Well-calibrated Model Uncertainty with Temperature Scaling for Dropout Variational Inference

Max-Heinrich Laves **Sontje Ihler** **Karl-Philipp Kortmann** **Tobias Ortmaier**

Institute of Mechatronic Systems
Leibniz University Hannover, Germany
{laves,ihler,kortmann,ortmaier}@imes.uni-hannover.de

Abstract

Model uncertainty obtained by variational Bayesian inference with Monte Carlo dropout is prone to miscalibration. The uncertainty does not represent the model error well. In this paper, temperature scaling is extended to dropout variational inference to calibrate model uncertainty. Expected uncertainty calibration error (UCE) is presented as a metric to measure miscalibration of uncertainty. The effectiveness of this approach is evaluated on CIFAR-10/100 for recent CNN architectures. Experimental results show, that temperature scaling considerably reduces miscalibration by means of UCE and enables robust rejection of uncertain predictions. The proposed approach can easily be derived from frequentist temperature scaling and yields well-calibrated model uncertainty. It is simple to implement and does not affect the model accuracy.

1 Introduction

For safety-critical vision tasks such as autonomous driving or computer-aided diagnosis, it is essential to consider the prediction *uncertainty* of deep learning models. Bayesian neural networks and recent advances in their approximation provide the mathematical tools for quantification of uncertainty [1, 2]. One practical approximation is variational inference with Monte Carlo (MC) dropout [3]. It is applied to obtain epistemic uncertainty, which is caused by uncertainty in the model weights due to training with data sets of limited size [1, 4]. However, it tends to be miscalibrated, i. e. the uncertainty does not correspond well to the model error [5, 6].

First, we consider the problem of miscalibration of the frequentist approach: The weights of a deep model are obtained by maximum likelihood estimation [1], and the normalized output likelihood for an unseen test input does not consider uncertainty in the weights [4]. The likelihood is generally unjustifiably high [5], and can be misinterpreted as high prediction *confidence*. This miscalibration can also be observed for model uncertainty provided by MC dropout variational inference. However, calibrated uncertainty is essential as miscalibration can lead to decisions with fatal consequences in the aforementioned task domains.

Overconfident predictions of neural networks have been addressed by entropy regularization techniques. Szegedy et al. present label smoothing as regularization of models during supervised training for classification [7]. They state that a model trained with one-hot encoded labels is prone to becoming overconfident about its predictions, which causes overfitting and poor generalization. Pereyra et al. link label smoothing to *confidence penalty* (CP) and propose a simple way to prevent overconfident networks [8]. Low entropy output distributions are penalized by adding the negative entropy to the training objective. However, the referred works do not apply entropy regularization to the calibration of confidence or uncertainty. In the last decades, several non-parametric and parametric

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

calibration approaches such as isotonic regression [9] or Platt scaling [10] have been presented. Recently, *temperature scaling* (TS) has been demonstrated to lead to well-calibrated model likelihood in non-Bayesian deep neural networks [5]. It uses a single scalar to smooth the softmax output and regularize the entropy. Scaling has also been introduced to approximate categorical distributions by the Gumbel-Softmax or Concrete distribution [11, 12].

Our work extends temperature scaling to variational Bayesian inference with dropout to obtain well-calibrated model uncertainty. The main contributions of this paper are 1. definition for perfect calibration of uncertainty and definition for the expected uncertainty calibration error, 2. the derivation of temperature scaling for dropout variational inference, and 3. experimental results of different network architectures on CIFAR-10/100 [13], that demonstrate the improvement of calibration by the proposed method and superiority over confidence penalty. By using temperature scaling together with Bayesian inference, we expect better calibrated uncertainty. To the best of our knowledge, temperature scaling has not yet been used to calibrate model uncertainty in variational Bayesian inference. Our code is available at: `https://github.com/mlaves/bayesian-temperature-scaling`.

2 Methods

The presented approach for obtaining well-calibrated uncertainty is applied to a general multi-class classification task. Let input $\mathbf{x} \in \mathcal{X}$ be a random variable with corresponding label $y \in \mathcal{Y} = \{1, \dots, C\}$. Let $\mathbf{f_w}(\mathbf{x})$ be the output (logits) of a neural network with weight matrices $\mathbf{w}$, and with model likelihood $p(y = c \,|\, \mathbf{f_w}(\mathbf{x}))$ for class c, which is sampled from a probability vector $\mathbf{p} = \boldsymbol{\sigma}_{\mathrm{SM}}(\mathbf{f_w}(\mathbf{x}))$, obtained by passing the model output through the softmax function $\boldsymbol{\sigma}_{\mathrm{SM}}(\cdot)$. From a frequentist perspective, the softmax likelihood is often interpreted as confidence of prediction. Throughout this paper, we follow this definition. However, due to optimizing the weights $\mathbf{w}$ via minimization of the negative log-likelihood of $p(y \,|\, \mathbf{f_w}(\mathbf{x}))$, modern deep models are prone to overly confident predictions and are therefore miscalibrated [5, 6].

