The impact of averaging logits over probabilities on ensembles of neural networks

Cedrique Rovile Njieutcheu Tassi[1], Jakob Gawlikowski[2], Auliya Unnisa Fitri[2] and Rudolph Triebel[3]

[1]German Aerospace Center (DLR), Institute of Optical Sensor Systems, Rutherfordstraße 2, 12489 Berlin, Germany
[2]German Aerospace Center (DLR), Institute of Data Science, Mälzerstraße 3-5, 07745 Jena, Germany
[3]German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Münchener Straße 20, 82234 Wessling, Germany

Abstract

Model averaging has become a standard for improving neural networks in terms of accuracy, calibration, and the ability to detect false predictions (FPs). However, recent findings show that model averaging does not necessarily lead to calibrated confidences, especially for underconfident networks. While existing methods for improving the calibration of combined networks focus on recalibrating, building, or sampling calibrated models, we focus on the combination process. Specifically, we evaluate the impact of averaging logits instead of probabilities on the quality of confidence (QoC). We compare combined logits instead of probabilities of members (networks) for models such as ensembles, Monte Carlo Dropout (MCD), and Mixture of Monte Carlo Dropout (MMCD). Comparison is done using experimental results on three datasets using three different architectures. We show that averaging logits instead of probabilities increase the confidence thereby improving the confidence calibration for underconfident models. For example, for MCD evaluated on CIFAR10, averaging logits instead of probabilities reduces the expected calibration error (ECE) from 12.03% to 5.44%. However, the increase in confidence can bring harm to confidence calibration for overconfident models and the separability between true predictions (TPs) and FPs. For example, for MMCD evaluated on MNIST, the average confidence on FPs due to the noisy data increases from 51.31% to 94.58% when averaging logits instead of probabilities. While averaging logits can be applied with underconfident models to improve the calibration on test data, we suggest to average probabilities for safety- and mission-critical applications where the separability of TPs and FPs is of paramount importance.

Keywords

Model averaging, Combination process, Logit averaging, Probability averaging, Ensemble, Monte Carlo Dropout (MCD), Mixture of Monte Carlo Dropout (MMCD), Quality of confidence (QoC), Confidence calibration, Separating true predictions (TPs) and false predictions (FPs)

1. Introduction

Recently, averaging the predictions of multiple stochastic or deterministic networks has become a standard approach for improving accuracy [1, 2] and uncertainty estimates [3]. Generally, the quality of uncertainty estimates (e.g.: QoC) is assessed by the degree of calibration and/or the ability to detect FPs. Model averaging can yield well-calibrated confidence [4, 5] and is one of the state-of-the-art methods for detecting FPs caused by out-of-distribution examples [4, 3]. However, recent findings [6, 7, 8] show that model averaging does not necessarily lead to calibrated confidence, especially when the networks are built using modern regularization techniques, such as mixup [9] or label smoothing [10, 11]. This is because modern regularization techniques can (strongly) regularize networks, resulting in underconfidence. Furthermore, averaging underconfident networks produce more underconfident networks. For example, [7] showed that averaging networks trained with modern regularization techniques resulted in more underconfident networks and therefore miscalibrated predictions. [12] supported this argument by theoretically and empirically showing that averaging calibrated networks do not always lead to calibrated confidences. Calibrating confidences of averaged networks has received little attention in the literature. Generally, post-processing calibration methods, such as temperature scaling [13], can be used to recalibrate the confidences of averaged networks, as demonstrated in [8, 12]. From [14] and further supported by [8], confidence calibration in model averaging is correlated to diversity inherent in individual networks and the more diverse the networks, the better the calibration. Motivated by this observation, [14] promoted model diversity using structured dropout to reduce calibration errors. [7] proposed class-adjusted mixup that trains less confident networks by evaluating the difference between accuracy (estimated on a validation dataset after each training epoch) and the confidence of each training sample to activate or deactive mixup training for overconfidence (average confidence $>$ accuracy) or underconfidence (average confidence $<$ accuracy), respec-

The IJCAI-ECAI-22 Workshop on Artificial Intelligence Safety (AISafety 2022)
Cedrique.NjieutcheuTassi@dlr.de (C. R. N. Tassi); Jakob.Gawlikowski@@dlr.de (J. Gawlikowski); Auliya.Fitri@dlr.de (A. U. Fitri); Rudolph.Triebel@dlr.de (R. Triebel)

tively. All these methods for improving the calibration of combined networks focus on recalibrating, building, or sampling the calibrated networks. However, this work focuses on combining the networks. Specifically, we address the question: ***What is the impact of averaging logits instead of probabilities of multiple (stochastic or deterministic) networks on the QoC?***

We hypothesized that averaging logits instead of probabilities of multiple networks increases the confidence of the averaged network. This is because logits (inputs to softmax), which can be interpreted as found evidence for possible classes [15], are continuous values normalized using the softmax to produce discrete probabilities. The softmax normalization of continuous values (logits) to discrete values (probabilities) causes information loss and possible robustness to changes in the magnitudes of logits. This implies that the softmax function is a nonlinear function that maps multiple logit vectors with large differences in magnitudes to the same discrete probability vector. We evaluated the impact of the increase in confidence caused by averaging logits instead of probabilities on the QoC. Specifically, we evaluated the QoC by assessing the degree of confidence calibration, which measures the difference between the predicted (average confidence) and true probabilities (empirical accuracy). Furthermore, we evaluated the QoC by assessing its ability to seperate TPs and FPs. To provide empirical evidence for evaluating the QoC, we considered the logit averaging against probability averaging and compared both approaches using different averaged models, such as ensemble, MCD, and MMCD. The comparison was based on results from different experiments conducted on three datasets, namely, MNIST, FashionMNIST, and CIFAR10 evaluated on VGGNet, ResNet, and DenseNet, respectively.

