IMPROVING THE POST-HOC CALIBRATION OF MODERN NEURAL NETWORKS WITH PROBE SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present "probe scaling": a post-hoc recipe for calibrating the predictions of modern neural networks. Our recipe is inspired by several lines of work, which demonstrate that early layers in the neural network learn general rules whereas later layers specialize. We show how such observations can be utilized in a post-hoc manner to calibrate the predictions of trained neural networks by injecting linear probes on the network's intermediate representations. Similar to temperature scaling, probe scaling neither retrains the architecture nor requires significantly more parameters. Unlike temperature scaling, however, it utilizes intermediate layers in the neural network. We demonstrate that probe scaling improves performance over temperature scaling on benchmark datasets across all five metrics: expected calibration error (ECE), negative log-likelihood, Brier score, classification accuracy, and the area under the ROC curve.

1 INTRODUCTION

Calibration is a measure of how accurate a model is in estimating its own uncertainty. It is an important metric for evaluating predictive models, particularly in critical domains such as medical diagnosis. In the latter case, having the ability to highlight predictions with low confidence is crucial in order for them to be handled separately, such as by referring to consultants for second opinions or by supplying additional information (e.g. lab tests), among others (Kompa et al., 2021). Providing an accurate estimate of uncertainty is also important in *selective predictions* (a.k.a. reject option classifiers) in which one optimizes a weighted combination of the cost of abstention and the cost of providing an incorrect prediction (Geifman & El-Yaniv, 2017)). It is also important when debiasing trained models via post-processing, where group-specific thresholding rules are adjusted to maximize accuracy subject to prescribed fairness guarantees (Corbett-Davies et al., 2017; Menon & Williamson, 2018; Celis et al., 2019; Alabdulmohsin & Lucic, 2021).

In order to achieve such objectives, however, trained machine learning models need to be *calibrated*, i.e. their predicted probabilities of outcomes should closely match with the observed empirical frequencies. More formally, writing $\mathcal{X}$ for the instance space (e.g. the space of images), $\mathcal{Y} = \{0, 1, 2, \dots, K-1\}$ for the target set, and Δ_K for the probability simplex in $\mathbb{R}^K$, a probabilistic multiclass classifier $f : \mathcal{X} \to \Delta_K$ that outputs class probabilities for K classes is said to be calibrated if (Guo et al., 2017):

$$\forall y \in \mathcal{Y} : \; p(\mathbf{y} = y \,|\, f(\mathbf{x}) = q) = q_y, \tag{1}$$

where q_y is the y-th coordinate of the probability distribution $q \in \Delta_K$. In a slightly weaker notion, $f : \mathcal{X} \to \Delta_K$ is said to be *classwise* calibrated if (Kull et al., 2019):

$$\forall y \in \mathcal{Y} : \; p(\mathbf{y} = y \,|\, f_y(\mathbf{x}) = q_y) = q_y. \tag{2}$$

Unfortunately, several lines of work observed that modern neural networks lack such calibration (Guo et al., 2017; Thulasidasan et al., 2019; Ovadia et al., 2019; Kumar & Sarawagi, 2019; Kull et al., 2019), particularly under distribution shift (Ovadia et al., 2019). To remedy this problem, various approaches have been proposed including Monte Carlo dropout (Gal & Ghahramani, 2016), deep ensembles (Lakshminarayanan et al., 2016), training multiple independent subnetworks (Havasi et al., 2021), and augmentation (Thulasidasan et al., 2019). Often, such calibration methods inject *diversity* by generating multiple predictions for the same instance $\mathbf{x} \in \mathcal{X}$.

