SHOULD ENSEMBLE MEMBERS BE CALIBRATED?

Anonymous authors
Paper under double-blind review

ABSTRACT

Underlying the use of statistical approaches for a wide range of applications is the assumption that the probabilities obtained from a statistical model are representative of the "true" probability that event, or outcome, will occur. Unfortunately, for modern deep neural networks this is not the case, they are often observed to be poorly calibrated. Additionally, these deep learning approaches make use of large numbers of model parameters, motivating the use of Bayesian, or ensemble approximation, approaches to handle issues with parameter estimation. This paper explores the application of calibration schemes to deep ensembles from both a theoretical perspective and empirically on a standard image classification task, CIFAR-100. The underlying theoretical requirements for calibration, and associated calibration criteria, are first described. It is shown that well calibrated ensemble members will not necessarily yield a well calibrated ensemble prediction, and if the ensemble prediction is well calibrated its performance cannot exceed that of the average performance of the calibrated ensemble members. On CIFAR-100 the impact of calibration for ensemble prediction, and associated calibration is evaluated. Additionally the situation where multiple different topologies are combined together is discussed.

1 INTRODUCTION

Modern deep neural networks have achieved significant accuracy improvement compared with model decades ago. However, deep neural networks tend to be overconfident and generate ill-calibrated predictions. A system is well calibrated, if when the system generate a probability of 0.4, then 40% of the time the system is correct in the associated predictions. Calibration is very important in deploying system, especially in cost-sensitive tasks, such as medicine[], auto-driving[], and economics[], etc. Decades ago, it is verified by Niculescu-Mizil & Caruana (2005) that shallow neural networks are well calibrated. Recently, Guo et al. (2017) found the ill-calibration phenomenon of recent neural network model with deep structures, which inspire lots of research work to address the calibration problem of deep neural network-based systems.

Previous methods mainly discuss calibration based on samples from the true data distribution $\{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \sim \mathtt{p}(\boldsymbol{x}, y), y \in \{1, ..., K\}$ (Zadrozny & Elkan, 2002; Vaicenavicius et al., 2019), i.e. the probabilistic framework is built based on $N \to +\infty$ limit of the standard definition of

$$\mathtt{P}(y = \hat{y}|\mathtt{P}(\hat{y}|\boldsymbol{x}, \boldsymbol{\theta}) = p) = p \iff \lim_{N \to +\infty} \sum_{i \in \mathcal{S}_j^p} \frac{\delta(y^{(i)}), \hat{y}^{(i)})}{|\mathcal{S}_j^p|} = p \tag{1}$$

where $\mathcal{S}_j^p = \{i|\mathtt{P}(\hat{y}^{(i)} = j|\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = p, i = 1, ..., N\}$. However, Eq. (1) doesn't explicitly reflect the relation between $\mathtt{P}(y = \hat{y}|\mathtt{P}(\hat{y}|\boldsymbol{x}, \boldsymbol{\theta})$ and $\mathtt{p}(\boldsymbol{x}, y)$. We propose to establish the theoretical basis based on $\mathtt{p}(\boldsymbol{x}, y)$, so that calibration evaluation criteria can be strictly defined.

Among the many calibration methodologies, deep ensembling (Lakshminarayanan et al., 2017) emerges as one of the most attractive strategies, that provides not only state-of-the-art calibration/uncertainty performance, but also significant accuracy improvement, with simple implementation. However, it hasn't been theoretically analyzed that why ensembling helps calibration. What's more, recent literature also report the empirical "contradictory" observations that ensembling calibrated members leads to ill-calibrated ensemble (Rahaman & Thiery, 2020; Wen et al., 2020). In this paper, we try to explain the reason via theoretical and empirical analysis. We found that ensembling operation tends to decrease the confidence of predictions. Therefore, the ensemble members

should be encouraged to be slightly overconfident before combination. This leads to the strategy of calibrating the ensemble as a whole.

In this study, we focus on post-hoc calibration of ensemble, based on temperature annealing. Guo et al. (2017) conducted thorough comparison of various existing post-hoc calibration methods and found that temperature scaling, as the simplest, fastest, and most straightforward of the methods, is often the most effective. However, the scaling is globally static for all regions of the input samples, i.e. all samples are scaled towards one single direction, either increasing or decreasing the distribution entropy. To remedy this dogmatic operation that may hurt some legitimately confident predictions, we investigate the effect of region dynamical temperatures. Empirically results demonstrate the effectiveness. We further extend the analysis to ensemble of different model topologies. We show that combining calibrated structures with bayesian model evidence-based weights is better than naively averaging the structures.

2 Related Work

Calibration is inherently related to uncertainty modeling. Two of the most important scopes of calibration are calibration evaluation and calibration system construction.

One direct evaluation tool is the reliability diagram (Vaicenavicius et al., 2019; Bröcker, 2012). It is still desirable to have a overall metric. Widmann et al. (2019) investigate different distance of probability simplex for estimating the calibration errors. Nixon et al. (2019) point out the problem of fixed spaced binning scheme that the bins with few predictions may have low-bias but high-variance measurement. Calibration error measure adaptive to dense populated regions are then proposed (Nixon et al., 2019). Vaicenavicius et al. (2019) treated the calibration evaluation as hypotheses tests.

In obtaining calibrated systems, there are mainly two approaches. The first is to recalibrate the uncalibrated systems with post-hoc calibration mapping, e.g. platt scaling(Platt et al., 1999), isotonic regression(Zadrozny & Elkan, 2002), Dirichlet calibration (Kull et al., 2017; 2019). The second is to directly build calibrated systems, via (1) improving model structures, e.g. deep Convolutional Gaussian processes (Tran et al., 2019); (2) data augmentation, e.g. adversarial samples(Stutz et al., 2020) or Mixup(Zhang et al., 2018); (3) minimize calibration error during training (Kumar et al., 2018).

Calibrators based on histogram binning(Zadrozny & Elkan, 2001), Bayesian binning(Naeini et al., 2015) and scaling binning (Kumar et al., 2019) are related to our proposed dynamic temperature scaling, in the sense that the samples are divided into regions and separate calibration mapping are applied. However, our method can preserve the property that all predictions belonging to one sample sum to 1. The region-based classifier by Kuleshov & Liang (2015) is also related with our proposal.

