Meta-Calibration: Meta-Learning of Model Calibration Using Differentiable Expected Calibration Error

Ondrej Bohdal [1] **Yongxin Yang** [1] **Timothy Hospedales** [1]

Abstract

Calibration of neural networks is a topical problem that is becoming increasingly important for real-world use of neural networks. The problem is especially noticeable when using modern neural networks, for which there is significant difference between the model confidence and the confidence it should have. Various strategies have been successfully proposed, yet there is more space for improvements. We propose a novel approach that introduces a differentiable metric for expected calibration error and successfully uses it as an objective for meta-learning, achieving competitive results with state-of-the-art approaches. Our approach presents a new direction of using meta-learning to directly optimize model calibration, which we believe will inspire further work in this promising and new direction.

1. Introduction

When deploying neural networks to real-world applications, it is crucial that models have accurate estimates of their belief confidence. If a model is over-confident about its predictions, we cannot rely on it. Models that are accurately confident about their predictions can be described as well-calibrated. However, modern neural networks are known to be badly calibrated (Guo et al., 2017), which has motivated research in the area of calibration.

Several calibration approaches have been proposed so far (Guo et al., 2017; Mukhoti et al., 2020; Kumar et al., 2018), but model calibration remains an active area of research where further important advances can be made. Guo et al. (2017) is a foundational work that discovers modern neural networks are typically miscalibrated. Guo et al. (2017) study a variety of potential solutions and find simple post-training rescaling of the logits – temperature scaling – works relatively well. Kumar et al. (2018) propose a kernel-based measure of calibration that they use as regularization during training of neural networks. Mukhoti et al. (2020) show focal loss – a relatively simple weighted alternative to cross-entropy – can be used to train well-calibrated neural networks. The classic Brier score (Brier, 1950), which is the squared error between the softmax vector with probabilities and the ground-truth one-hot encoding, has also been show to work well. Similarly, label smoothing (Müller et al., 2019) has also been shown to improve model calibration.

Calibration is typically measured using expected calibration error (ECE). Prior work seeks indirect ways to optimize ECE, since it is not differentiable. We take a more direct approach by deriving a differentiable approximation to ECE (DECE), which can be optimized directly. Since ECE is based on measuring accuracy of predictions binned by confidence, this requires both a differentiable approximation to $0/1$ accuracy and to the binning operator. Further, since calibration is highly influenced by model hyperparameters such as regularizers, we introduce a novel meta-learning framework for tuning hyperparameters to optimize calibration without altering the underlying model training strategy.

We show we can successfully use DECE-driven meta-learning to obtain well-calibrated and high-accuracy models. In particular, we meta-learn unit-wise L2 regularization on the classifier layer and demonstrate competitive results on CIFAR-10 and CIFAR-100 benchmarks (Krizhevsky, 2009). To summarize our contributions: 1) We introduce a novel differentiable approximation to calibration error. 2) We analyse the proposed metric in detail and show the approximation closely matches the original non-differentiable version. 3) We demonstrate proof-of-concept results that show the measure can be successfully used as part of meta-learning pipeline to optimize hyperparameters for calibration.

2. Methods

2.1. Preliminaries

We first discuss the definition of expected calibration error (ECE) (Naeini et al., 2015), before we derive a differentiable approximation to it. ECE measures the expected difference

[1]School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom. Correspondence to: Ondrej Bohdal <ondrej.bohdal@ed.ac.uk>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

(in absolute value) between the accuracies and the confidences of the model on examples that belong to different confidence intervals. ECE is defined as

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

where accuracy and confidence for bin B_m are

$$\mathrm{acc}\left(B_m\right) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}\left(\hat{y}_i = y_i\right)$$
$$\mathrm{conf}\left(B_m\right) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i.$$

There are M interval bins each of size $1/M$ and n samples. Confidence $\hat{p}_i$ is the probability of the top prediction as given by the model for example i. We group the confidences into their corresponding bins, with bin B_m covering interval $(\frac{m-1}{M}, \frac{m}{M}]$. The predicted class of example i is $\hat{y}_i$, while y_i is the true class of example i and $\mathbf{1}$ is an indicator function.

