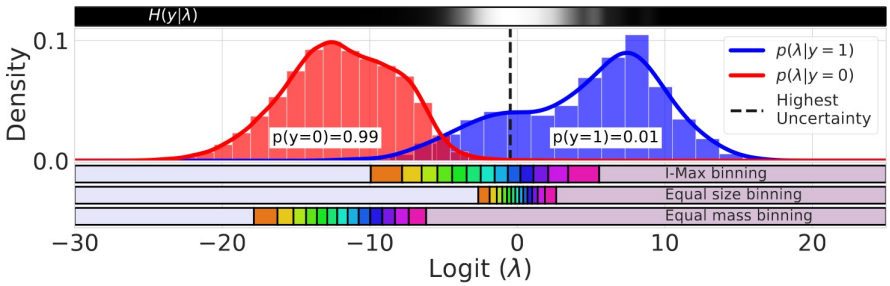

Figure 1: Histogram and KDE of CIFAR-100 logits constructed from 1k calibration samples. Due to the imbalanced class ratio, the bin edges of Eq. mass binning mainly cover class-0 resulting in an inaccurate calibration method. Eq. size and I-Max binning both cover the region with high uncertainty but the dense binning scheme of Eq. size binning will likely not have enough samples per bin to compute accurate bin representatives. The proposed I-Max binning results in a dense binning in the critical area with high uncertainty but still enough bin width to compute accurate representatives and cover some of the areas with lower uncertainty. Figure taken from Patel et al. (2020).

3.3 Intra Order-Preserving Functions for Calibration of Multi-Class Neural Networks

The third work that is analyzed in this report is again based on a different idea. Namely, Rahimi et al. (2020) argue that commonly used calibration techniques are often too simple and hence lack the expressiveness to calibrate complicated function landscapes generated by deep neural networks. In order to keep the accuracy of the network's output, they propose the use of a novel family of neural networks, namely intra order-preserving neural networks. Using a neural network for calibration of predictions is not a new idea and has been used in several previous works. However, Rahimi et al. (2020) argue that the use of unconstrained neural networks might harm the accuracy of the top-1 (or top-k) predictions and that unconstrained neural networks are more likely to overfit on a small calibration set. They therefore propose to use a class of functions named intra-order preserving functions instead which is going to presented in the next section.

3.3.1 Intra Order-Preserving Functions

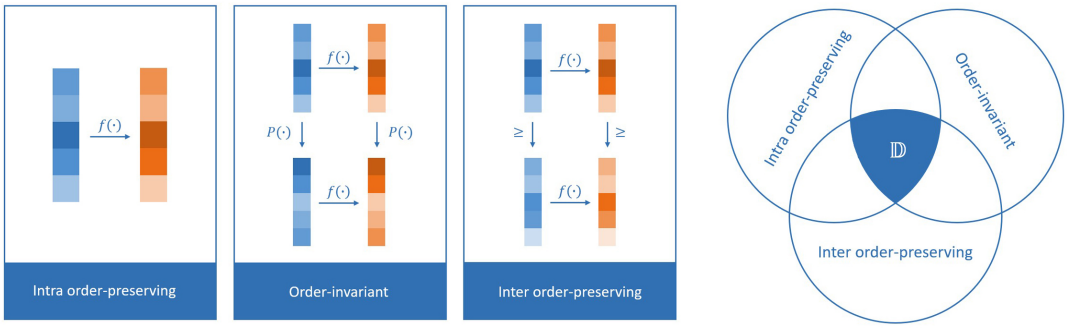

Figure 2: Visual representation of the function families that are interesting for use as recalibration functions. For intra order-preserving functions, the ranking within the resulting vector remains the same as in the input. Order-invariant functions are invariant to permutations of the input, and inter order-preserving functions maintain the element-wise relationships between two vectors. The relationship between these three families of functions is shown on the right. The family of diagonal order-preserving functions $\mathbb{D}$ lies in the intersection of the three families.

Rahimi et al. (2020) first give an overview on the desirable class of functions for calibration[5].