Results show that averaging logits instead of probabilities preserves accuracy, but increases confidence. For example, for MCD evaluated on CIFAR10 (see Table 2), the accuracy remained around 85.36% while the average confidence increased from 73.35% to 80.04% when we averaged logits instead of probabilities. Furthermore, given underconfident models, the increase in the degree of confidence reduces the calibration error on the test data. For example, for MCD evaluated on CIFAR10, ECE dropped from 12.04% to 5.40% when the average confidence increased from 73.35% to 80.04%. However, given overconfident models, the increase in the degree of confidence increased the calibration error on the test data. For example, for the ensemble evaluated on CIFAR10 (see Table 3), ECE increased from 3.03% to 7.40% when the average confidence increased from 89.43% to 96.17%. Finally, for underconfident or overconfident models, the increase in the degree of confidence can harm the separability between TPs and FPs. This is because averaging logits instead of probabilities increases the confidence of both TPs and FPs. Therefore, FPs can be made with high confidence similar to TPs. For example, for MMCD evaluated on FashionMNIST (see Table 4), the average confidence on FPs due to the noisy data increased from 51.31% to 94.58% when averaging logits instead of probabilities. In summary, we provide empirical evidence demonstrating how *combining logits instead of probabilities* of multiple (stochastic or deterministic) networks

- preserves accuracy, but increases the confidence on TPs and FPs.
- reduces the calibration error (given underconfident networks), but increases the calibration error (given overconfident networks).
- can harm the separability between TPs and FPs.

2. Related works

The combination process describes how multiple members are combined and the information type (e.g., logits or probabilities) that is combined. Several approaches such as stacking [16] and voting [17, 18, 19]) have been reported for aggregating multiple predictions. Some of these approaches have been reviewed and discussed in [20, 21] and experimentally compared in [18, 16] to find the one with the best accuracy. It was found that one approach improves accuracy better than another depending on several factors, such as the number of members, diversity inherent in individual members, and accuracy of individual members. However, in [22], we compared approaches such as averaging, plurality voting, or majority voting to find the one that better captures uncertainty. We found that the averaging approach captures uncertainty better than voting approaches. Before our work, [23] argued that simple averaging approaches are more robust than voting approaches. This argument was further supported by [24]. This is because the averaging approach considers all members' predictions, whereas plurality/majority voting ignores uncertain predictions and therefore, reduces the uncertainty in the combined members' prediction. Although various combination approaches have been presented and compared in the literature, the information type that is combined has received relatively little attention. [25] showed that averaging quantiles rather than probabilities improve the predictive performance. Generally, for neural networks and classification problems in particular, multiple members (networks) are combined by averaging probabilities [16]. [16] evaluated the impact of combining logits instead of probabilities on accuracy, however, the impact on the QoC remains unclear. Thus, we investigated the impact of combining logits instead of probabilities on the QoC.

3. Background

In the context of image classification, let the training data $D_{train} = \{x_i \in \mathbb{R}^{H\times W\times C}, y_i \in U^K\}_{i\in[1,N]}$ be a realization of independently and identically distributed random variables $(x, y) \in X \times Y$, where x_i denotes the i^{th} input and y_i its corresponding one hot encoded class label from the set of standard unit vectors of $\mathbb{R}^K$, U^K. X and Y denote the input and label spaces. $H \times W \times C$ denotes the dimension of input images, where H, W, and C refer to the height, weight, and number of channels, respectively. K and N denote the numbers of possible output classes and samples within the training data, respectively.

3.1. Convolutional neural network (CNN)

A CNN is a nonlinear function f_θ parameterized by model parameters θ, called the network weights. Here, it maps input images $x_i \in \mathbb{R}^{H\times W\times C}$ to class labels $y_i \in U^K$,

$$f_\theta : x_i \in \mathbb{R}^{H\times W\times C} \rightarrow y_i \in [0,1]^K;\ f_\theta(x_i) = y_i \quad (1)$$

The network parameters are optimized on the training dataset, D_{train}. Given a new data sample $x \in \mathbb{R}^{H\times W\times C}$, a trained CNN f_θ predicts the corresponding target $y = f_\theta(x)$ using the set of trained weights θ. The network output (logit) is given by $z = f_\theta(x)$, from which a probability vector $p(y|x, D_{train}) = softmax(z)$, can be computed. In the following, this probability vector will be abbreviated by p and its entries by p_k with $k = 1, \ldots, K$ and $\sum_{k=1}^K p_k = 1$. Further, we get the predicted confidence $c = \max_k(p_k)$ and predicted class label $y = \arg\max_k(p_k)$