Ensemble diversity is advocated for better calibration performance (Raftery et al., 2005; Stickland & Murray, 2020). In the work by Zhong & Kwok (2013), ensembles of SVM, logistic regressor, boosted decision trees are investigated. the combination weights of calibrated probabilities is based on AUC of ROC. This study is different in two aspects: (1) we investigate the combination of different deep neural network structures. (2) we use the likelihood-based weights, with the consideration that likelihood are better comparable than AUC values across different models (Ashukha et al., 2020).

3 Calibration Framework

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{Y} = \{\omega_1, ..., \omega_K\}$ be the output space consisting of K classes. The true distribution to be modeled is $\mathtt{p}(\boldsymbol{x}, \omega) = \mathtt{P}(\omega|\boldsymbol{x})\mathtt{p}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}, \omega \in \mathcal{Y}$. Given some training data $\mathcal{D} \sim \mathtt{p}(\boldsymbol{x}, \omega)$, a model $\boldsymbol{\theta}$ is learned to predict the distribution given observation features $\mathtt{P}(\omega|\boldsymbol{x}, \boldsymbol{\theta})$.

For all forms of calibration discussed, there are two extreme cases that will yield perfect calibration:

- $\mathtt{P}(\omega_j|\boldsymbol{x}; \boldsymbol{\theta}) = \mathtt{P}(\omega_j)$, all predictions are equally the prior distribution.

- $\mathtt{P}(\omega_j|\boldsymbol{x};\boldsymbol{\theta}) = \frac{\mathtt{p}(\boldsymbol{x},\omega_j)}{\sum_{k=1}^{K}\mathtt{p}(\boldsymbol{x},\omega_k)}$, the minimum Bayes risk classifier is by definition perfectly calibrated.

Note that perfect calibration doesn't imply high accuracy, as shown by the above system predicting prior distribution.

3.1 Distribution Calibration

A system is calibrated if the predictive probability values can accurately indicate the portion of correct predictions. For example, for the predictions with probability of 0.4, there is a chance of 40% that the predictions are correct according to the true distribution. *Perfect calibration* for the system that yields $\mathtt{P}(\boldsymbol{\omega}|\boldsymbol{x},\boldsymbol{\theta})$ according to the true distribution $\mathtt{p}(\boldsymbol{x},\boldsymbol{\omega})$ can be defined as:

$$\int_{\boldsymbol{x}\in\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)} \mathtt{P}(\omega_j|\boldsymbol{x},\boldsymbol{\theta})\mathtt{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{\boldsymbol{x}\in\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)} \mathtt{P}(\omega_j|\boldsymbol{x})\mathtt{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \quad \forall p,\omega_j,\epsilon\to 0 \tag{2}$$

$$\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon) = \left\{\boldsymbol{x}\middle||\mathtt{P}(\omega_j|\boldsymbol{x},\boldsymbol{\theta})-p|\le\epsilon,\boldsymbol{x}\in\mathcal{X}\right\} \tag{3}$$

$\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)$ denotes the region of input space where the system predictive probability for class ω_j is sufficiently close, within error of ϵ, to the probability p. Calibration means that upon every such region, the expected predictive probability (left side of Eq. (2)) is identical to the expected correctness, i.e., expected true probability (right side of Eq. (2)).

$\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)$ defines the region of what specific calibration is required. For *top-label calibration*, only the most probable class is considered and the region based on Eq. (3) is:

$$\tilde{\mathcal{R}}_j^p(\boldsymbol{\theta},\epsilon) = \mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)\cap\left\{\boldsymbol{x}\middle|\omega_j = \arg\max_{\omega}\mathtt{P}(\omega|\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{x}\in\mathcal{X}\right\} \tag{4}$$

Eq. (4) is a strict subset of Eq. (3). When the regions vary, Eq. (2) doesn't necessary hold. Therefore perfect calibration doesn't imply top-label calibration, and vise versa. One example given in Appendix shows that upon the region in Eq. (3) the system is averagely calibrated, while on the region with only top labels in Eq. (4), the system is uncalibrated. One exception is binary classification, where the regions for top-label calibration are equivalent to those considered in perfect calibration, i.e. $\tilde{\mathcal{R}}_j^p(\boldsymbol{\theta},\epsilon) = \mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)$. Hence, perfect calibration is equivalent to top-label calibration in binary classification (Nguyen & O'Connor, 2015).

Eq. (2) defines the requirements for a perfectly calibrated system. It is useful to define metrics that allow how close a system is to perfect calibration to be assessed. Define

$$\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon) = \int_{\boldsymbol{x}\in\mathcal{R}_j^p(\boldsymbol{\theta},\epsilon)} (\mathtt{P}(\omega_j|\boldsymbol{x},\boldsymbol{\theta})-\mathtt{P}(\omega_j|\boldsymbol{x}))\mathtt{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{5}$$

This then allows two forms of expected calibration losses to be defined

$$\mathtt{ACE}(\boldsymbol{\theta}) = \int_0^1\left|\sum_{j=1}^{K}\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon)\right|\mathrm{d}p;\quad \mathtt{ACCE}(\boldsymbol{\theta}) = \sum_{j=1}^{K}\int_0^1\left|\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon)\right|\mathrm{d}p \tag{6}$$

$$\mathtt{ACE}(\boldsymbol{\theta}) = \frac{1}{K}\int_0^1\left|\sum_{j=1}^{K}\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon)\right|\mathrm{d}p;\quad \mathtt{ACCE}(\boldsymbol{\theta}) = \frac{1}{K}\sum_{j=1}^{K}\int_0^1\left|\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon)\right|\mathrm{d}p \tag{7}$$

All Calibration Error (ACE) only considers the expected calibration error for a particular probability, irrespective of the class associated with the data (Hendrycks et al., 2019). Hence, All Class Calibration Error (ACCE) that requires that all classes minimises the calibration error for all probabilities over all classes is advocated by Kull et al. (2019); Kumar et al. (2019). Naeini et al. (2015) also propose to only consider the class with maximum error. Nixon et al. (2019) propose the Thresholded Adaptive Calibration Error (TACE) to consider only the prediction larger than a threshold τ, and it can be described as a special case of ACCE by replacing the integral range:

$$\mathtt{TACE}(\boldsymbol{\theta}) = \sum_{j=1}^{K}\int_{\tau}^1\left|\mathcal{C}_j^p(\boldsymbol{\theta},\epsilon)\right|\mathrm{d}p \tag{8}$$