Definition 3.1. A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is intra order-preserving if, for any $x \in \mathbb{R}^n$, both x and $f(x)$ share the same ranking. Meaning that for any tie braker t, x and $f(x)$ have the same sorting matrices $S(x) = S(f(x))$.

It intuitively makes sense that using intra order-preserving functions for calibration keeps the top-k accuracy the same since the output of these functions has the same order as the original predictions. Common intra order-preserving functions are the softmax operator and temperature scaling.

Rahimi et al. (2020) further prove that all intra order-preserving functions can be described in a certain way, namely:

Theorem 3.1. A continuous function $f : \mathbb{R}^n \to \mathbb{R}^n$ is intra order-preserving iff $f(x) = S(x)^{-1}Uw(x)$ with U being an upper-triangular matrix of ones, $S(x)$ as sorting matrix of the input x and $w : \mathbb{R}^n \to \mathbb{R}^n$ being a continuous function such that

- $w_i(x) = 0$, if $y_i = y_{i+1}$ and $i < n$,
- $w_i(x) > 0$, if $y_i > y_{i+1}$ and $i < n$,
- $w_n(x)$ is arbitrary,

where $y = S(x)x$ is the sorted input x.

Note that the matrix U results in a reverse cumulative sum of any vector v when applied on it (i.e. $(Uv)_i = \sum_{j=i}^{n} v_j$). Furthermore, since $w(x) \geq 0$, applying this upper-triangular matrix results in a sorted vector.

In many cases, different classes share some common characteristics and therefore it might be interesting to share some properties of the calibrator across all classes. This is where the concept of order-invariant functions becomes useful.

Definition 3.2. A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is order-invariant if, for any permutation matrix $P \in \mathbb{P}^n$ and all $x \in \mathbb{R}^n$, $f(Px) = Pf(x)$.

By using order-invariant functions, swapping the elements x_i and x_j results in simply swapping the corresponding $f_i(x)$ and $f_j(x)$. In the context of calibration, this means that the mapping learned for the i-th class can also be used for class j. Therefore, the calibration function is shared between different functions while still allowing the output of each class to depend on all other class predictions.

One can again derive a representation similar to the one in Theorem 3.1 for functions that are intra order-preserving **and** order-invariant.

Theorem 3.2. A continuous, intra order-preserving function $f : \mathbb{R}^n \to \mathbb{R}^n$ is order-invariant iff $f(x) = S(x)^{-1}Uw(\boldsymbol{y})$ with x, y, S(x), U and w(x) as defined in Theorem 3.1.

Note that the only difference to functions that are only intra order-preserving is, that we use the sorted input $y = S(x)x$ instead of x.

Finally, one can share even more properties of the calibrator across different classes by using diagonal functions.

Definition 3.3. A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is diagonal, if $f(x) = [f_1(x_1), f_2(x_2), ..., f_n(x_n)]$ for $f_i : \mathbb{R} \to \mathbb{R}$ with $i \in [n]$.

Using diagonal functions for calibration means that the predictions for different classes do not interact with each other in f.

Theorem 3.3. A continuous, intra order-preserving function $f : \mathbb{R}^n \to \mathbb{R}^n$ is diagonal iff $f(x) = [\bar{f}(x_1), \bar{f}(x_2), ..., \bar{f}(x_n)]$ with some continuous and *increasing* function $\bar{f} : \mathbb{R} \to \mathbb{R}$.

[5]In this report paper, we will not cover the proofs of any of the theorems presented in Rahimi et al. (2020). For detailed explanations and proofs, please refer to the original paper.

Diagonal, intra order-preserving functions are also order-invariant and inter order-preserving and the use of functions of this family is motivated by the good results shown from temperature scaling which is also a diagonal, intra order-preserving method.

Definition 3.4. A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is inter order-preserving, if for any $x, y \in \mathbb{R}^n$ with $x \geq y$, $f(x) \geq f(y)$.

In the context of calibration, inter order-preserving functions guarantee that $f_i(x)$ increases with the original class logit x_i (Rahimi et al., 2020).

An overview on the different families presented in this section and their relation can be found in Figure 2.