Let $\hat{y} = \arg\max \mathbf{p}$ be the most likely class prediction of input $\mathbf{x}$ with likelihood $\hat{p} = \max \mathbf{p}$ and true label y. Then, following Guo et al. [5], *perfect calibration* is defined as

$$\mathbb{P}\left(\hat{y} = y \,|\, \hat{p} = q\right) = q, \quad \forall q \in [0, 1]\,. \tag{1}$$

To determine model uncertainty, dropout variational inference is performed by training the model $\mathbf{f_w}$ with dropout [14] and using dropout at test time to sample from the approximate posterior by performing N stochastic forward passes [3, 4]. This is also referred to as MC dropout. In MC dropout, the final probability vector is obtained by MC integration:

$$\mathbf{p}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\sigma}_{\mathrm{SM}}\left(\mathbf{f}_{\mathbf{w}_i}(\mathbf{x})\right). \tag{2}$$

Entropy of the softmax likelihood is used to describe *uncertainty* of prediction [4]. In contrast to confidence as a measure of goodness of prediction, uncertainty takes into account the likelihoods of all C classes. We introduce normalization to scale the values to a range between 0 and 1:

$$\tilde{\mathcal{H}}(\mathbf{p}) := -\frac{1}{\log C} \sum_{c=1}^{C} p^{(c)} \log p^{(c)}\,, \quad \tilde{\mathcal{H}} \in [0, 1]\,. \tag{3}$$

From Eq. (1) and Eq. (3), we define *perfect calibration of uncertainty* as

$$\mathbb{P}(\hat{y} \neq y \,|\, \tilde{\mathcal{H}}(\mathbf{p}) = q) = q, \quad \forall q \in [0, 1]\,. \tag{4}$$

That is, in a batch of inputs all predicted with uncertainty of e.g. 0.2, a top-1 error of $20\,\%$ is expected.

2.1 Expected Uncertainty Calibration Error (UCE)

A popular way to quantify miscalibration of neural networks with a scalar value is the expectation of the difference between predicted softmax likelihood $\hat{p}$ and accuracy

$$\mathbb{E}_{\hat{p}}\left[\,\left|\mathbb{P}\left(\hat{y} = y \,|\, \hat{p} = q\right) - q\right|\,\right], \quad \forall q \in [0, 1]\,, \tag{5}$$

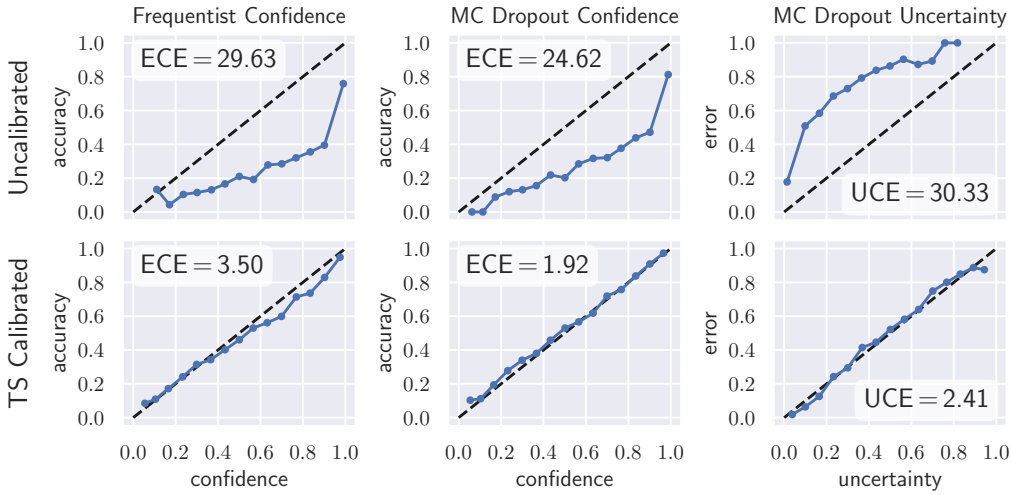

Figure 1: Reliability diagrams ($M = 15$ bins) for ResNet-101 on CIFAR-100. Top row: Uncalibrated frequentist confidence (left), and confidence and uncertainty obtained by dropout variational inference (right). Bottom row: Results from calibration with TS. Dashed lines denote perfect calibration.