ECE metric is not differentiable because assigning examples into bins is not differentiable and also accuracy is not differentiable due to the indicator function. We propose approximations to both binning and accuracy and derive a new metric called differentiable ECE (DECE).

2.2. Differentiable ECE

There are three main components in ECE: accuracy, confidence and bins. Only confidence is differentiable.

Differentiable Accuracy: In order to obtain a differentiable approximation to accuracy, we consider approaches that allow us to find a differentiable way to calculate the rank of a given class. Two approaches stand out: differentiable ranking (Blondel et al., 2020) and an all-pairs approach (Qin et al., 2010). While both allow us to approximate the rank in a differentiable way, differentiable ranking can only be done on CPU, which would introduce a potential bottleneck for modern applications. All-pairs approach has asymptotic complexity of $\mathcal{O}(n^2)$ for n classes, while differentiable ranking is $\mathcal{O}(n \log n)$. However, if the number of classes is not in thousands or millions, differentiable ranking would be slower because of not using GPUs. We use the all-pairs approach to estimate the rank of a given class.

All-pairs (Qin et al., 2010) calculates a rank of class i as $\left[R\left(\cdot\right)\right]_i = 1 + \sum_{j \neq i} \mathbf{1}\left[\boldsymbol{\phi}_i < \boldsymbol{\phi}_j\right]$, where $\boldsymbol{\phi}$ are the logits. We can obtain soft ranks by replacing the indicator function with a sigmoid scaled with some temperature value τ_a to obtain reliable estimates of the rank of the top predicted class. Once the rank $[R(\cdot)]_l$ for true class l is calculated, we can estimate the accuracy as $\mathrm{acc} = \max(0, \left[R\left(\cdot\right)\right]_l - L + 1)$, where L is the total number of classes.

Soft Binning: Our approach is similar to Yang et al. (2018). We take confidence $\hat{p}_i$ for example x_i and pass it through one-layer neural network $\mathrm{softmax}((w\hat{p}_i + b)/\tau_b)$ parameterized with different values of w and b as explained in Yang et al. (2018), with temperature τ_b to control the binning. This leads to M different probabilities, saying how likely it is that $\hat{p}_i$ belongs to the specific bin $B_{m \in 1..M}$. We will denote these probabilities as $o_m(x_i) = p(B_m|\hat{p}_i)$.

Putting these parts together, we define DECE using a mini-batch of n examples as:

$$\mathrm{DECE} = \sum_{m=1}^{M} \frac{\sum_{i=1}^{n} o_m(x_i)}{n} \left|\mathrm{acc}\left(B_m\right) - \mathrm{conf}\left(B_m\right)\right|,$$
$$\mathrm{acc}\left(B_m\right) = \frac{1}{\sum_{i=1}^{n} o_m(x_i)} \sum_{i=1}^{n} o_m(x_i) \mathbf{1}\left(\hat{y}_i = y_i\right),$$
$$\mathrm{conf}\left(B_m\right) = \frac{1}{\sum_{i=1}^{n} o_m(x_i)} \sum_{i=1}^{n} o_m(x_i) \hat{p}_i.$$

2.3. Meta-Learning

Differentiable ECE is only one component – it provides an objective to optimize. But we still must decide how to utilize it. One option could be to directly use it as an extra objective in combination with standard cross-entropy. However, given that calibration is largely influenced by choice of hyperparameters, and to avoid disturbing the standard optimization by multi-task learning with new losses, we explore the novel approach of using DECE as an objective for hyperparameter meta-learning in an outer loop while retaining conventional cross-entropy-driven learning for model training.