3.3.2 Intra Order-Preserving Neural Networks

Rahimi et al. (2020) further propose an approach on how to implement the presented families of functions as trainable neural networks in practice. As there already is a decomposition of intra order-preserving (and order-invariant) functions, the implementation in a neural network is pretty straight forward with the only remaining challenge on how to design w in such a way that it fulfills all rules given in Theorem 3.1.

The authors propose to use a parameterization trick by setting $w_i(x) = \sigma(y_i - y_{i+1})m_i(x)$ with a strictly positive $m_i(x)$ that consists of a multi-layer trainable neural network and $\sigma : \mathbb{R} \to \mathbb{R}$ being a positive function that is only $\sigma(a) = 0$ in case $a = 0$. In practice, one simply uses the absolute value function as $\sigma(x) = |x|$. Lastly, in order to ensure that $m(x)$ is strictly positive, the softplus activation is used in the final layer of m [6].

Using these modifications, one can implement intra order-preserving (and optionally order-invariant) neural networks by using an architecture as the one shown in Figure 3. For the learning of diagonal, intra order-preserving functions, the authors suggest utilizing the work of Wehenkel & Louppe (2019), who proposed neural networks that learn increasing functions.

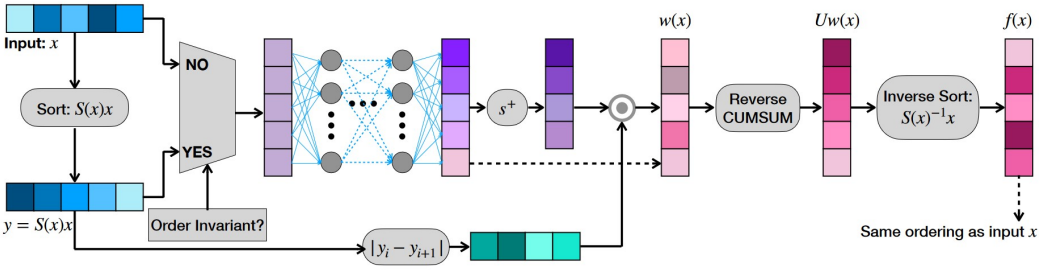

Figure 3: Flow graph of intra order-preserving neural networks. Taken from Rahimi et al. (2020).

[6] The softplus activation is defined as $s^+(a) = \log(1 + \exp(a))$.

4 EXPERIMENTS

After presenting the three calibration methods, the following section gives an overview of the experiments performed in the papers and aims to present the results as comparably as possible. However, since a quantitative comparison of reported results can be error-prune and not really fair, section 5 will focus on a qualitative analysis and comparison of the presented methods.

4.1 SETUP OF REPORTED EXPERIMENTS

The authors used pretrained models of well-known neural network architectures for classification as the Resnet (He et al., 2015), Wide Resnet (Zagoruyko & Komodakis, 2016) or the DenseNet (Huang et al., 2017) and evaluated the calibration performance of the proposed methods on commonly used datasets such as ImageNet (Deng et al., 2009), Cifar-10(0) (Krizhevsky et al., 2009) or SVHN (Netzer et al., 2018). Unfortunately, comparing the three papers in a quantitative way is a hard without performing any experiments and only relying on the reported numbers. The following subsection shows the collected results and is restricted to only the top-1 ECE since this is the only calibration measure that all three papers reported.

It is important to note that the quantitative comparison shown here is **not representative**. To make a comparison in a scientific manner, one would have to conduct new experiments with the three methods oneself. In particular, one would have to be careful to use the same pre-trained models, use the same subsets of the data sets as calibration sets, and run a test with multiple metrics (including KS error). The results listed in Table 1 are only intended to give an impression of the performance of the methods presented.