which can be approximated by the Expected Calibration Error (ECE) [15, 5]. The output of a neural network is partitioned into M bins with equal width and a weighted average of the difference between accuracy and confidence (softmax likelihood) is taken:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| , \tag{6}$$

with total number of inputs n and set of indices B_m of inputs whose confidence falls into that bin (see [5] for more details). We propose the following slightly modified notion of Eq. (5) to quantify miscalibration of uncertainty:

$$\mathbb{E}_{\tilde{\mathcal{H}}}[\,|\mathbb{P}(\hat{y} \neq y \,|\, \tilde{\mathcal{H}}(\mathbf{p}) = q) - q|\,], \quad \forall q \in [0, 1]\,. \tag{7}$$

We refer to this as Expected Uncertainty Calibration Error (UCE) and approximate analogously with

$$\mathrm{UCE} := \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{err}(B_m) - \mathrm{uncert}(B_m)\right| . \tag{8}$$

See § A.1 for definitions of $\mathrm{err}(B_m)$ and $\mathrm{uncert}(B_m)$.

2.2 Temperature Scaling for Dropout Variational Inference

State-of-the-art deep neural networks are generally miscalibrated with regard to softmax likelihood [5]. However, when obtaining model uncertainty with dropout variational inference, this also tends to be not well-calibrated [6]. Fig. 1 (top row) shows reliability diagrams [16] for uncalibrated ResNet-101 [17] trained on CIFAR-100 [13]. The divergence from the identity function reveals miscalibration.

In this work, dropout is inserted before the last layer with fixed dropout probability of 0.5 as in [3]. Temperature scaling with $T > 0$ is inserted before final softmax activation and before MC integration:

$$\hat{\mathbf{p}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\sigma}_{\mathrm{SM}} \left(T^{-1} \mathbf{f}_{\mathbf{w}_i}(\mathbf{x})\right) . \tag{9}$$

T is optimized with respect to negative log-likelihood while performing MC dropout on the validation set. This is equivalent to maximizing the entropy of $\hat{\mathbf{p}}$ [5]. See § A.2 for more details on T.

Table 1: ECE and UCE test set results in % ($M = 15$ bins). 0 % means perfect calibration. In TS calibration with MC dropout the same value of T was used to report both ECE and UCE.

		Uncalibrated			TS Calibrated		
		Freq.	MC Dropout		Freq.	MC Dropout	
Data Set	Model	ECE	ECE	UCE	ECE	ECE	UCE
CIFAR-10	ResNet-18	8.95	8.41	7.60	1.40	**0.47**	**5.27**
CIFAR-100	ResNet-101	29.63	24.62	30.33	3.50	**1.92**	**2.41**
CIFAR-100	DenseNet-169	30.62	23.98	29.62	6.10	**2.89**	**2.69**

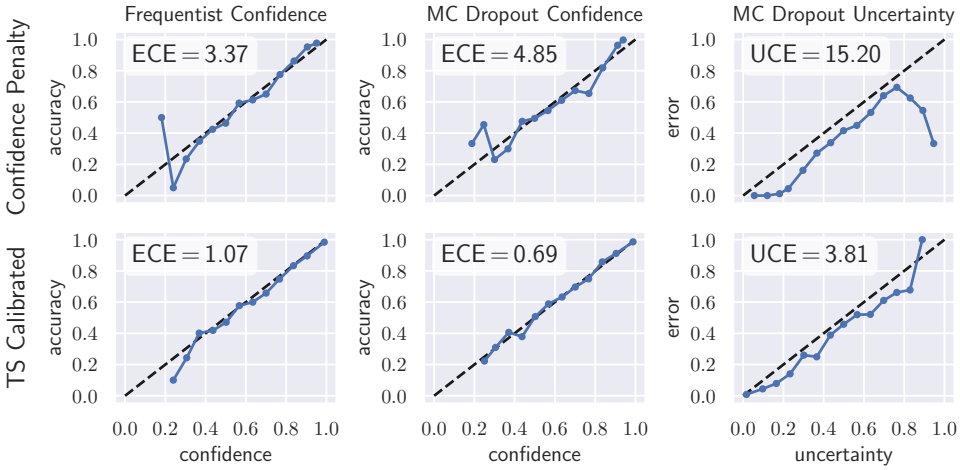

Figure 2: Reliability diagrams ($M = 15$ bins) for DenseNet-121 on CIFAR-10. Top row: Training with confidence penalty. Bottom row: TS calibrated (trained without confidence penalty).