A key part of meta-learning is to select suitable meta-knowledge (hyperparameters) that we will optimize to achieve the given goal (Hospedales et al., 2021). In our case meta-knowledge will be the L2 regularization coefficients. In particular, we learn L2 regularization coefficients of *each weight* in the classifier layer. This is similar to the approach of Balaji et al. (2018), who used a similar representation for improving domain generalization. We adopt online meta-learning approach (Luketina et al., 2016) where we alternate base model and meta-knowledge (coefficient) updates. This is an efficient strategy as we do not need to backpropagate through many inner-loop steps or retrain the model from scratch for each update of meta-knowledge.

We formulate our approach as a bilevel optimization problem. The goal is to find unit-wise L2 regularization coefficients $\boldsymbol{\omega}$ for the classifier layer $\boldsymbol{\phi}$ so that training with them optimizes validation DECE ($\boldsymbol{\theta}$ is the feature extractor):

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \mathcal{L}_{DECE}^{val}(\boldsymbol{\phi}^* \circ \boldsymbol{\theta}^*(\boldsymbol{\omega})),$$
$$\boldsymbol{\phi}^*, \boldsymbol{\theta}^*(\boldsymbol{\omega}) = \arg\min_{\boldsymbol{\phi}, \boldsymbol{\theta}} \left(\mathcal{L}_{CE}^{train}\left(\boldsymbol{\phi} \circ \boldsymbol{\theta}\right) + \boldsymbol{\omega} \|\boldsymbol{\phi}\|^2\right).$$

When simulating training during the inner loop, we only update the classifier and keep the feature extractor frozen for efficiency (Balaji et al., 2018). Base model training is done separately using a full model update and more advanced optimizer. We give overview of our meta-learning algorithm in Algorithm 1. The inner loop that trains the main model (line 8) is regularized using the learnable L2 regularization, while the outer loop (line 10) that trains the meta-knowledge (learnable L2 regularization) does not directly use it for evaluating DECE. We backpropagate through one step of update of the main model.

Algorithm 1 Meta-Calibration

1: **Input:** α, β: inner and outer-loop learning rates
2: **Output:** trained feature extractor $\boldsymbol{\theta}$, classifier $\boldsymbol{\phi}$ and regularization $\boldsymbol{\omega}$
3: $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$
4: $\boldsymbol{\phi}, \boldsymbol{\theta} \sim p(\boldsymbol{\phi}), p(\boldsymbol{\theta})$
5: **while** $\boldsymbol{\omega}$ not converged **do**
6: Sample minibatch of training x_t, y_t and validation x_v, y_v examples
7: Calculate $\mathcal{L}_i = \mathcal{L}_{CE}\left(f_{\boldsymbol{\phi}\circ\boldsymbol{\theta}}\left(x_t\right), y_t\right) + \boldsymbol{\omega}\|\boldsymbol{\phi}\|^2$
8: Update $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow \boldsymbol{\theta}, \boldsymbol{\phi} - \alpha\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathcal{L}_i$
9: Calculate $\mathcal{L}_o = \mathcal{L}_{DECE}\left(f_{\boldsymbol{\phi}\circ\boldsymbol{\theta}}\left(x_v\right), y_v\right)$
10: Update $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \beta\nabla_{\boldsymbol{\omega}}\mathcal{L}_o$
11: **end while**

3. Experiments

3.1. Experiment Settings

We use ResNet-18 (He et al., 2015) as a model and CIFAR-10 and CIFAR-100 as the benchmarks. We use the implementation from Mukhoti et al. (2020) and use the same hyperparameters. We train the models for 350 epochs, with a multi-step scheduler that decreases the initial learning rate of 0.1 by a factor of 10 after 150 and 250 epochs. The model is trained with SGD with momentum of 0.9, weight decay of 0.0005 and minibatch size of 128. 90% of the original training set is used for training and 10% for validation. In the case of meta-learning, we create a further separate meta-validation set that is of size 10% of the original training data, so we directly train with 80% of the original training data.