3.2. Monte Carlo Dropout (MCD)

MCD was investigated in [26, 27, 28] for uncertainty estimation. It is one of the most widespread Bayesian methods reviewed in [3]. It approximates the prediction $p(y|x, D_{train})$ using the mean of S stochastic forward passes, $p(y|x, \theta_1), ..., p(y|x, \theta_S)$, representing S stochastic CNNs parameterized by samples θ_1, θ_2,..., and θ_S. That is

$$\overline{p}(y|x, D_{train}) \approx \frac{1}{S}\sum_{s=1}^{S} p(y|x, \theta_s) \approx \frac{1}{S}\sum_{s=1}^{S} f_{\theta_s}(x). \quad (2)$$

Specifically, MCD approximates the prediction with a dropout distribution realized by sampling weights with masks drawn from known distributions, such as Gaussian, Bernoulli, or a cascade of Gaussian and Bernoulli distributions [22]. For example, given the activation vector a fed to a MCD layer (placed for example at the input of the first fully-connected layer) and assuming that sampling is realized with masks drawn from a cascade of Gaussian and Bernoulli distribution, the MCD layer samples the j^{th} element of a as $a_j^s = a_j * \alpha_j * \beta_j$ with $\alpha_j \sim \mathcal{N}(1, \sigma^2 = q/(1-q))$ and $\beta_j \sim Bernoulli(q)$. Here, q denotes the dropout probability. In this work, we can refer to MCD as an average of S stochastic CNNs.

3.3. Ensemble

An (explicit) ensemble was investigated in [4, 27, 28] for uncertainty estimation. It approximates the prediction $p(y|x, D_{train})$ by learning different settings. Given a set of CNNs f_{θ_m} for $m \in 1, 2, ..., M$, the ensemble prediction is obtained by averaging over the predictions of the CNNs. That is,

$$\begin{aligned}\overline{p}(y|x, D_{train}) &:= \frac{1}{M}\sum_{m=1}^{M} p(y|x, \theta_m) \\ &:= \frac{1}{M}\sum_{m=1}^{M} f_{\theta_m}(x).\end{aligned} \quad (3)$$

In this work, we can refer to an ensemble as an average of M deterministic CNNs.

3.4. Mixture of Monte Carlo Dropout (MMCD)

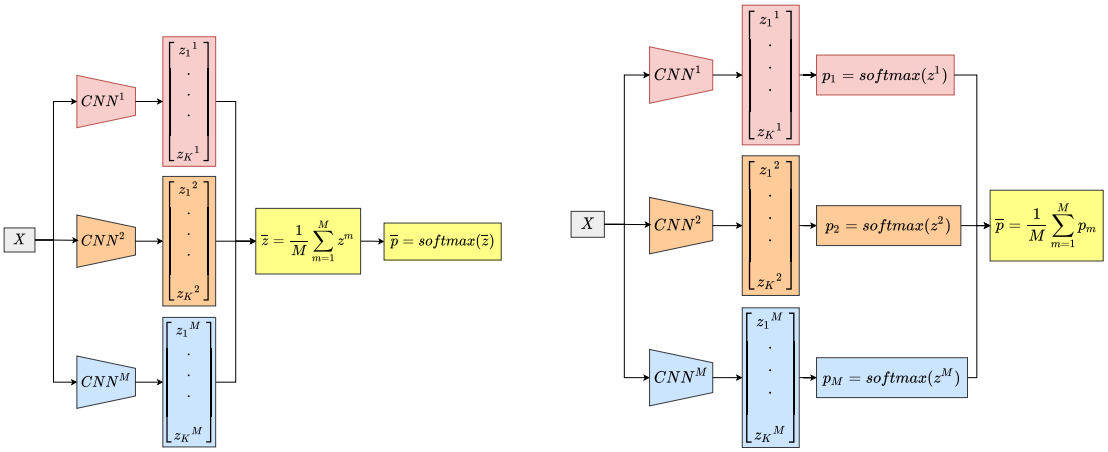
MMCD was investigated in [29, 30, 31] for uncertainty estimation. It combines both MCD and ensemble. For prediction estimation, MCD evaluates a single feature representation, but additionally considers the uncertainty associated with the feature representation. However, an ensemble evaluates multiple feature representations without considering the uncertainty associated with individual feature representations. Hence, MMCD applies MCD to an ensemble to evaluate multiple feature representations and consider the uncertainty associated with individual feature representations. Given a set of CNNs f_{θ_m} for $m \in 1, 2, ..., M$, the MMCD prediction is obtained by averaging over the predictions of all stochastic CNNs. That is,

$$\begin{aligned}\overline{p}(y|x, D_{train}) &\approx \frac{1}{M\cdot S}\sum_{m=1}^{M}\sum_{s=1}^{S} p(y|x, \theta_{m_s}) \\ &\approx \frac{1}{M\cdot S}\sum_{m=1}^{M}\sum_{s=1}^{S} f_{\theta_{m_s}}(x).\end{aligned} \quad (4)$$

In this work, we can refer to MMCD as an average of $M \cdot S$ stochastic CNNs.