Though measures such as ACE and ACCE require consistency of the expected posteriors with the true distribution, for tasks with multiple classes, particularly large numbers of classes, the same weight is given to the ability of the model to assign low probabilities to highly unlikely classes, and high probabilities to the "correct" class. For systems with large numbers of classes this can yield artificially low scores. To address this problem it is more common to replace the regions in Eq 5 with the top-label regions in Eq 4, to give a top-label calibration error $\tilde{C}_j^p(\boldsymbol{\theta}, \epsilon)$. This then yields the the expected top-label equivalents of ACCE and ACE, Expected Class Calibration Error (ECCE) and Expected Calibration Error (ECE). Here for example ECE may be expressed as

$$\mathtt{ECE}(\boldsymbol{\theta}) = \int_0^1 \left| \sum_{j=1}^{K} \int_{\boldsymbol{x} \in \tilde{\mathcal{R}}_j^p(\boldsymbol{\theta}, \epsilon)} (\mathtt{P}(\omega_j|\boldsymbol{x}, \boldsymbol{\theta}) - \mathtt{P}(\omega_j|\boldsymbol{x}))\mathtt{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right| \mathrm{d}p \tag{9}$$

$$= \int_0^1 \mathcal{O}(\boldsymbol{\theta}, p) |\mathtt{Conf}(\boldsymbol{\theta}, p) - \mathtt{Acc}(\boldsymbol{\theta}, p)| \mathrm{d}p \tag{10}$$

where $\mathcal{O}(\boldsymbol{\theta}, p) = \sum_{j=1}^{K} \int_{\boldsymbol{x} \in \tilde{\mathcal{R}}_j^p(\boldsymbol{\theta}, \epsilon)} \mathtt{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ is the fraction observations that are assigned to that particular probability and $\mathtt{Conf}(\boldsymbol{\theta}, p)$ and $\mathtt{Acc}(\boldsymbol{\theta}, p)$ are the ideal distribution accuracy and confidences from the model for that probability. For more details see the appendix.

3.2 Sample-based Calibration

Usually only samples from the true joint distribution are available. Any particular training set is drawn from the distribution to yield

$$\mathcal{D} = \left\{ \{\boldsymbol{x}^{(i)}, y^{(i)}\} \right\}_{i=1}^{N}, \quad \{\boldsymbol{x}^{(i)}, y^{(i)}\} \sim \mathtt{p}(\boldsymbol{x}, \boldsymbol{\omega}), \quad y^{(i)} \in \{\omega_1, ..., \omega_K\}.$$

With the region defined in Eq. (3) changed to indices of the considered samples:

$$\mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon) = \left\{ i \middle| |\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) - p| \leq \epsilon, \boldsymbol{x}^{(i)} \in \mathcal{D} \right\}, \tag{11}$$

the exact forms of calibration in Eq. (2) can be written in terms of the samples from the joint distribution $\mathcal{D}$:

$$\frac{1}{|\mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)|} \sum_{i \in \mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)} \mathtt{P}(\omega_j|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)|} \sum_{i \in \mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)} \delta(y^{(i)}, \omega_j), \quad \forall p, \omega_j, \epsilon \to 0 \tag{12}$$

When considering finite data, in this case N samples, it is important to set ϵ appropriately. Setting different ϵ yields different regions and leads to different calibration results (Kumar et al., 2019). It would be useful to specify ϵ when referring to calibration.

Similarly, the exact form of top-label calibration can be written in terms of samples as Eq. (4), with different regions considered:

$$\tilde{\mathcal{S}}_j^p(\boldsymbol{\theta}, \epsilon) = \mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon) \cap \left\{ i \middle| \omega_j = \arg\max_{\omega} \mathtt{P}(\omega|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{x}^{(i)} \in \mathcal{D} \right\} \tag{13}$$

The sample-based calibration losses on regions $\mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)$ can be defined based on Eq. (12). Eq. (7) and (9) the expected calibration losses over the whole range of [0,1] and all classes are

$$\mathtt{ACE}(\boldsymbol{\theta}, \epsilon) = \frac{1}{N} \sum_{p \in \mathcal{P}(\epsilon)} \left| \sum_{j=1}^{K} \sum_{i \in \mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)} \left(\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) - \delta(y^{(i)}, \omega_j) \right) \right| \tag{14}$$

where $\mathcal{P}(\epsilon) = \{p | p = \min\{1, (2z-1)\epsilon\}, z \in \mathbb{Z}^+\}$, and $\mathbb{Z}^+$ is the set of positive integers. The measure of ECE relating to Eq. (9), which only considers the top regions in Eq. (13) can be defined as

$$\mathtt{ECE}(\boldsymbol{\theta}, \epsilon) = \frac{1}{N} \sum_{p \in \mathcal{P}(\epsilon)} \left| \sum_{j=1}^{K} \sum_{i \in \tilde{\mathcal{S}}_j^p(\boldsymbol{\theta}, \epsilon)} \left(\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) - \delta(y^{(i)}, \omega_j) \right) \right| \tag{15}$$

$$= \sum_{p \in \mathcal{P}(\epsilon)} \frac{\left(\sum_{j=1}^{K} |\tilde{\mathcal{S}}_j^p(\boldsymbol{\theta}, \epsilon)| \right)}{N} \left| \mathtt{Conf}(\boldsymbol{\theta}, p) - \mathtt{Acc}(\boldsymbol{\theta}, p) \right| \tag{16}$$

It should be noted that for finite number of sample, the regions $\mathcal{S}_j^p(\boldsymbol{\theta}, \epsilon)$ and $\tilde{\mathcal{S}}_j^p(\boldsymbol{\theta}, \epsilon)$ derived from the samples is different from the theoretical regions, leading to difference between theoretical calibration error measures and the values estimated from the finite samples. This is also referred to as "estimator randomness" by Vaicenavicius et al. (2019). An example is given in Appendix to illustrate this mismatch. For these sample based versions of calibration errors the weighting term, γ_j^p, is not required as the approproriate weighting is obtained from the relative frequency of the samples, and associated classes, drawn from the ideal distributioin.