For DECE, we use $M = 15$ bins and scaling parameters $\tau_a = 100, \tau_b = 0.01$. Learnable unit-wise L2 regularization for the classifier layer is optimized using Adam (Kingma and Ba, 2015) optimizer with learning rate of 0.001. The meta-learnable parameters are initialized at 0.0 and their number is $512 \times C + C$, where C is the number of classes.

We compare our approach with the following: 1) cross-entropy, 2) Brier score (Brier, 1950), 3) Weighted MMCE (Kumar et al., 2018) with $\lambda = 2$, 4) Focal loss (Lin et al., 2017) with $\gamma = 3$, 5) Adaptive (sample dependent) focal loss (FLSD) (Mukhoti et al., 2020) with $\gamma = 3$, 6) Label smoothing (LS) with smothing factor of 0.05. We also explore temperature scaling using the validation set.

3.2. Results

The results in Table 1 show means and standard deviations of 3 repetitions of each experiment on CIFAR-10 and CIFAR-100. While temperature scaling (TS) benefits all models, the key column is plain ECE since TS is not always possible, e.g. in online or few-shot learning without a validation set (Kim and Yun, 2020).

The results confirm our DECE-driven meta-learning obtains strong calibration results while maintaining comparable accuracy to the competitors. The slightly worse test error is due to training with a marginally smaller dataset, but this is an acceptable cost when calibration is the priority. We also study adaptive (Nixon et al., 2019) and classwise (Kull et al., 2019) ECE calibration metrics in Table 3 and 4 in the Appendix. Note that while Brier score, Focal loss, FLSD and LS modify the base model's loss function objective, our Meta-Calibration corresponds exactly to the vanilla cross-entropy baseline, but where L2 regularization is tuned by our DECE-driven hyperparameter meta-learning rather than chosen ad-hoc. While we have explored calibration via simple L2 regularization, diverse hyperparameters of the base model can be trained by meta-learning (Hospedales et al., 2021), and future work exploring DECE's application to different meta-learning targets is likely to improve the results further.

3.3. Further Analysis

DECE vs ECE: To assess the quality of our differentiable approximation to ECE, we evaluate DECE and ECE correlation. We trained the same ResNet-18 backbone on both CIFAR-10 and CIFAR-100 benchmarks for 350 epochs, recording DECE and ECE values at various points. The results in Figure 1 show both Spearman and Pearson correlation. In both cases they are close to 1, and become even closer to 1 as training continues. This shows that DECE is an accurate differentiable approximation of ECE.

Learned Regularizers: We also analyse the learned classifier regularization coefficients. We can see in Figure 2 that the values are distributed around 0 (their value at initialization), but their spread indicates that we have successfully meta-learned useful values for the calibration task.

Ablation Study: Finally, we perform an ablation study on the design of our approach. The results in Table 2 show various alternative designs in terms of the use of DECE, CE and the alternative calibration metric MMCE (Kumar et al., 2018) in either conventional or meta-loss roles. From the results we can conclude that: (1) Meta-learning with

Table 1. Test errors (%), test ECE (%) and test ECE (%) after applying temperature scaling (TS) optimized with the validation set. Mean and standard deviation across 3 repetitions. Optimal temperature is in brackets and is averaged over the 3 repetitions.