4. Combining logits instead of probabilities

The output layer of a CNN-based classifier includes K output neurons with a softmax activation function,



(a) Logit averaging

(b) Probability averaging

Figure 1: Example showing the difference between averaging logits and averaging probabilities in an ensemble.

which normalizes its inputs (continuous values) to produce discrete probabilities p_k (with $k = 1, \ldots, K$ and $\sum_{k=1}^{K} p_k = 1$) representing the probability that the input image belongs to the class associated with the k^{th} output neuron. The input to the softmax function are logits and interpreted as evidence for possible classes [15]. The discrete probability p_k is interpreted as the model confidence that the input belongs to the class associated with the k^{th} output neuron. Given the logit vector $z = \begin{bmatrix} z_1 & \ldots & z_K \end{bmatrix}^T$, the softmax estimates $p = \begin{bmatrix} p_1 & \ldots & p_K \end{bmatrix}^T$ as

$$\begin{aligned} p &= softmax(z) \\ &= \frac{1}{\sum_{k=1}^{K} \exp(z_k)} \begin{bmatrix} \exp(z_1) & \ldots & \exp(z_K) \end{bmatrix}^T . \end{aligned} \tag{5}$$

From Figure 1, given an ensemble of M deterministic CNNs with logits z^m, the average logit $\overline{z}$ can be estimated as

$$\overline{z} := \frac{1}{M} \sum_{m=1}^{M} z^m := \frac{1}{M} \sum_{m=1}^{M} f_{\theta_m}(x). \tag{6}$$

and the predicted probability vector of the ensemble of deterministic CNNs can be reformulated as

$$\overline{p}(y|x, D_{train}) = softmax(\overline{z}). \tag{7}$$

Given MCD representing an ensemble of S stochastic CNNs with logits z^s, we can estimate the average logit $\overline{z}$ as

$$\overline{z} \approx \frac{1}{S} \sum_{s=1}^{S} z^s \approx \frac{1}{S} \sum_{s=1}^{S} f_{\theta_s}(x), \tag{8}$$

and reformulate the predicted probability vector of MCD, as shown in (7). Similarly, given MMCD representing an ensemble of $M \cdot S$ stochastic CNNs with logits z^{m_s}, we can estimate the average logit $\overline{z}$ as

$$\overline{z} \approx \frac{1}{M \cdot S} \sum_{m=1}^{M} \sum_{s=1}^{S} z^{m_s} \approx \frac{1}{M \cdot S} \sum_{m=1}^{M} \sum_{s=1}^{S} f_{\theta_{m_s}}(x), \tag{9}$$

and reformulate the predicted probability vector of MMCD, as shown in (7). From Figure 2, averaging logits instead of probabilities of multiple stochastic or deterministic CNNs increases the confidence of the averaged CNNs. Intuitively, logit averaging provides the best evidence (characterized by a low level of uncertainty caused by the reduction of inductive biases inherent in individual logits) for making decisions. However, probability averaging provides the best confidence associated with decisions made using weak evidence (characterized by a high level of uncertainty caused by inductive biases inherent in individual logits). This implies that a decision made using probability averaging considers more uncertainty than that made using logit averaging. In this work, we evaluated the impact of the possible increase in the degree of confidence caused by applying logit averaging instead of probability averaging on the QoC.

5. Experiments

5.1. Experimental setup

We hypothesized that the QoC of CNNs (strongly) depends on the task-difficulty (specified using the training data), the underlying architecture, and/or the training procedure (mostly influenced by the regularization

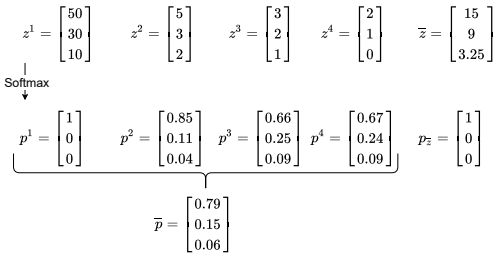


Figure 2: Example showing how averaging logits instead of probabilities increases the confidence of an ensemble of four deterministic CNNs: $\overline{z} = \frac{1}{4}\sum_{m=1}^{4} z^m$, $p_{\overline{z}} = softmax(\overline{z})$, and $\overline{p} = \frac{1}{4}\sum_{m=1}^{4} p^m$ with $p^1 = softmax(z^1)$, $p^2 = softmax(z^2)$, $p^3 = softmax(z^3)$, and $p^4 = softmax(z^4)$. One can see that averaging logits ($p_{\overline{z}}$) results in more confident predictions than averaging probabilities ($\overline{p}$). This is attributed to averaging logits being more sensitive to the magnitude of logit values than averaging probabilities. Here, z^m with large values contributes most to $\overline{z}$. In our example, $\overline{z}$ is mostly influenced by the values of z^1. That is, the contributions of z^2, z^3, and z^4 to $\overline{z}$ are minor. However, $\overline{p}$ is influenced by the values of all probability vectors p^m and therefore, is less sensitive to the magnitude of individual logits.