The simplest requirement for calibration is $\epsilon = 1$. In this case, $|\mathcal{S}_j^p(\boldsymbol{\theta}, 1)| = N$, and the "minimum" perfect calibration requirement for a system with parameters $\boldsymbol{\theta}$ becomes

$$\frac{1}{N}\sum_{i=1}^{N} \mathtt{P}(\omega_j|\boldsymbol{x}^{(i)};\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \delta(y^{(i)}, \omega_j), \quad \forall \omega_j \tag{17}$$

This is also referred to as *global calibration* in this paper. Similarly,*global top-label calibration* can be defined as

$$\frac{1}{N}\sum_{i=1}^{N} \mathtt{P}(\omega_{\mathtt{max}}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \delta(y^{(i)}, \omega_{\mathtt{max}}^{(i)}), \quad \omega_{\mathtt{max}}^{(i)} = \arg\max_{\omega} \mathtt{P}(\omega|\boldsymbol{x}^{(i)};\boldsymbol{\theta}) \tag{18}$$

4 Ensemble Calibration

Given the data $\mathcal{D}$ the observed features $\boldsymbol{x}^* \in \mathcal{X}$, the prediction of class ω can be expressed as:

$$\begin{aligned} \mathtt{P}(\omega|\boldsymbol{x}^*, \mathcal{D}) &= \mathbb{E}_{\boldsymbol{\theta}\sim\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathtt{P}(\omega|\boldsymbol{x}^*, \boldsymbol{\theta})] = \int \mathtt{P}(\omega|\boldsymbol{x}^*, \boldsymbol{\theta})\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})\mathrm{d}\boldsymbol{x} \\ &\approx \frac{1}{M}\sum_{m=1}^{M} \mathtt{P}(\omega|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)}); \quad \boldsymbol{\theta}^{(m)} \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D}) \end{aligned} \tag{19}$$

where Eq. (19) is an ensemble, Monte-Carlo approximation to the full Bayesian integration, with $\boldsymbol{\theta}^{(m)}$ the m-th member parameters. The predictions of ensemble and members are:

$$\hat{y}_m = \arg\max_{\omega}\{\mathtt{P}(\omega|\boldsymbol{x};\boldsymbol{\theta}^{(m)})\}, \quad \hat{y}_{\mathtt{E}} = \arg\max_{\omega}\Big\{\frac{1}{M}\sum_{m=1}^{M} \mathtt{P}(\omega|\boldsymbol{x};\boldsymbol{\theta}^{(m)})\Big\} \tag{20}$$

It is useful to discuss the relation of the calibration of members and the calibration of the corresponding ensemble. Calibration is defined as integrals over regions $\mathcal{R}_j^p(\boldsymbol{\theta}, \epsilon)$ and $\tilde{\mathcal{R}}_j^p(\boldsymbol{\theta}, \epsilon)$. Unless system yield identical performance for all classes for all observations these regions will be different for different members of the ensemble.

Proposition 1. If all members are calibrated and the regions are the same, The ensemble is also calibrated on the same regions.

Proof. All proofs in this paper are included in Appendix. □

If the members are calibrated, but on different regions, the combined ensemble is not always calibrated. One example is shown in Algorithm 1. Figure 1 shows calibration errors of the ensemble and members.

Proposition 2. If all members are globally calibrated, then the overall ensemble is globally calibrated.

Proposition 3. If all members and the corresponding ensemble are globally top-label calibrated, the ensemble performance is no better than the average performance of the members:

$$\frac{1}{N}\sum_{i=1}^{N} \delta(y^{(i)}, \hat{y}_{\mathtt{E}}^{(i)}) \quad \leq \quad \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N}\sum_{i=1}^{N} \delta(y^{(i)}, \hat{y}_{\mathtt{m}}^{(i)})\Big) \tag{21}$$

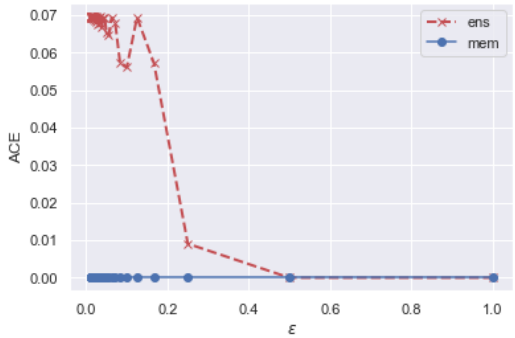


Figure 1: ACE of ensemble that combines members calibrated on different regions. The ensemble is globally calibrated, as shown by $\epsilon \geq 0.5$. However, since the members are not calibrated on the same region, the corresponding ensemble is actually uncalibrated.

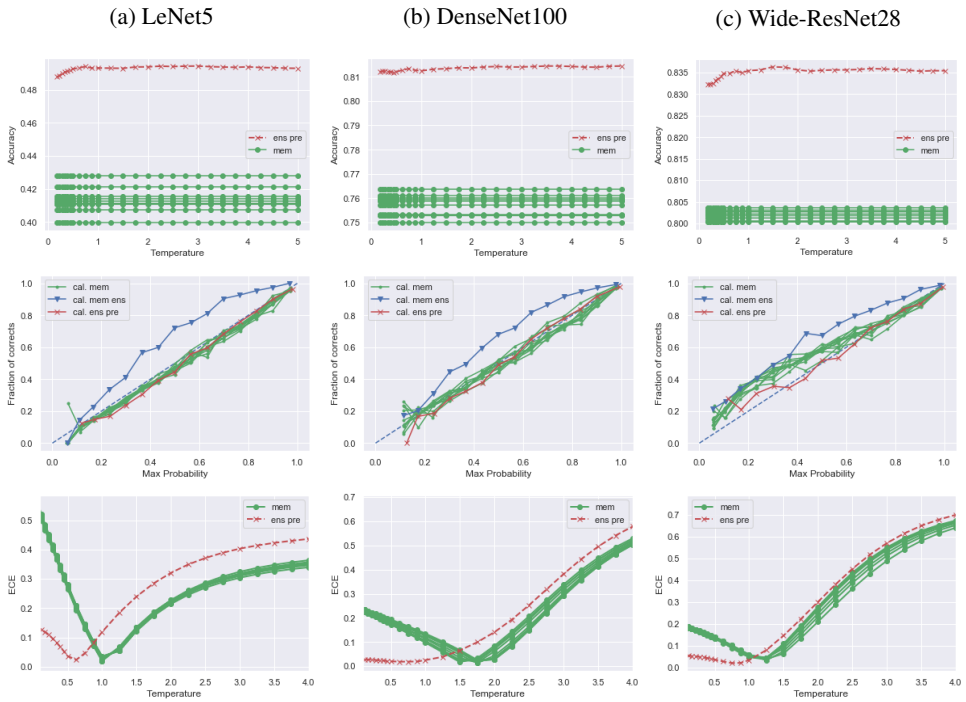


Figure 2: Top-label calibration error and accuracy of members (mem) and the whole ensemble (ens) on *cifar100* (test set) using *LeNet*, *DenseNet* and *ResNet*. "pre" denotes the calibration where shared temperature is applied to members *before* combination. The reliability curves shows the calibrated members and calibrated ensembles with optimal temperature values (estimated on test set).[1]

Therefore, given the ensemble performance is better than the average performance of members, if the ensemble is desired to be calibrated, then *the members shouldn't be calibrated*.