3 Experiments & Results

The experimental results of the proposed method are presented threefold: First, TS is used to calibrate confidence and uncertainty obtained by MC dropout; second, TS calibration is compared with calibration by entropy regularization using confidence penalty; and finally, uncertain predictions are rejected based on well-calibrated uncertainty. Details on the training procedure can be found in § A.3.

3.1 Uncertainty Calibration

Tab. 1 reports test set results for different networks [17, 18] and data sets used to evaluate the performance of temperature scaling for dropout variational inference. The proposed UCE metric is used to quantify calibration of uncertainty. Fig. 1 shows reliability diagrams [16] for different calibration scenarios of ResNet-101 [17] on CIFAR-100. For MC dropout $N = 25$ forward passes are performed. Uncalibrated ECE shows, that MC dropout already reduces miscalibration of model likelihood by up to 6.6 percentage points. With TS calibration, MC dropout reduces ECE by 45–66 % and UCE drops drastically (especially for larger networks). This illustrates the magnitude of how much TS calibration benefits from Bayesian inference using MC dropout. Additional reliability diagrams showing similar results can be found in § A.4.

3.2 Temperature Scaling vs. Confidence Penalty

Low entropy output distributions are penalized by adding the negative entropy $\mathcal{H}$ of the softmax output to the negative log-likelihood training objective, weighted by an additional hyperparameter β. This leads to the following optimization function:

$$\mathcal{L}_{\text{CP}}(\mathbf{w}) = -\sum_{\mathcal{X},\mathcal{Y}} \log \mathbf{p}_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) - \beta\, \mathcal{H}\left(\mathbf{p}_{\mathbf{w}}(\mathbf{y}|\mathbf{x})\right) \ . \tag{10}$$

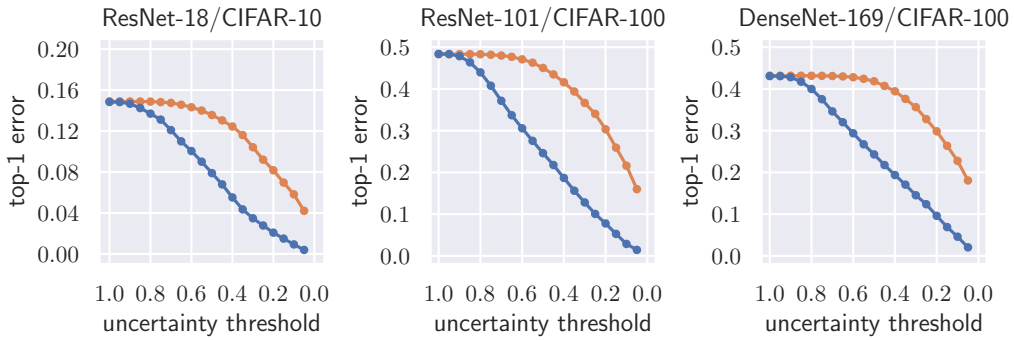

Figure 3: The effect of the uncertainty threshold $\mathcal{H}_{\max}$ on the test set error for the rejection of uncertain predictions (orange: uncalibrated, blue: TS calibrated). As $\mathcal{H}_{\max}$ decreases, more uncertain predictions are rejected, which results in a lower error.

We reproduce the experiment of Pereyra et al. on supervised image classification [8] and compare the goodness of calibration of confidence and uncertainty to our presented approach. DenseNet-121 with dropout is trained on CIFAR-10 as described in § A.3. We fix $\beta = 0.1$ for CP loss and omit data augmentation for this experiment as presented in [8].

Fig. 2 compares training with confidence penalty to our approach. CP reduces miscalibration ($\mathrm{ECE} = 5.20\,\%$ without CP vs. $\mathrm{ECE} = 3.37\,\%$ with CP for DenseNet-121). However, it is not as effective as TS and still produces largely miscalibrated uncertainty. A combination of CP during training and subsequent TS is conceivable and could possibly lead to an even better calibration. We have not followed this approach yet.