4.2 Results of reported Experiments

Table 1: Reported top-1 ECE scores. To make the comparison of the reported results somewhat possible, the corresponding baseline scores are given. Note that these scores and the general experimental setup are the same for Rahimi et al. (2020) and Gupta et al. (2020) but is different for Patel et al. (2020). The best calibration results as well as the biggest improvements over the corresponding baseline scores are marked in bold. An in-depth analysis on the performance of the presented calibration methods can be found in section 5. For Rahimi et al. (2020), the results of diagonal intra order-preserving functions (DIAG), order-invariant and intra order-preserving (OI) and simple intra order-preserving (OP) are given. For Patel et al. (2020), in addition to the simple I-Max binning calibration, the results of a combination of I-Max and GP (state-of-the-art non-parametric scaling method, Wenger et al. (2020)) is reported.

		Intra Order-Preserving Functions							Splines			MI Maximization-based Binning				
		Baseline	DIAG		OI		OP		Baseline	Splines		Baseline	I-Max		I-Max + GP	
CIFAR10	Resnet[1]	0,0475	0,0067	14%	0,0061	***13%***	0,0119	25%	0,0475	0,0101	21%	0,0201	0,0104	52%	**0,0052**	26%
	Wide Resnet-32	0,0451	0,0136	30%	**0,0064**	***14%***	0,0077	17%	0,0451	0,0100	22%	0,0288	0,0113	39%	0,0074	26%
	DenseNet-40[2]	0,055	**0,0069**	***13%***	0,0116	21%	0,0128	23%	0,055	0,0139	25%	0,0253	0,0098	39%	**0,0048**	19%
CIFAR100	Resnet[1]	0,1848	0,0507	27%	**0,0119**	***6%***	0,0253	14%	0,1848	0,0187	10%	0,059	0,0205	35%	0,0121	21%
	Wide Resnet-32	0,1878	0,0172	9%	**0,0126**	***7%***	0,0173	9%	0,1878	0,0167	9%	0,0748	0,0231	31%	0,0179	24%
	DenseNet-40[2]	0,2116	**0,0075**	***4%***	0,0098	5%	0,0154	7%	0,2116	0,0211	10%	0,0762	0,0189	25%	0,0114	15%
ImageNet	DenseNet-161	0,0572	0,0103	18%	0,0123	22%	0,0168	29%	0,0572	**0,0080**	***14%***	0,0571	0,0201	35%	0,0204	36%
	Resnet-152	0,0654	0,0087	13%	0,0109	17%	0,0167	26%	0,0654	**0,0091**	***14%***	0,0512	0,0196	38%	0,0144	28%
SVHN[3]	Resnet-152	0,0087	**0,0057**	66%	0,0116	133%	0,0118	136%	0,0087	0,0083	95%	0,0201	0,0164	82%	0,0074	***37%***
Mean improvement				22%		26%		32%			25%			42%		26%

[1]Resnet-110 (He et al., 2015) is used in Rahimi et al. (2020); Gupta et al. (2020), while Patel et al. (2020) uses a ResNext8x64 (Xie et al., 2016).
[2]Patel et al. (2020) uses a DenseNet-BC(L=190, k=40).
[3]Rahimi et al. (2020); Gupta et al. (2020) report the top-1 ECE evaluated with 25 bins and use a Resnet with stochastic depth (Huang et al., 2016), while Patel et al. (2020) reports the top-1 ECE on 15 bins (as also used in all other experiments). Patel et al. (2020) also uses only 1k instead of 13k available calibration samples to demonstrate the sample-efficiency of the proposed calibration method.

5 DISCUSSION

The following section contains some of my thoughts on the strengths and possible weaknesses of the works presented, as well as a brief quantitative comparison between the reported results from Table 1. Since the three methods are based on fundamentally different ideas, it is difficult to compare their performance, even if one performed some experiments on its own.