As observed in upper row of Figure 2 that ensemble accuracy is higher than the maximum member accuracy, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathrm{E}}^{(i)}) > \max_{m}\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{m}^{(i)}) \geq \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{m}^{(i)})\right) \tag{22}$$

[1]Accuracy discrepancy with original papers because 10% of training data are held out for validation. Performance of full data training is included in appendix.

As shown in middle row of Figure 2, the members are calibrated, then according to Eq. (18):

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_m^{(i)}) = \frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\hat{y}_m^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{\theta}^{(m)}), \quad m=1,...,M \tag{23}$$

by definition,

$$\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_m^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{\theta}^{(m)}) \geq \frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{\theta}^{(m)}) \tag{24}$$

Combining Eq. (22)-(24), the ensemble is under-confident:

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathtt{E}}^{(i)}) > \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{\theta}^{(m)})\right) \tag{25}$$

The bottom row in Figure 2 also verifies that the ensemble is under-confident than members. The ensemble requires lower temperature for scaling up the confidence.

4.1 Temperature Scaling for Calibration

Having show that calibrating members hurts the calibration performance of ensemble, if the ensemble accuracy is higher than member average, in this section, we investigate the calibration of ensemble. Establishing a function $f:[0,1]\rightarrow[0,1]$ for scaling probabilities.
There are two modes for calibrating an ensemble:
Pre-combination Mode: the function is applied to the probabilities predicted by members, before the combination of members to obtain ensemble prediction.

$$\mathtt{P}_{\mathtt{pre}}(\hat{y}_{\mathtt{E}}|\boldsymbol{x};\boldsymbol{\Theta}) \quad = \quad \frac{1}{M}\sum_{m=1}^{M} f\Big(\mathtt{P}(\hat{y}_{\mathtt{E}}|\boldsymbol{x};\boldsymbol{\theta}^{(m)})\Big) \tag{26}$$

Post-combination Mode: the function is applied to the ensemble predicted probability after combining members' prediction.

$$\mathtt{P}_{\mathtt{post}}(\hat{y}_{\mathtt{E}}|\boldsymbol{x};\boldsymbol{\Theta}) \quad = \quad f\left(\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_{\mathtt{E}}|\boldsymbol{x};\boldsymbol{\theta}^{(m)})\right) \tag{27}$$

There are many functions for transforming predicted probability in previous calibration literature, e.g. histogram binning, Platt scaling and temperature annealing. However, note that histogram binning shouldn't be adopted in the pre-combination mode as scaling function f for calibrating multi-class ensemble. Because the transformed values may not satisfy the constraint that the sum of the probabilities of all possible outcomes is 1. In such case, the combination in Eq. (26) is meaningless.

As demonstrated by Guo et al. (2017), temperature scaling is a promising option of the mapping function f:

$$f(\boldsymbol{z};t) = \frac{\exp\{\boldsymbol{z}/t\}}{\sum_j \exp\{z_j/t\}} \tag{28}$$

Previous method uses one single temperature for scaling the posterior probabilities of all samples. Which leads to the problem that the predictions are treated as either all over-confident or all less-confident. According to Eq. (2), ideally, for each region a temperature is used to adjust the confidence. To dynamically determine the required temperature, the samples in validation set are divided into R regions based on the max probability of ensemble predictions (e.g. $\mathcal{R}_1=[0,0.3)$, $\mathcal{R}_2=[0.3,0.6)$, and $\mathcal{R}_3=[0.6,1]$). Each region has an individual temperature for scaling $\{\mathcal{R}_r,t_r\}_{r=1}^{R}$.

$$f_{\mathtt{dyn}}(\boldsymbol{z};\boldsymbol{t}) = \frac{\exp\{\boldsymbol{z}/t_r\}}{\sum_j \exp\{z_j/t_r\}}, \quad \text{if } \max_i \frac{\exp\{z_i\}}{\sum_j \exp\{z_j\}} \in \mathcal{R}_r \tag{29}$$

Model	Cal.	Acc.(%)	NLL	ACCE(10^{-4})	ACE(10^{-4})	ECCE(10^{-2})	ECE(10^{-2})
LEN	–	49.32	1.9759	30.68	23.81	16.47	11.83
	pre	49.37	1.9640	22.93	8.66	13.09	3.19
	post	49.32	1.9290	21.65	5.96	13.38	**2.02**
	dyn.	49.32	**1.9275**	**20.84**	**4.43**	**12.90**	2.05
DSN	–	81.24	0.6704	16.50	5.63	8.73	2.48
	pre	81.25	0.6901	16.89	6.67	8.60	1.98
	post	81.24	0.6886	16.89	6.63	8.57	1.95
	dyn.	81.24	**0.6764**	**16.24**	**4.87**	**8.36**	**1.11**
RSN	–	83.54	0.6254	17.02	7.62	9.20	3.42
	pre	83.50	0.6143	**15.33**	**2.36**	8.88	1.81
	post	83.54	0.6129	15.54	3.02	**8.83**	1.75
	dyn.	83.54	**0.6119**	15.75	2.75	8.90	**0.83**

Table 1: Calibration using temperature annealing on CIFAR-100. The temperatures are optimized to minimize ECE on validation set. In the 'pre' mode, each member is scaled with one separate temperature. 'dyn.' denotes dynamic temperature scaling in post-combination mode using 6 region-based temperatures. The three structures investigated are LeNet 5, DenseNet 100 and Wide ResNet 28.