Dataset	Loss	Error	ECE	ECE + TS
CIFAR-10	Cross-entropy	4.99 ± 0.14	4.23 ± 0.15	1.16 ± 0.10 (2.30)
	Brier score (Brier, 1950)	5.27 ± 0.21	1.23 ± 0.03	1.23 ± 0.03 (1.00)
	MMCE (Kumar et al., 2018)	5.21 ± 0.17	4.41 ± 0.14	1.27 ± 0.10 (2.27)
	Focal loss (Lin et al., 2017)	5.17 ± 0.09	2.17 ± 0.03	1.22 ± 0.12 (0.90)
	FLSD (Mukhoti et al., 2020)	5.28 ± 0.13	2.26 ± 0.03	1.63 ± 0.18 (0.87)
	LS-0.05 (Müller et al., 2019)	4.94 ± 0.13	3.63 ± 0.06	1.31 ± 0.08 (0.90)
	Meta-Calibration (our)	5.69 ± 0.15	1.33 ± 0.18	1.33 ± 0.18 (1.00)
CIFAR-100	Cross-entropy	22.85 ± 0.17	8.79 ± 0.59	5.47 ± 0.22 (1.33)
	Brier score (Brier, 1950)	23.50 ± 0.17	5.19 ± 0.18	4.21 ± 0.23 (0.90)
	MMCE (Kumar et al., 2018)	23.64 ± 0.25	8.21 ± 0.20	6.25 ± 0.18 (1.20)
	Focal loss (Lin et al., 2017)	23.02 ± 0.08	2.77 ± 0.23	2.77 ± 0.23 (1.00)
	FLSD (Mukhoti et al., 2020)	23.06 ± 0.11	2.50 ± 0.11	2.84 ± 0.48 (0.97)
	LS-0.05 (Müller et al., 2019)	22.35 ± 0.27	6.87 ± 0.29	4.37 ± 0.45 (0.90)
	Meta-Calibration (our)	25.27 ± 0.16	2.96 ± 0.38	3.61 ± 0.54 (0.97)

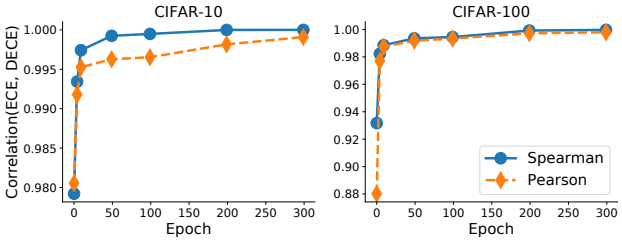

Figure 1. Correlation between DECE and ECE for ResNet-18 CIFAR-10 and CIFAR-100 is very close to 1.

Table 2. Ablation study on the design of our approach. CIFAR-10 at the top and CIFAR-100 below it.

Model	Meta-Loss	Loss	Error	ECE
M0: Vanilla CE	-	CE	4.99 ± 0.14	4.23 ± 0.15
M1: Multi-task	-	CE + DECE	10.24 ± 0.21	3.80 ± 0.03
M2: Meta-CE	CE	CE	5.93 ± 0.09	3.96 ± 0.18
M3: Meta-MMCE	MMCE	CE	5.95 ± 0.29	9.84 ± 2.62
M4: Meta-DECE	DECE	CE	5.69 ± 0.15	1.33 ± 0.18
M0: Vanilla CE	-	CE	22.85 ± 0.17	8.79 ± 0.59
M1: Multi-task	-	CE + DECE	29.49 ± 0.17	4.40 ± 0.39
M2: Meta-CE	CE	CE	26.47 ± 0.60	9.17 ± 3.89
M3: Meta-MMCE	MMCE	CE	26.38 ± 0.67	26.41 ± 1.33
M4: Meta-DECE	DECE	CE	25.27 ± 0.16	2.96 ± 0.38

DECE meta-objective is beneficial for improving calibration (M4 vs M0), while CE meta-objective does not generally improve ECE over vanilla training (M2 vs M0). (2) MMCE does not work well as a meta-objective (M3 vs M0). (3) DECE provides calibration benefit as a secondary loss in conventional learning, but at greater detriment to test error (M1 vs M0). (4) Our DECE-driven meta-objective (M4) is the best overall.

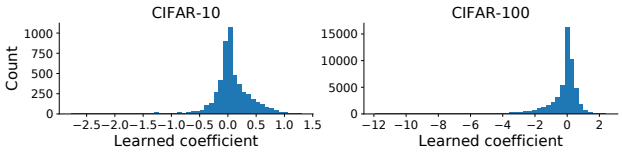

Figure 2. Histograms of the learned unit-wise L2 classifier regularization values. CIFAR-100 has 10 times as many values as CIFAR-10 because of 10 times more classes.