5.1 QUALITATIVE COMPARISON

5.1.1 CALIBRATION USING SPLINES

Table 2: Overview of the qualitative analysis of the proposed method in Gupta et al. (2020)

	Calibration of NNs using Splines (Gupta et al., 2020)
Accuracy	One-vs-rest approach, modification provably keeps top-r accuracy
Hyperaparameters	Only number of knots for spline fitting
Runtime	Learning-free
Multiclass calibration	One-vs-rest approach
Data efficiency	One-vs-rest might cause problems with rare classes
Overfitting risk	Low
$\sum_i P^{(i)} = 1$	No

Gupta et al. (2020) introduced a new binning-free calibration metric derived from the KS statistical test. They also propose a new calibration method that uses the interpolation of an empirically found sequence to derive a continuous calibration function for new samples. The method is learning-free and has no hyperparameters[7] so that it is easy to apply. The risk of overfitting is low with this method. However, the KS test in its basic definition (used in the paper) is intended for comparing univariate distributions, and therefore the calibration method is also primarily designed for a binary calibration task and needs to be extended with a one-vs-rest approach to be used in the multiclass environment. Therefore, there is no guarantee that the method will maintain the accuracy of the predictions when the calibration method is applied independently to the different classes. However, with a small modification of the method, one can demonstrably maintain the top-r accuracy of the prediction, and no loss of accuracy has been empirically observed in the conducted experiments. The extension of the method using the one-vs-rest approach may also result in poorer performance when the prior of a class is low (i.e. for rare classes). Finally, the method does not guarantee by default that the sum of the calibrated probabilities is one. In cases where this behaviour is required, one would need to use some post-processing.

5.1.2 CALIBRATION VIA MI MAXIMIZATION-BASED BINNING

Table 3: Overview of the qualitative analysis of the proposed method in Patel et al. (2020)

	Calibration via MI Maximization-based Binning (Patel et al., 2020)
Accuracy	One-vs-rest approach, Can influence accuracy
Hyperaparameters	No hyperparameters
Runtime	Optimization of bin edges
Multiclass calibration	One-vs-rest approach but contains some multi-class concepts
Data efficiency	Concepts to deal with little available data
Overfitting risk	Low
$\sum_i P^{(i)} = 1$	No

Patel et al. (2020) proposed a new binning-based method for calibration that uses the available information not only to compute the bin representatives but also to optimize the bin edges. They incorporate this information in the objective function that tries to maximize the mutual information

[7] No hyperparameters except the number of knots for spline-fitting, but the authors provide a heuristic for this value.

between the actual class labels and the calibrated values. The bins produced in this way are especially more suitable for calibration of predictions for classes with a small prior. The second idea introduced in the paper can be applied to any method that uses the one-vs-rest approach to extend to multi-class calibration. The idea proposed is to merge the calibration sets for classes with a similar prior distribution to achieve more accurate bin representatives (or other parameters of the calibration methods) for rare classes.

The proposed method has no hyperparameters to be tuned empirically and incorporates two ideas that improve calibration performance for rare classes in unbalanced environments. However, the problem with all binning-based methods is that the output produced is discrete and reducing the size of the bins (i.e. more bins) to achieve a finer-grained output usually results in poorer bin representatives because there are too few samples in each bin. Combined with the "one-vs-rest" approach used to extend binning methods to the multi-class setting, the accuracy of the original predictions will most likely still be compromised, although the proposed MI maximisation-based binning is the binning method that suffers least from this problem. Furthermore, the sum of the calibrated results is not guaranteed to be equal to 1, and thus the calibrated values might need to be post-processed.

5.1.3 Intra Order-Preserving Functions for Multi-Class Calibration

Table 4: Overview of the qualitative analysis of the proposed method in Rahimi et al. (2020)

	Intra Order-Preserving Functions for Calibration (Rahimi et al., 2020)
Accuracy	Keeps Accuracy
Hyperaparameters	NN introduces a good amount of hyperparameters
Runtime	Training of NN
Multiclass calibration	Explicitly handles multi-class calibration
Data efficiency	Small risk of overfitting the NN
Overfitting risk	NN can overfit on calibration set
$\sum_i P^{(i)} = 1$	Yes

Rahimi et al. (2020) proposed to use a special kind of neural networks that can be trained to calibrate the outputs of a machine learning method. The method that is explicitly designed to handle multi-class calibration makes use of so called intra order-preserving methods. By constraining the functions that are learned by the NN in this way, the method can provably preserve the accuracy of the original prediction while returning calibrated confidences. On the other hand, the use of a neural network introduces a good amount of hyperparameters that need to be empirically found (e.g. structure of neural network, training procedure). The training and optimization of the hyperparameters of the neural network might therefore take some time. Using intra order-preserving (and possibly order-invariant or diagonal) functions prevents the neural network from overfitting to the calibration set and altering the accuracy of the method. However, the NN can still overfit the "calibrated" outputs to the calibration set which might cause problems with little available data. Another advantage of the proposed method is the ability to output calibrated probabilities that sum to one, which can be useful in some cases (e.g. when using the predicted probabilities for a sequence of elements for further decoding in the pipeline).