3.3 Example: Rejection of Uncertain Predictions

An example application of well-calibrated prediction uncertainty is the rejection of uncertain predictions. We define an uncertainty threshold $\mathcal{H}_{\max}$ and reject all predictions from the test set where $\tilde{\mathcal{H}}(\mathbf{p}) > \mathcal{H}_{\max}$. A decrease in false predictions of the remaining test set is expected. Fig. 3 shows the top-1 error as a function of decreasing $\mathcal{H}_{\max}$. For both uncalibrated and calibrated uncertainty, decreasing $\mathcal{H}_{\max}$ reduces the top-1 error. Using calibrated uncertainty, the relationship is almost linear (for $\mathcal{H}_{\max} < 0.8$), allowing robust rejection of uncertain predictions.

4 Conclusion

Temperature scaling calibrates uncertainty obtained by dropout variational inference with high effectiveness. The experimental results confirm the hypothesis that the presented approach yields better calibrated uncertainty. In addition, substantially better calibrated softmax probability was achieved. MC dropout TS is simple to implement, more effective than confidence penalty during training and the scaling does not change the maximum of the output of a network, thus model accuracy is not compromised. Therefore, it is an obvious choice in Bayesian deep learning with dropout variational inference because well-calibrated uncertainties are of utmost importance for safety-critical decision-making. However, there are many factors (e. g. network architecture, weight decay, dropout configuration) influencing the uncertainty in Bayesian deep learning that have not been discussed in this paper and are open to future work.

Acknowledgments

This work has received funding from European Union EFRE projects *OPhonLas* and *ProMoPro*. We thank the reviewers for their helpful comments.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016.

[4] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, pages 5574–5584, 2017.

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330, 2017.

[6] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *NeurIPS*, pages 3581–3590, 2017.

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

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

[9] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, pages 694–699, 2002.

[10] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *Bayesian Deep Learning Workshop, NeurIPS*, 2016.

[12] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Bayesian Deep Learning Workshop, NeurIPS*, 2016.

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15:1929–1958, 2014.

[15] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *AAAI*, pages 2901–2907, 2015.

[16] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, pages 625–632, 2005.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

A Appendix

A.1 Expected Uncertainty Calibration Error

We restate the definition of Expected Uncertainty Calibration Error (UCE) from Eq. (8):

$$\mathrm{UCE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \big|\mathrm{err}(B_m) - \mathrm{uncert}(B_m)\big| \,.$$

The error per bin is defined as

$$\mathrm{err}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i \neq y) \,, \tag{11}$$

where $\mathbf{1}(\hat{y}_i \neq y) = 1$ and $\mathbf{1}(\hat{y}_i = y) = 0$. Uncertainty per bin is defined as

$$\mathrm{uncert}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \tilde{\mathcal{H}}(\mathbf{p}_i) \,. \tag{12}$$

A.2 Temperature Scaling with Monte Carlo Dropout

Temperature scaling with MC dropout variational inference is derived by closely following the derivation of frequentist temperature scaling in the appendix of [5]. Let $\{\mathbf{z}_{1,j}, \ldots, \mathbf{z}_{N,j}\}$ be a set of logit vectors obtained by MC dropout with N stochastic forward passes for each input $\mathbf{x}_j \in \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ with true labels $\{y_1, \ldots, y_M\}$. Temperature scaling is the solution $\hat{p}$ to entropy maximization

$$\max_{\hat{p}} \; -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{c=1}^{C} \hat{p}\left(\mathbf{z}_{i,j}\right)^{(c)} \log \hat{p}\left(\mathbf{z}_{i,j}\right)^{(c)} \,, \tag{13}$$

subject to

$$\hat{p}(\mathbf{z}_{i,j})^{(c)} \geq 0 \quad \forall i, j, c \,, \tag{14}$$

$$\sum_{c=1}^{C} \hat{p}(\mathbf{z}_j)^{(c)} = 1 \quad \forall j \,, \tag{15}$$

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} z_{i,j}^{(y_j)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{c=1}^{C} z_{i,j}^{(c)} \hat{p}(\mathbf{z}_{i,j})^{(c)}. \tag{16}$$

Guo et al. solve this constrained optimization problem with the method of Lagrange multipliers. We skip reviewing their proof as one can see that the solution to $\hat{p}$ in the case of MC dropout integration provides

$$\frac{1}{N} \sum_{i=1}^{N} \hat{p}_i\left(\mathbf{z}_j\right)^{(c)} = \frac{1}{N} \sum_{i=1}^{N} \frac{e^{\lambda z_{i,j}^{(c)}}}{\sum_{\ell=1}^{C} e^{\lambda z_{i,j}^{(\ell)}}} \tag{17}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\sigma}_{\mathrm{SM}}\left(\lambda \mathbf{f}_{\mathbf{w}_i}(\mathbf{x}_j)\right)^{(c)} \,, \tag{18}$$

which recovers temperature scaling for $\lambda = T^{-1}$ [5]. T is optimized with respect to negative log-likelihood on the validation set using MC dropout.