strength). Therefore, we compared logits and probabilities averaging on three datasets to evaluate the impact of the task-difficulty on the QoC. Moreover, we compared logits and probabilities averaging using three different architectures to evaluate the impact of the underlying architecture on the QoC. Specifically, we evaluated MNIST [32] on VGGNets [1], FashionMNIST [33] on ResNets [2] and CIFAR10 [34] on DenseNets [35]. Finally, we compared logits and probabilities averaging on CNNs trained using two regularization strengths (strong and weak regularization summarized in Table 1) to evaluate the impact of the regularization strength on the QoC. We observed strong and weak regularization results in underconfident and overconfident CNNs, respectively. All CNNs were regularized using batch normalization [36] layers placed before each convolutional activation function. All CNNs were randomly initialized and trained with random shuffling of training samples. All CNNs were trained using the categorical cross-entropy and stochastic gradient descent with momentum of 0.9, learning rate of 0.02, batch size of 128, and epochs of 100. All images were standardized and normalized by dividing pixel values by 255. For all MCD and MMCD, we sampled activations of the first fully-connected layer using masks drawn from a cascade of Bernoulli and Gaussian distributions [22] and using a dropout probability of 0.5. We performed 100 stochastic forward passes ($S = 100$) and considered ensembles consisting of five deterministic CNNs ($M = 5$).

Table 1
Summary of values assigned to regularization hyper-parameters.

Hyper-parameters	Values (weak regularization)	Values (strong regularization)
Probability of dropout applied at inputs to max pooling layers	-	0.05
Probability of dropout applied at inputs to fully-connected layers	0.05	0.5
Rotation range [Degree]	[-5, +5]	[-45, +45]
Width and height shift range [Pixel]	[-1, +1]	[-5, +5]
Scale intensity range	[0.95, 1.05]	[0.9, 1.2]
Shear intensity range	0.05	0.1
Additive Gaussian noise standard deviation range	0.05	0.5

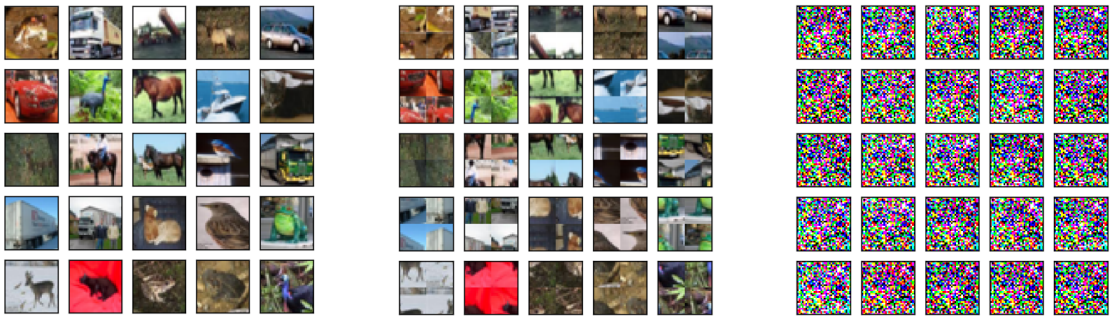
5.2. Evaluation metrics

QoC was evaluated by assessing the degree of confidence calibration. Specifically, we evaluated the calibration error using measures, such as the negative log likelihood (NLL) applied in [4, 5, 31], expected calibration error (ECE) applied in [13, 8, 12], and Brier score (BS) applied in [4]. Low values of NLL, ECE, and BS indicate low calibration error and vice versa. Furthermore, we evaluated QoC by assessing its ability to separate TPs and FPs. Here, we evaluated the average confidence on evaluation data causing TPs or FPs. Given evaluation data causing TPs, we expect the average confidence on the evaluation data to be high. However, for the evaluation data causing FPs, we expect low average confidence on the evaluation data. Moreover, we evaluated the ability to separate TPs and FPs by evaluating the area under the receiver operator characteristic (AU-ROC) applied in [37, 5]. AU-ROC summarizes the trade-off between the fraction of TPs that are correctly detected and those of FPs that are undetected using different thresholds. In summary, in addition to the NLL, ECE, and BS, we evaluated the accuracy, average confidence, and AUC-ROC.

5.3. Evaluation data

We used five evaluation data for different purposes, namely *test data*, *subsets of the correctly classified test data*, *out-of-domain data*, *swapped data*, and *noisy data*.

Test data represent the test data from the experimental data, namely, MNIST, CIFAR10, and FashionMNIST. These datasets include both correctly classified and misclassified test data. Test data are used for estimating the accuracy, NLL, ECE, and



(a) Test data (b) Swapped data (c) Noisy data

Figure 3: Examples of evaluation data for experiments conducted on CIFAR10.

BS. We expect the accuracy to be high and NLL, ECE and BS to be low on test data.

Subsets of the correctly classified test data include 1000 correctly classified test data from the experimental data. Since CNNs will make TPs on these data, we used these data for evaluating the average confidence on TPs.

Swapped data were simulated using subsets of the correctly classified test data structurally perturbed by dividing images into four regions and diagonally permuting the regions. From Figure 3b, the upper left and right are permuted with the bottom right and left regions, respectively. Swapped data include structurally perturbed objects within the given images. We expect CNNs to make FPs on swapped data. Therefore, we used these data for evaluating the average confidence on FPs caused by structurally perturbed objects.