Model	Cal.	Acc.(%)	NLL	ACCE(10^{-4})	ACE(10^{-4})	ECCE(10^{-2})	ECE(10^{-2})
LEN	–	80.15	0.6071	18.85	18.12	9.57	9.26
	pre	80.01	0.5918	7.89	5.32	4.13	1.77
	post	80.15	**0.5671**	**5.90**	**2.61**	**3.67**	**1.31**
	dyn.	80.15	0.5776	6.10	3.31	3.95	1.65
DSN	–	96.19	0.1229	2.87	1.38	**1.40**	0.58
	pre	96.18	0.1227	2.91	1.38	1.42	0.59
	post	96.19	0.1230	2.92	1.36	1.41	0.57
	dyn.	96.19	**0.1225**	**2.84**	**1.19**	1.41	**0.50**
RSN	–	96.49	0.1090	2.54	0.94	1.33	0.52
	pre	96.51	**0.1089**	2.53	0.74	**1.22**	0.46
	post	96.49	**0.1089**	2.53	0.69	1.30	0.46
	dyn.	96.49	0.1090	**2.46**	**0.61**	1.26	**0.33**

Table 2: Calibration using temperature annealing on CIFAR-10. The temperatures are optimized to minimize ECE on validation set. In the 'pre' mode, each member is scaled with one separate temperature. 'dyn.' denotes dynamic temperature scaling in post-combination mode using 6 region-based temperatures. The three structures investigated are LeNet 5, DenseNet 100 and Wide ResNet 28.

5 Ensemble of Multiple Model Topologies

Assume that we now have multiple, M, network topologies trained and marginalised over the data that can yield an ensemble of posteriors

$$\mathtt{P}(\omega_j|\boldsymbol{x}, \mathcal{M}^{(1)}), ..., \mathtt{P}(\omega_j|\boldsymbol{x}, \mathcal{M}^{(M)}) \tag{30}$$

The task is to optimally combine the set of classifiers in a linear fashion given access to the true joint distribution $\mathtt{p}(\boldsymbol{\omega}, \boldsymbol{x})$. Again a simple linear combination approach is to be adopted

$$\mathtt{P}(\omega_j|\boldsymbol{x}, \mathcal{M}^{(1)}, ..., \mathcal{M}^{(M)}) = \sum_{m=1}^{M} \alpha_m \mathtt{P}(\omega_j|\boldsymbol{x}, \mathcal{M}^{(m)}) \tag{31}$$

As usual there are only samples from the joint distribution so maximum likelihood training of the samples can be used

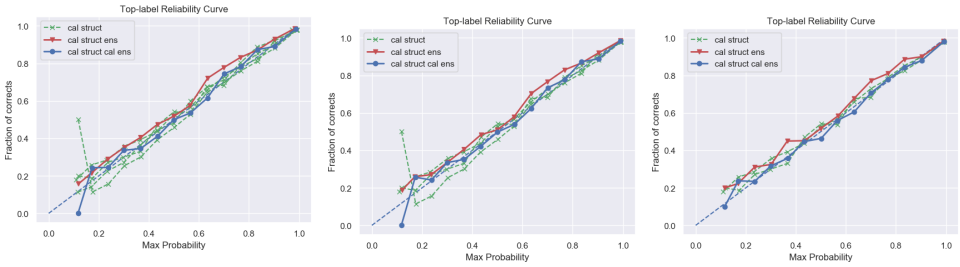
$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{N} \log \left(\sum_{m=1}^{M} \alpha_m \mathtt{P}(y^{(i)}|\boldsymbol{x}^{(i)}, \mathcal{M}^{(m)}) \right) \right\} \tag{32}$$

Expectation maximisation can then be used to find the optimal values of $\boldsymbol{\alpha}$. However, this is again an ensemble and previous shown propositions suggest a further calibration is required. In Figure 3,

Weight Est.	**Comb. Weight**				**Ens Cal.**	**Acc.**(%)	**NLL**	**ACE**(10^{-4})	**ECE**(10^{-2})
	LEN	DSN100	DSN121	RSN					
Avg. AUC	0.20	0.26	0.27	0.27	-	83.30	0.6649	20.84	10.07
	0.20	0.26	0.27	0.27	✓	83.30	0.6096	2.60	2.00
	-	0.33	0.33	0.34	-	83.55	0.5776	3.67	1.89
	-	0.33	0.33	0.34	✓	83.55	0.5787	3.03	1.38
	-	-	0.49	0.51	-	83.78	0.5820	3.54	2.03
	-	-	0.49	0.51	✓	83.78	0.5805	2.30	1.34
Max LL	0.02	0.19	0.30	0.49	-	83.75	0.5766	4.97	2.24
	0.02	0.19	0.30	0.49	✓	83.75	0.5698	1.42	1.20
	-	0.22	0.30	0.48	-	83.80	0.5741	3.74	2.00
	-	0.22	0.30	0.48	✓	83.80	0.5714	1.52	1.29
	-	-	0.44	0.56	-	83.86	0.5816	3.64	2.06
	-	-	0.44	0.56	✓	83.86	0.5801	2.36	1.35

Table 3: Combining calibrated structure still requires calibration. The four structures are calibrated first and combined with weights. Calibrations of structure and ensemble adopt post-combination mode.

the ensemble of calibrated structures is less calibrated than the structures. A final calibration can improve the performance significantly, as shown in Table 3. We investigate the above maximum likelihood (Max LL) weight estimation and the estimation based on AUC averaging (Avg AUC) by Zhong & Kwok (2013) for combining structures. The results in Table 3 show that both combination methods degrade calibration performance and a final calibration helps to reduce the ECE values.



(a) LEN+DSN100+DSN121+RSN (b) DSN100+DSN121+RSN (c) DSN121+RSN

Figure 3: Reliability curve of weighted combination of 4 calibrated structures, LeNet, DenseNet 100 and DenseNet 121 and ResNet wide. Each structure is an ensemble of 10 models.

6 Experiments

We conduct experiments with LeNet, DenseNet and ResNet on cifar100 dataset. The model setup and training detail are included in Appendix.

7 Conclusions

Reproducibility

`https://github.com/EncodeWood/EnsembleCalibration`

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.

Jochen Bröcker. Estimating reliability and resolution of probability forecasts through decomposition of the empirical score. *Climate dynamics*, 39(3-4):655–667, 2012.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *ICML*, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019.

Volodymyr Kuleshov and Percy S Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems*, pp. 3474–3482, 2015.

Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pp. 623–631, 2017.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pp. 12316–12326, 2019.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pp. 3792–3803, 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, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, pp. 2901. NIH Public Access, 2015.

Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. *EMNLP*, 2015.

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

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, pp. 38–41, 2019.

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.

Adrian E Raftery, Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski. Using bayesian model averaging to calibrate forecast ensembles. *Monthly weather review*, 133(5):1155–1174, 2005.