A.3 Training Settings

The model implementations from PyTorch 1.2 [19] are used and trained with following settings:

- batch size of 256
- AdamW optimizer [20] with initial learn rate of 0.01 and $\beta_1 = 0.9, \beta_2 = 0.999$

- weight decay of 0.01
- negative-log likelihood (cross entropy) loss
- reduce-on-plateau learn rate scheduler (patience of 10 epochs) with factor of 0.1
- additional validation set is randomly extracted from the training set (5000 samples)
- dropout with probability of 0.5 before the last linear layer was used in all models during training
- in MC dropout, $N = 25$ forward passes with dropout probability of 0.5 were performed

Code is available at: `https://github.com/mlaves/bayesian-temperature-scaling`.

A.4 Additional Reliability Diagrams

In this section, reliability diagrams for the other data set/model combinations from Tab. 1 are visualized to provide additional insight into the calibration performance. The proposed method is able to calibrate all models with respect to both UCE and ECE across all bins.

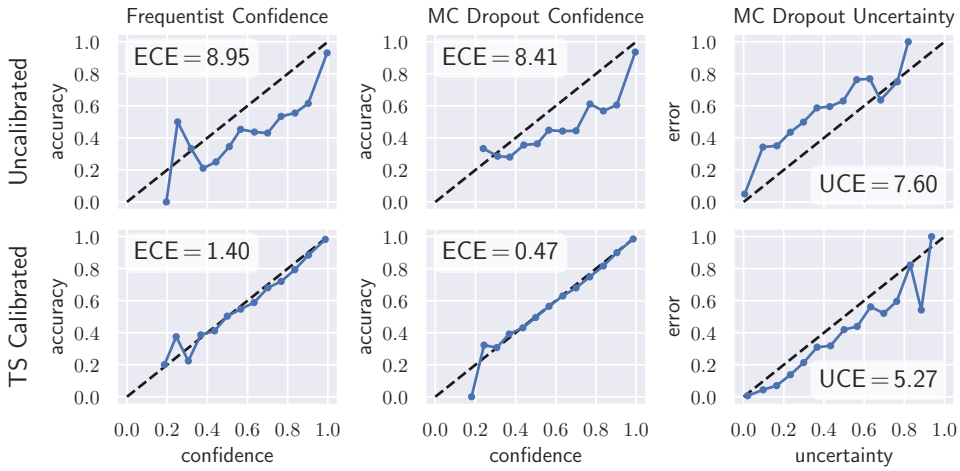

Figure 4: Reliability diagrams ($M = 15$ bins) for ResNet-18 on CIFAR-10.

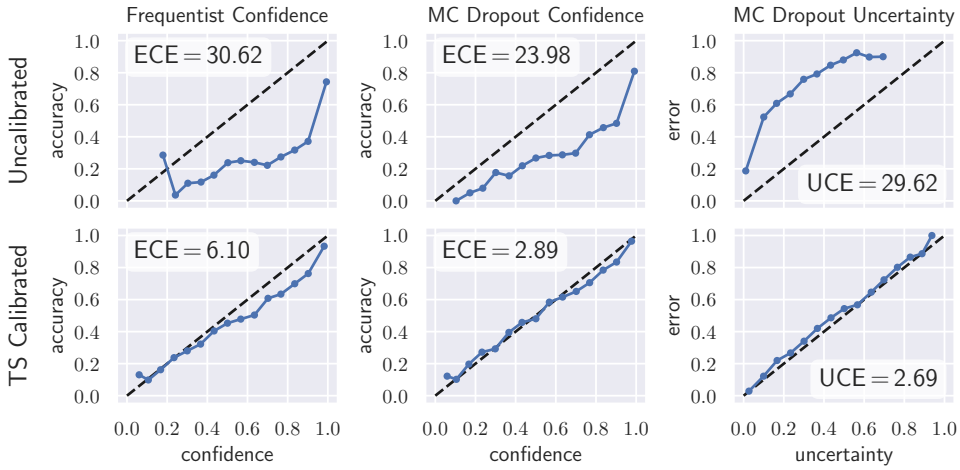

Figure 5: Reliability diagrams ($M = 15$ bins) for DenseNet-169 on CIFAR-100.