4. Discussion and Conclusion

This work is the first step in using meta-learning to directly optimize model calibration, and there are many ways how it could be expanded in the future. One direction is to target different types of meta-knowledge – we have used unit-wise L2 regularization for the classifier, and there are many other options that could work better. Second direction is to consider different ways of meta-learning, for example implicit meta-learning (Lorraine et al., 2020) could provide a better way to optimize the meta-objective. Third direction is to extend the differentiable metric itself.

We have demonstrated proof-of-concept approach that shows meta-learning can be used to optimize model calibration via hyperparameter tuning. The results show the approach is competitive with the existing approaches for model calibration, and it is likely different types of meta-knowledge and improved meta-optimizers will improve the calibration even further.

Software and Data

We provide a PyTorch implementation of our approach at https://github.com/ondrejbohdal/meta-calibration.

Acknowledgements

This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

References

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. (2018). MetaReg: towards domain generalization using meta-regularization. In *NeurIPS*.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. (2020). Fast differentiable sorting and ranking. In *ICML*.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. In *Monthly weather review*.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *ICML*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. In *CVPR*.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. (2021). Meta-learning in neural networks: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1.

Kim, S. and Yun, S.-Y. (2020). Task calibration for distributional uncertainty in few-shot classification. In *OpenReview*.

Kingma, D. P. and Ba, J. (2015). Adam: a method for stochastic optimization. In *ICLR*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Kull, M., Perello-Nieto, M., Kängsepp, M., Filho, T. S., Song, H., and Flach, P. (2019). Beyond temperature scaling: obtaining well-calibrated multiclass probabilities with Dirichlet calibration. In *NeurIPS*.

Kumar, A., Sarawagi, S., and Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *ICCV*.

Lorraine, J., Vicol, P., and Duvenaud, D. (2020). Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*.

Luketina, J., Berglund, M., Klaus Greff, A., and Raiko, T. (2016). Scalable gradient-based tuning of continuous regularization hyperparameters. In *ICML*.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., and Dokania, P. K. (2020). Calibrating deep neural networks using focal loss. In *NeurIPS*.

Müller, R., Kornblith, S., and Hinton, G. (2019). When does label smoothing help? In *NeurIPS*.

Naeini, P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *AAAI*.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. (2019). Measuring calibration in deep learning. In *CVPR Workshop*.

Qin, T., Liu, T.-Y., and Li, H. (2010). A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 13(4):375–397.

Yang, Y., Morillo, I. G., and Hospedales, T. M. (2018). Deep neural decision trees. In *ICML Workshop on Human Interpretability in Machine Learning*.

A. Additional Results

Tables 3 and 4 show that optimizing for ECE leads to strong improvements on adaptive ECE (Nixon et al., 2019), but less clear improvements on classwise ECE (Kull et al., 2019). The reason is that adaptive ECE is relatively close to the ECE formulation, while classwise ECE is further away due to the focus on all classes rather than only the top one.

Table 3. Test errors (%) and various calibration metrics evaluated on the test set (all in %) – ECE, Adaptive ECE (AECE) and Classwise ECE (CECE). Mean and standard deviation across 3 repetitions.