Noisy data were simulated using subsets of the correctly classified test data perturbed by applying additive Gaussian noise with a standard deviation of 500. From Figure 3c, noisy data include noise within the given images. We expect CNNs to make FPs on these data. Therefore, we used these data for evaluating the average confidence on FPs caused by noisy objects.

Out-of-domain data were simulated using 1000 test data of CIFAR100 [34]. Since CNNs will make FPs on these data, we used these data for evaluating the average confidence on FPs caused by unknown objects.

In general, we expect the average confidence to be high on TPs and to be low on FPs.

5.4. Experimental results

We evaluate the conducted experiments with respect to accuracy and QoC.

Table 2 and Table 3 summarize the accuracy, average confidence, NLL, ECE, and BS of different models using the two averaging approaches and CNNs trained using strong regularization (causing underconfidence) and weak regularization (causing overconfidence). The results show that averaging logits instead of probabilities do not strongly affect the accuracy. This means that *averaging logits can preserve accuracy.* Furthermore, averaging logits instead of probabilities significantly increases the average confidence. Figure 2 illustrates why the confidence increases. Further, Table 2 shows that averaging logits instead of probabilities significantly decreases the NLL, ECE, and BS for undercondifent CNNs (trained using strong regularization). This means that *averaging logits, unlike averaging probabilities, reduces the calibration error for undercondifent CNNs.* This is because the stronger the regularization, the lower the confidence and the higher the gap between accuracy and average confidence. Here, the increase in the degree of confidence caused by averaging logits instead of probabilities reduces the gap between accuracy and average confidence. For example, Table 2 shows that averaging logits instead of probabilities of the ensemble reduces the gap between accuracy and average confidence from $18.24(= |88.75 - 70.51|)\%$ to $9.52(= |88.94 - 79.42|)\%$ on CIFAR10.

However, *the increase in the degree of confidence caused by averaging logits instead of probabilities increases the calibration error for overconfident CNNs* (trained using weak regularization). Table 3 provides empirical evidence for this claim by showing that, on CIFAR10 and FashionMNIST, NLL, ECE, and BS of the ensembles increase when the logits are averaged instead of probabilities. We argued that the more overconfident the CNNs, the higher

Table 2
Comparison of accuracy[%], average confidence[%] (in bracket), NLL[10^{-2}], ECE[10^{-2}], and BS[10^{-2}] of different models using two approaches for averaging underconfident CNNs trained using strong regularization: average probabilities (AP) and average logits (AL). The results were obtained using the test data described in Section 5.3.

	Accuracy (Average confidence)↑		**NLL ↓**		**ECE ↓**		**BS ↓**	
	AP	AL	AP	AL	AP	AL	AP	AL
CIFAR10 (DenseNets)								
Ensemble	89.52 (84.31)	89.60 (**87.97**)	34.66	**32.81**	5.23	**2.47**	16.13	**15.38**
MCD	85.36 (73.35)	85.37 (**80.04**)	52.13	**46.55**	12.04	**5.40**	23.32	**21.57**
MMCD	88.75 (70.51)	88.94 (**79.42**)	50.83	**40.99**	18.24	**9.55**	21.82	**18.34**
FashionMNIST (ResNets)								
Ensemble	92.70 (87.86)	92.58 (**90.16**)	22.57	**20.99**	5.15	**2.86**	11.37	**10.87**
MCD	90.56 (79.22)	90.56 (**83.95**)	35.45	**30.18**	11.47	**6.85**	15.82	**14.57**
MMCD	92.65 (76.37)	92.73 (**83.78**)	35.57	**26.96**	16.31	**9.10**	14.87	**12.47**
MNIST (VGGNets)								
Ensemble	99.04 (98.24)	99.04 (**98.89**)	3.25	**2.90**	1.03	**0.52**	1.52	**1.41**
MCD	98.16 (94.53)	98.16 (**96.48**)	8.73	**6.87**	3.81	**1.98**	2.99	**2.79**
MMCD	99.03 (94.67)	99.04 (**97.46**)	6.91	**4.13**	4.49	**1.75**	1.89	**1.52**

Table 3
Comparison of accuracy[%], average confidence[%] (in bracket), NLL[10^{-2}], ECE[10^{-2}], and BS[10^{-2}] of ensembles using two approaches for averaging overconfident CNNs trained using weak regularization: average probabilities (AP) and average logits (AL). The results were obtained using the test data described in Section 5.3.

	Accuracy (Average confidence) ↑		**NLL ↓**		**ECE ↓**		**BS ↓**	
	AP	AL	AP	AL	AP	AL	AP	AL
CIFAR10 (DenseNets)	88.67 (89.43)	88.88 (**96.17**)	**40.69**	54.23	**3.03**	7.40	**16.69**	18.07
FashionMNIST (ResNets)	94.49 (95.86)	94.58 (**98.43**)	**20.20**	28.00	**1.98**	4.11	**8.36**	9.32

the confidence and the higher the gap between accuracy and average confidence. Here, the increase in the degree of confidence caused by averaging logits instead of probabilities further increases the gap between the accuracy and average confidence and therefore, increases the calibration error. For example, Table 3 shows that, on CIFAR10, averaging logits of the ensemble increases the gap between the accuracy and average confidence from $0.76(= |88.67-89.43|)\%$ to $7.29(= |88.88-96.17|)\%$.