Rahul Rahaman and Alexandre H Thiery. Uncertainty quantification and deep ensembles. *arXiv preprint arXiv:2007.08792*, 2020.

Asa Cooper Stickland and Iain Murray. Diverse ensembles improve calibration. *ICML 2020 workshop on Uncertainty Robustness in Deep Learning*, 2020.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. *ICML 2020 workshop on Uncertainty Robustness in Deep Learning*, 2020.

Gia-Lac Tran, Edwin V Bonilla, John Cunningham, Pietro Michiardi, and Maurizio Filippone. Calibrating deep convolutional gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1554–1563. PMLR, 2019.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B Schön. Evaluating model calibration in classification. *Proceedings of Machine Learning Research*, 2019.

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Improving calibration of batchensemble with data augmentation. *ICML 2020 workshop on Uncertainty Robustness in Deep Learning*, 2020.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *Advances in Neural Information Processing Systems*, pp. 12257–12267, 2019.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pp. 609–616. Citeseer, 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.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018.

Wenliang Zhong and James T Kwok. Accurate probability calibration for multiple classifiers. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

A APPENDIX

A.1 THEORETICAL PROOF

Proof of Proposition 1. If all members are calibrated and the regions are the same, The ensemble is also calibrated on the same regions.

Proof. If different members $\boldsymbol{\theta}^{(m)}$ and $\boldsymbol{\theta}^{(m')}$ are calibrated and the regions are the same, i.e. the following equation holds:

$$\mathcal{R}_j^p(\boldsymbol{\theta}^{(m)}, \epsilon) = \mathcal{R}_j^p(\boldsymbol{\theta}^{(m')}, \epsilon) \quad \forall p, \omega_j, \quad \epsilon \to 0$$

The ensemble is also calibrated and the regions are the same:

$$\mathcal{R}_j^p(\boldsymbol{\Theta}, \epsilon) = \left\{ \boldsymbol{x} \middle| \left| \frac{1}{M} \sum_{m=1}^{M} \mathtt{P}(\omega_j | \boldsymbol{x}; \boldsymbol{\theta}^{(m)}) - p \right| \leq \epsilon \right\} = \mathcal{R}_j^p(\boldsymbol{\theta}^{(m)} \quad \forall p, \omega_j, \quad \epsilon \to 0 \tag{33}$$

$$\begin{aligned}
\int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\Theta}, \epsilon)} \frac{1}{M} \sum_{m=1}^{M} \mathtt{P}(\omega_j | \boldsymbol{x}; \boldsymbol{\theta}^{(m)}) \mathtt{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} &= \frac{1}{M} \sum_{m=1}^{M} \int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\theta}^{(m)}, \epsilon)} \mathtt{P}(\omega_j | \boldsymbol{x}; \boldsymbol{\theta}^{(m)}) \mathtt{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{M} \sum_{m=1}^{M} \int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\theta}^{(m)}, \epsilon)} \mathtt{P}(\omega_j | \boldsymbol{x}) \mathtt{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\theta}^{(m)}, \epsilon)} \mathtt{P}(\omega_j | \boldsymbol{x}) \mathtt{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\Theta}, \epsilon)} \mathtt{P}(\omega_j | \boldsymbol{x}) \mathtt{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}
\end{aligned}$$

□

Proof of Proposition 2. If all members are globally calibrated, then the overall ensemble is globally calibrated.

Proof. If all members are globally calibrated, then

$$\frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)}) \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\omega_j), \quad \forall\omega_j \tag{34}$$

Hence,

$$\frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)};\boldsymbol{\Theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\omega_j|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)})\right) \tag{35}$$

$$= \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\omega_j)\right) = \frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\omega_j) \tag{36}$$

□

Proof of Proposition 3. If all members and the corresponding ensemble are globally top-label calibrated, the ensemble performance is no better than the average performance of the members:

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathtt{E}}^{(i)}) \quad \leq \quad \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathtt{m}}^{(i)})\Big) \tag{37}$$

Proof. If all members and the ensemble are globally top-label calibrated,

$$\frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\hat{y}_m^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)}) \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_m^{(i)}), \quad m=1,...,M \tag{38}$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)})\right) \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathtt{E}}^{(i)}) \tag{39}$$

By definition,

$$\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)}) \leq \mathtt{P}(\hat{y}_m^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)}) \tag{40}$$

Hence,

$$\frac{1}{N}\sum_{i=1}^{N}\Big(\frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)})\Big) \quad = \quad \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\hat{y}_{\mathtt{E}}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)})\Big) \tag{41}$$

$$\leq \quad \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N}\sum_{i=1}^{N}\mathtt{P}(\hat{y}_m^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)})\Big) \tag{42}$$

Combining Eq. (38), (39) and (42):

$$\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_{\mathtt{E}}^{(i)}) \quad \leq \quad \frac{1}{M}\sum_{m=1}^{M}\Big(\frac{1}{N}\sum_{i=1}^{N}\delta(y^{(i)},\hat{y}_m^{(i)})\Big) \tag{43}$$

□

A.2 Toy Datasets

Example 1. In this example, we show the mismatch between calibration and top-label calibration on the different regions in Eq. (3) and (4).

Assuming $\mathtt{p}(\boldsymbol{x}) \propto 1$, the whole input space $\mathcal{X}$ is consisted of three regions $\mathcal{R}_1, \mathcal{R}_2$ and $\mathcal{R}_3$, and

$$\int_{\boldsymbol{x}\in R_1}\mathtt{p}(\boldsymbol{x})d\boldsymbol{x} = \int_{\boldsymbol{x}\in R_2}\mathtt{p}(\boldsymbol{x})d\boldsymbol{x} = \int_{\boldsymbol{x}\in R_3}\mathtt{p}(\boldsymbol{x})d\boldsymbol{x}. \tag{44}$$

The corresponding system prediction $\hat{\boldsymbol{P}}$ and the true distribution $\boldsymbol{P}$ is:

$$\hat{\boldsymbol{P}} = \Big(\mathtt{P}(\omega_j|\boldsymbol{x} \in \mathcal{R}_r; \boldsymbol{\theta})\Big) = \begin{array}{c} \begin{array}{cccc} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{array} \\ \begin{pmatrix} 0.5 & \underline{0.4} & 0.05 & 0.05 \\ 0.3 & \underline{0.4} & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.35 & 0.05 \end{pmatrix} \end{array} \begin{array}{c} \\ \mathcal{R}_1 \\ \mathcal{R}_2 \\ \mathcal{R}_3 \end{array} \tag{45}$$

$$\boldsymbol{P} = \begin{array}{c} \begin{array}{ccccc} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{array} \\ \begin{pmatrix} 0.5 & \underline{0.4-\tau} & 0.05 & 0.05+\tau \\ 0.3-\tau & \underline{0.4+\tau} & 0.2 & 0.1 \\ 0.3+\tau & 0.3 & 0.35 & 0.05-\tau \end{pmatrix} \end{array} \begin{array}{c} \\ \mathcal{R}_1 \\ \mathcal{R}_2 \\ \mathcal{R}_3 \end{array}, \quad \tau > 0 \tag{46}$$