5.2 Quantitative comparison

As already mentioned in Section 4, the comparison of the reported results is only of limited significance. Especially because the reported results are only given for the top-1 ECE for all of the three methods.

However, it is still interesting to observe that the intra order-preserving functions in general performed very well and were able to decrease the calibration error by in average 73% (Rahimi et al., 2020). It is also interesting to see that the order-invariant, intra order-preserving functions and the diagonal, intra order-preserving functions performed better than the functions that are only intra order-preserving.

The calibration method proposed by Gupta et al. (2020) leads to comparable improvements in calibration. It is no wonder that the method outperforms all methods that it gets compared to in the

paper in terms of the newly proposed KS calibration error. However, even when considering the ECE metric, the method performs very good and achieves a new state-of-the-art in ImageNet calibration (Gupta et al., 2020).

Finally, the mutual information maximization-based binning proposed by Patel et al. (2020) performs a worse than the other two methods (but performance is also tested on different models and possibly different calibration sets), but the calibration can be improved when combining the binning-based method with a non-parametric scaling method (Wenger et al., 2020). However, it still achieves the biggest improvement compared to the baseline method on the SVHN dataset (Netzer et al., 2018) which has an imbalanced calibration set. From this you can see that the methods for improving performance in the face of class inequality and low data availability actually work.

6 Conclusion

To sum up, the main contributions of Gupta et al. (2020) are a new binning-free calibration metric, the KS calibration error, and a theoretically justified learning-free calibration method using interpolation. Patel et al. (2020) proposed a binning-based method with an optimized choice of bin edges through incorporation of available label information. They further proposed a concept to increase sample efficiency when extending methods for multi-class calibration using a one-vs-rest approach, by merging the calibration sets of classes with a similar prior. Finally, Rahimi et al. (2020) introduced a new family of neural networks, intra order-preserving functions (and its variants), that could also be used in other domains where an intra order-preserving function needs to be optimized.

Calibrated outputs from neural networks are of immense importance in times when some decisions are based entirely on their predictions, especially in safety-critical areas. The methods that were presented in this report paper are some of the most advanced post-hoc calibration methods at the moment and can be easily used in every ML prediction task as they can just be appended to an already trained model by using the validation set as calibration set. There is no one right method of calibration that fits all purposes. Choosing the calibration method can be seen as a part of the design process in some way. This report is intended to provide the reader with an overview and basic understanding of the methods presented and can serve as a decision-making aid for the choice of calibration method.

REFERENCES

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yarin Gal and Z Ghahramani. Dropout as a bayesian approximation: representing model uncertainty in deep learning. arxiv. *arXiv preprint arxiv:1506.02142*, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Andrey Kolmogorov. Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, 4:83–91, 1933.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. *arXiv preprint arXiv:1909.10155*, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1): 1049–1060, 1998.

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

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and A Ng. The street view house numbers (svhn) dataset. Technical report, Technical report, Accessed 2016-08-01.[Online], 2018.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019.

Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. *arXiv preprint arXiv:2006.13092*, 2020.

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.

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.

Nikolai V Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2):3–14, 1939.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3459–3467. PMLR, 2019.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. *Advances in Neural Information Processing Systems*, 32:1545–1555, 2019.

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

Saining Xie, Ross B Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. corr abs/1611.05431 (2016). *arXiv preprint arXiv:1611.05431*, 2016.

Bianca Zadrozny and Charles Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 204–213, 2001.

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*, pp. 694–699, 2002.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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*, pp. 11117–11128. PMLR, 2020.