In Table 4, the average confidence on TPs and FPs is shown for underconfident models using both averaging approaches. The results show that *averaging logits instead of probabilities increases the confidence level on TPs and FPs*. The increase in the average confidence is sometimes very large for FPs due to the noisy data. For example, for MMCD evaluated on FashionMNIST, the average confidence on the noisy data increases from 51.31% to 94.58% when averaging logits. This is because noisy data can increase the magnitude of logits and averaging logits is more sensitive to changes in the magnitude of logits than averaging probabilities (see Figure 2). *The increase in the degree of confidence caused by averaging logits can harm the separability of TPs and FPs*. For example, the increase in the average confidence on the noisy data from 51.31% to 94.58% causes the AUC-ROC obtained based on the evaluation of the degree of confidence to decrease from 84.80% to 42.42%.

6. Discussion

The term 'combination process' encompasses how multiple networks are combined and the information type combined. It was found in [23, 24, 22] that simple averaging is more robust and captures uncertainty better than voting approaches. This is because the simple averaging equally weights all predictions, while voting ignores uncertain predictions. In this work, we compared the process of averaging logits instead of probabilities. We empirically showed that averaging logits instead of probabilities increases the confidence while preserving the accuracy for underconfident or overconfident networks. This might be because logit averaging preserves the position of the max element of individual logit vectors, but is more sensitive to the magnitude of logit values than probability averaging. Thus, logit values with a large magnitude contribute the most to the average logit. In this way, the magnitude of logit values induces a non-uniform weighting (for logit averaging), which is lost (for probability averaging). Furthermore, we provided empirical evidence showing that for underconfident networks (trained using strong regularization), the increase in the confidence caused by averaging logits instead of

Table 4
Comparison of average confidence[%] of different models using two approaches (average probabilities (AP) and average logits (AL)) for averaging underconfident networks trained using strong regularization and evaluated on TPs and FPs: TPs were obtained on *subsets of the correctly classified test data*, while FPs were obtained on *swapped*, *noisy* and out-of-domain (OOD) data described in Section 5.3.

	TP ↑		**FP (OOD)** ↓		**FP (Swapped)** ↓		**FP (Noisy)** ↓	
	AP	AL	AP	AL	AP	AL	AP	AL
CIFAR10 (DenseNets)								
Ensemble	93.94	**96.63**	**35.39**	40.08	**51.84**	56.03	**39.42**	58.69
MCD	81.39	**88.45**	**31.61**	33.27	**40.39**	44.69	**44.83**	69.53
MMCD	79.48	**89.53**	**22.81**	23.83	**36.26**	40.67	**28.01**	33.08
FashionMNIST (ResNets)								
Ensemble	88.01	**90.16**	**55.48**	63.21	**59.30**	67.91	**81.39**	99.82
MCD	79.39	**83.76**	**47.08**	50.36	**55.75**	59.29	**41.23**	65.79
MMCD	76.40	**83.76**	**42.76**	49.09	**45.73**	52.70	**51.31**	94.58
MNIST (VGGNets)								
Ensemble	99.09	**99.55**	**57.16**	80.45	**51.96**	62.01	**69.58**	88.84
MCD	95.12	**97.11**	**64.36**	69.17	**58.92**	62.84	**97.95**	99.53
MMCD	95.37	**98.17**	**48.89**	63.56	**43.53**	49.39	**57.17**	78.14

probabilities reduces the calibration error on the test data. This is because the increase in the degree of confidence reduces the gap between accuracy and average confidence. However, the increase in confidence caused by averaging logits instead of probabilities for overconfident networks (trained using weak regularization) increases the calibration error on the test data. This is because the increase in the confidence further increases the gap between the accuracy and average confidence. This finding suggests that for underconfident networks, we can average logits instead of probabilities to reduce the calibration error. However, we should average probabilities instead of logits for overconfident networks to avoid increasing the calibration error. Although the increase in the confidence caused by averaging logits reduces the calibration error on the test data for underconfident networks, we empirically showed that it can harm the separability of TPs and FPs. This is because averaging logits increases the confidence on both TPs and FPs. Therefore, FPs can also be made with high confidence similar to TPs. These findings suggest that reducing the calibration error on the test data and improving the separability of TPs and FPs can be two contradicting goals. Improving one may be at the detriment of the other. Furthermore, for two models A and B, if A is better calibrated than B, then A does not necessarily separate TPs and FPs better than B. This implies that calibration methods may be insufficient for separating TPs and FPs and therefore, ensuring safe decision-making. Additionally, existing methods for confidence calibration may not help in separating TPs and FPs. Subsequently, *future work will evaluate the ability of existing methods for confidence calibration to separate TPs and FPs. We also recommend researchers to evaluate both the calibration error of their proposed method for confidence calibration and the ability of their proposed method to separate TPs and FPs.* Finally, for mission- and safety-critical applications where the separability of TPs and FPs is of paramount importance, we suggest to average probabilities to avoid the negative impact of logits averaging on the ability to separate TPs and FPs.