It can be verified that

Example 2. In this example, we show that

Algorithm 1: Algorithm for generating calibrated members that yield uncalibrated ensemble

Result: $\{\boldsymbol{p}^{(i)}\}_{i=1}^N$,$\{\hat{\boldsymbol{p}}^{(i)}, \hat{y}^{(i)}\}_{i=1}^N$
$b = 2$; //bin size, number of samples in one bin;
$K = 4$; //number of classes;
$M = 10$; //number of members;
$N = 1000000$; //number of samples;
$\boldsymbol{p}^{(i)} \sim$ Dirichlet($\boldsymbol{\alpha} = \mathbf{1}$), $i = 1, ..., N$; //true data distribution, sampled from Dirichlet distribution with equal concentration parameters of 1;
$\boldsymbol{I} = [1, 2, ..., N]$; // index vector;
for m *in* $[1, ..., M]$ **do**
  $\tilde{\boldsymbol{I}} \leftarrow$ shuffle($\boldsymbol{I}$);
  for j *in* $[1, ..., \lceil N/b \rceil]$ **do**
    $\boldsymbol{B}_j = (b * (j-1), \min\{b * j, N\}]$;
    $\hat{\boldsymbol{p}}_{\boldsymbol{B}_j} = \frac{1}{b} \sum_{l \in \boldsymbol{B}_j} \boldsymbol{p}^{(\tilde{I}_l)}$;
  end
  for i *in* $[1, ..., N]$ **do**
    $j = \lceil \frac{i}{b} \rceil$;
    $\hat{\boldsymbol{p}}^{(\tilde{I}_i, m)} = \hat{\boldsymbol{p}}_{\boldsymbol{B}_j}$;
  end
end

Example 3. The theoretical ACE of the predicted probabilities in Algorithm 2 is $\frac{9}{32}$. However, by applying 7 to $\mathcal{D}$, the ACE can be found to be 0. The true data distribution is:

$$\mathtt{P}(\omega_j|\boldsymbol{x}) = \frac{1}{4}, \ \mathtt{p}(\boldsymbol{x}) \propto 1 \tag{47}$$

The samples in $\mathcal{D}$ are assigned to bins of size 2. The type of bin that a sample is assigned to determines the predicted probability of sample. There are three types of bins:
$\mathbf{B}_{p=1}$: both samples belong to class ω_j.
$\mathbf{B}_{p=0.5}$: only one sample is class ω_j.
$\mathbf{B}_{p=0}$: both samples are not class ω_j.

$$\mathtt{P}(\boldsymbol{x} \in \mathbf{B}_1) = \frac{1}{16}, \ \mathtt{P}(\boldsymbol{x} \in \mathbf{B}_{0.5}) = \frac{6}{16}, \ \mathtt{P}(\boldsymbol{x} \in \mathbf{B}_0) = \frac{9}{16} \tag{48}$$

then

$$\int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\theta}, 0)} \big(\mathtt{P}(\omega_j|\boldsymbol{x}; \boldsymbol{\theta}) - \mathtt{P}(\omega_j|\boldsymbol{x})\big)\mathtt{p}(\boldsymbol{x})d\boldsymbol{x} = \int_{\boldsymbol{x} \in \mathcal{R}_j^p(\boldsymbol{\theta}, 0)} \big(p - \frac{1}{4})\big)\mathtt{p}(\boldsymbol{x})d\boldsymbol{x} \tag{49}$$

$$= (p - \frac{1}{4})\mathtt{P}(\boldsymbol{x} \in \mathbf{B}_p) \tag{50}$$

therefore,

$$\mathcal{E}_{\texttt{ECE}}(\boldsymbol{\theta}, \epsilon = 0) = \frac{1}{4}\int_0^1 \Big| \sum_{j=1}^{4} \int_{\boldsymbol{x}\in\mathcal{R}_j^p(\boldsymbol{\theta},0)} \big(\texttt{P}(\omega_j|\boldsymbol{x};\boldsymbol{\theta}) - \texttt{P}(\omega_j|\boldsymbol{x})\big)\texttt{p}(\boldsymbol{x})d\boldsymbol{x}\Big| dp \tag{51}$$

$$= \sum_{p\in\{0,0.5,1\}} \Big|(p - \frac{1}{4})\texttt{P}(\boldsymbol{x} \in \mathbf{B}_p)\Big| = \frac{9}{32} \tag{52}$$

Algorithm 2: Algorithm for generating data with overfitted calibration

Result: $\{\hat{\boldsymbol{p}}^{(i)}, \hat{y}^{(i)}\}_{i=1}^N$
$b = 2$; //bin size, number of samples in one bin;
$K = 4$; //number of classes;
$\boldsymbol{g} = [g_1, ..., g_N]$; //vector of ground-truth labels , where $g \in \{1, ..., K\}$. Numbers of labels for different classes are equal;
$\boldsymbol{I} = [1, 2, ..., N]$; // index vector;
$\tilde{\boldsymbol{I}} \leftarrow$ shuffle($\boldsymbol{I}$);
for j *in* $[1, ..., \lceil N/b \rceil]$ **do**
  $\boldsymbol{B}_j = (b * (j-1), \min\{b * j, N\}]$;
  $\hat{\boldsymbol{p}}_{\boldsymbol{B}_j} = \frac{1}{b}\sum_{l\in\boldsymbol{B}_j}[\delta(1, g_{\tilde{I}_l}), \delta(2, g_{\tilde{I}_l}), ..., \delta(K, g_{\tilde{I}_l})]$;
end
for i *in* $[1, ..., N]$ **do**
  $j = \lceil \frac{i}{b} \rceil$;
  $\hat{\boldsymbol{p}}^{(\tilde{I}_i)} = \hat{\boldsymbol{p}}_{\boldsymbol{B}_j}$;
end