Dataset	Loss	Error	ECE	AECE	CECE
CIFAR-10	Cross-entropy	4.99 ± 0.14	4.23 ± 0.15	4.21 ± 0.14	0.87 ± 0.04
	Brier score (Brier, 1950)	5.27 ± 0.21	1.23 ± 0.03	2.00 ± 0.25	0.46 ± 0.02
	MMCE (Kumar et al., 2018)	5.21 ± 0.17	4.41 ± 0.14	4.39 ± 0.14	0.92 ± 0.03
	Focal loss (Lin et al., 2017)	5.17 ± 0.09	2.17 ± 0.03	2.16 ± 0.13	0.52 ± 0.02
	FLSD (Mukhoti et al., 2020)	5.28 ± 0.13	2.26 ± 0.03	2.20 ± 0.02	0.54 ± 0.02
	LS-0.05 (Müller et al., 2019)	4.94 ± 0.13	3.63 ± 0.06	4.18 ± 0.25	0.73 ± 0.01
	Meta-Calibration (our)	5.69 ± 0.15	1.33 ± 0.18	1.86 ± 0.22	0.61 ± 0.05
CIFAR-100	Cross-entropy	22.85 ± 0.17	8.79 ± 0.59	8.68 ± 0.61	0.23 ± 0.01
	Brier score (Brier, 1950)	23.50 ± 0.17	5.19 ± 0.18	5.16 ± 0.18	0.24 ± 0.00
	MMCE (Kumar et al., 2018)	23.64 ± 0.25	8.21 ± 0.20	8.14 ± 0.22	0.23 ± 0.00
	Focal loss (Lin et al., 2017)	23.02 ± 0.08	2.77 ± 0.23	2.71 ± 0.17	0.20 ± 0.00
	FLSD (Mukhoti et al., 2020)	23.06 ± 0.11	2.50 ± 0.11	2.49 ± 0.12	0.20 ± 0.00
	LS-0.05 (Müller et al., 2019)	22.35 ± 0.27	6.87 ± 0.29	6.87 ± 0.28	0.26 ± 0.00
	Meta-Calibration (our)	25.27 ± 0.16	2.96 ± 0.38	2.99 ± 0.37	0.25 ± 0.02

Table 4. Test errors (%) and ECE, AECE and CECE calibration metrics evaluated on the test set after applying temperature scaling (all in %). Mean and standard deviation across 3 repetitions. The temperature scaling values are the same as reported in Table 1.

Dataset	Loss	Error	ECE + TS	AECE + TS	CECE + TS
CIFAR-10	Cross-entropy	4.99 ± 0.14	1.16 ± 0.10	2.24 ± 0.13	0.45 ± 0.02
	Brier score (Brier, 1950)	5.27 ± 0.21	1.23 ± 0.03	2.00 ± 0.25	0.46 ± 0.02
	MMCE (Kumar et al., 2018)	5.21 ± 0.17	1.27 ± 0.10	2.06 ± 0.04	0.49 ± 0.01
	Focal loss (Lin et al., 2017)	5.17 ± 0.09	1.22 ± 0.12	1.90 ± 0.12	0.41 ± 0.03
	FLSD (Mukhoti et al., 2020)	5.28 ± 0.13	1.63 ± 0.18	1.95 ± 0.14	0.45 ± 0.01
	LS-0.05 (Müller et al., 2019)	4.94 ± 0.13	1.31 ± 0.08	3.10 ± 0.27	0.51 ± 0.02
	Meta-Calibration (our)	5.69 ± 0.15	1.33 ± 0.18	1.86 ± 0.22	0.61 ± 0.05
CIFAR-100	Cross-entropy	22.85 ± 0.17	5.47 ± 0.22	5.40 ± 0.26	0.21 ± 0.01
	Brier score (Brier, 1950)	23.50 ± 0.17	4.21 ± 0.23	4.24 ± 0.23	0.21 ± 0.00
	MMCE (Kumar et al., 2018)	23.64 ± 0.25	6.25 ± 0.18	6.21 ± 0.22	0.21 ± 0.00
	Focal loss (Lin et al., 2017)	23.02 ± 0.08	2.77 ± 0.23	2.71 ± 0.17	0.20 ± 0.00
	FLSD (Mukhoti et al., 2020)	23.06 ± 0.11	2.84 ± 0.48	2.83 ± 0.49	0.20 ± 0.00
	LS-0.05 (Müller et al., 2019)	22.35 ± 0.27	4.37 ± 0.45	4.24 ± 0.22	0.21 ± 0.00
	Meta-Calibration (our)	25.27 ± 0.16	3.61 ± 0.54	3.61 ± 0.52	0.25 ± 0.02