7. Conclusion

Due to averaging logits instead of averaging probabilities of stochastic or deterministic networks, the degree of confidence on TPs and FPs increased. This reduces the calibration error on the test data for underconfident networks but affects the separability of TPs and FPs. Our empirical results show that there is a trade-off between improving calibration on the test data and improving the separability of TPs and FPs. Additionally, the increase in the degree of confidence increases the calibration error on the test data for overconfident networks. Therefore, averaging logits should only be applied when combining underconfident networks. For example, we can average logits instead of probabilities of an ensemble of networks trained with mixup or other modern data augmentation techniques to improve calibration on the test data. Notwithstanding this, for mission- and safety-critical applications where the separability of TPs and FPs is essential, we suggest traditionally average probabilities. However, it remains unclear if the findings of this paper will change if the given networks or the average logit are calibrated, for example, with temperature scaling [13]. This suggests a new research direction.

References

[1] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015. URL: http://arxiv.org/abs/1409.1556.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., A survey of uncertainty in deep neural networks, arXiv preprint arXiv:2107.03342 (2021).

[4] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Advances in neural information processing systems 30 (2017).

[5] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, S. Michalak, On mixup training: Improved calibration and predictive uncertainty for deep neural networks, Advances in Neural Information Processing Systems 32 (2019).

[6] Y. Qin, X. Wang, A. Beutel, E. Chi, Improving calibration through the relationship with adversarial robustness, in: A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, 2021. URL: https://openreview.net/forum?id=NJex-5TZIQa.

[7] Y. Wen, G. Jerfel, R. Muller, M. W. Dusenberry, J. Snoek, B. Lakshminarayanan, D. Tran, Combining ensembles and data augmentation can harm your calibration, arXiv preprint arXiv:2010.09875 (2020).

[8] R. Rahaman, A. H. Thiery, Uncertainty quantification and deep ensembles, Advances in Neural Information Processing Systems 34 (2021).

[9] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[11] R. Müller, S. Kornblith, G. Hinton, When Does Label Smoothing Help?, Curran Associates Inc., Red Hook, NY, USA, 2019.

[12] X. Wu, M. Gales, Should ensemble members be calibrated?, arXiv preprint arXiv:2101.05397 (2021).

[13] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1321–1330.

[14] Z. Zhang, A. V. Dalca, M. R. Sabuncu, Confidence calibration for convolutional neural networks using structured dropout, arXiv preprint arXiv:1906.09551 (2019).

[15] M. Sensoy, L. Kaplan, M. Kandemir, Evidential deep learning to quantify classification uncertainty, Advances in Neural Information Processing Systems 31 (2018).

[16] C. Ju, A. Bibaut, M. van der Laan, The relative performance of ensemble methods with deep convolutional neural networks for image classification, Journal of Applied Statistics 45 (2018) 2800–2818.

[17] L. I. Kuncheva, Combining pattern classifiers: methods and algorithms, John Wiley & Sons, 2014.

[18] M. Van Erp, L. Vuurpijl, L. Schomaker, An overview and comparison of voting methods for pattern recognition, in: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, IEEE, 2002, pp. 195–200.

[19] T. Tajti, New voting functions for neural network algorithms, in: Annales Mathematicae et Informaticae, volume 52, Eszterházy Károly Egyetem Líceum Kiadó, 2020, pp. 229–242.

[20] T. G. Dietterich, Machine-learning research, AI magazine 18 (1997) 97–97.

[21] S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, Review of classifier combination methods, Machine learning in document analysis and recognition (2008) 361–386.

[22] N. Tassi, C. Rovile, Bayesian convolutional neural network: Robustly quantify uncertainty for misclassifications detection, in: Mediterranean Conference on Pattern Recognition and Artificial Intelligence, Springer, 2019, pp. 118–132.

[23] R. T. Clemen, Combining forecasts: A review and annotated bibliography, International journal of forecasting 5 (1989) 559–583.

[24] J. Kittler, M. Hatef, R. P. Duin, J. Matas, On combining classifiers, IEEE transactions on pattern analysis and machine intelligence 20 (1998) 226–239.

[25] K. C. Lichtendahl Jr, Y. Grushka-Cockayne, R. L. Winkler, Is it better to average probabilities or quantiles?, Management Science 59 (2013) 1594–1611.

[26] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR, 2016, pp. 1050–1059.

[27] W. H. Beluch, T. Genewein, A. Nürnberger, J. M. Köhler, The power of ensembles for active learning in image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9368–9377.

[28] F. K. Gustafsson, M. Danelljan, T. B. Schon, Evalu-

ating scalable bayesian deep learning methods for robust computer vision, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 318–319.
[29] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, S. Levine, Uncertainty-aware reinforcement learning for collision avoidance, arXiv preprint arXiv:1702.01182 (2017).
[30] B. Lütjens, M. Everett, J. P. How, Safe reinforcement learning with model uncertainty estimates, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8662–8668.
[31] A. G. Wilson, P. Izmailov, Bayesian deep learning and a probabilistic perspective of generalization, Advances in neural information processing systems 33 (2020) 4697–4708.
[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.
[33] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).
[34] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[35] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[36] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, PMLR, 2015, pp. 448–456.
[37] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, Proceedings of International Conference on Learning Representations (2017).