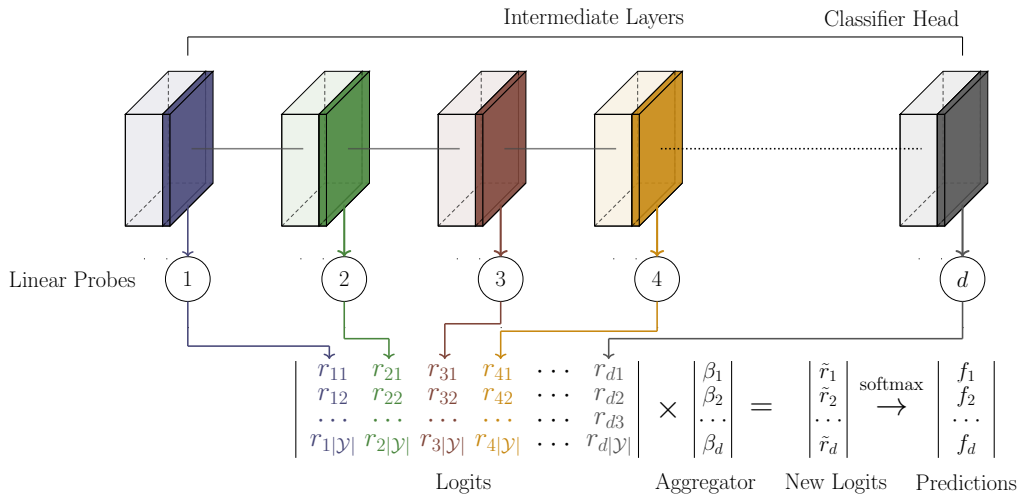

Figure 1: In probe scaling, d linear probes are injected into the intermediate representations of the neural network. Each probe is trained to maximize a proper score, such as the cross-entropy. Afterward, the logits of all probes are concatenated together to form a new representation. Finally, a classifier with d parameters (shown as β's) is trained to output the calibrated scores using a 1D convolutional layer with strides d (equivalent to matrix multiplication) followed by softmax activations.

Besides diversity-based approaches, one particularly effective calibration rule is *temperature scaling* (Guo et al., 2017). It is an extension of the classical Platt scaling algorithm (Platt et al., 1999) to the multiclass setting. Temperature scaling rescales the neural networks' logits by a scalar $\tau \in \mathbb{R}^+$ chosen according to a separate validation dataset to optimize a *proper* scoring rule, where proper scores are those that guarantee to yield a perfectly calibrated classifier at its minimum at the infinite data limit (Gneiting & Raftery, 2007). Proper scores include, for example, the negative log-likelihood (a.k.a. cross-entropy loss) and the Brier score (a.k.a. squared loss). Despite the fact that temperature scaling is a post-hoc rule that neither requires changes to the training procedure nor adds significantly more parameters, it has been observed to yield competitive results in practice for in-distribution data (Guo et al., 2017; Ovadia et al., 2019; Kull et al., 2019).

In this work, we propose extending temperature scaling to "probe scaling", which is a post-hoc calibration method that utilizes the learned representations of *intermediate layers* in the neural network. An overview of probe scaling is presented in Figure 1. First, given a deep neural network that can be partitioned into d blocks (e.g. convolutional blocks in the standard ResNet and VGG architectures (He et al., 2016; Simonyan & Zisserman, 2015)), the k-th linear probe is trained to predict the final class $\mathbf{y} \in \mathcal{Y}$ given the representation learned at block k for the instance $\mathbf{x} \in \mathcal{X}$. In total, $d-1$ linear probes are trained accordingly while the d-th probe is an identity mapping that preserves the logits of the original classifier. Finally, the logits of all probes are concatenated to form a new representation in the logit space with $|\mathcal{Y}| \times d$ features as shown in Figure 1, where $|\mathcal{Y}|$ is the number of classes. A final linear classifier is trained to optimize a proper score, such as the cross-entropy, on the new representation. Full pseudocode is provided in Algorithm 1.

Clearly, temperature scaling is a particular instance of probe scaling in which a single probe is used (i.e. $d = 1$). However, unlike temperature scaling, probe scaling often *improves* the accuracy of the original classifier besides improving its calibration. In fact, as discussed in Section 3, probe scaling performs better than temperature scaling on all considered five metrics: expected calibration error (ECE), negative log-likelihood, Brier score, classification accuracy, and the area under the ROC curve. The following proposition provides a formal argument for the advantage of probe scaling over temperature scaling when the aggregator is trained to optimize a proper, convex, Lipschitz scoring function, such as the logistic loss in the binary classification setting:

Proposition 1 *Let $f : \mathbb{R}^d \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a scoring function that is convex and ρ-Lipschitz continuous on its first argument. For a given instance space $\mathcal{X}$ and a target set $\mathcal{Y}$, denote by $\boldsymbol{S} \in$*

$(\mathcal{X} \times \mathcal{Y})^N \sim \mathcal{D}^N$ a hold-out validation dataset of size N. Let β_{probe} be the aggregator weights in probe scaling that minimize the regularized empirical loss on the hold-out sample:

$$\beta_{probe} = \arg\min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \in \boldsymbol{S}}\big[(\lambda/2)\,||\beta||^2 + f(\beta, \boldsymbol{x}, \boldsymbol{y})\big], \tag{3}$$

Then, the probability that the solution of temperature scaling β_{temp} performs better than probe scaling β_{probe} with respect to f by more than $\epsilon > 0$ over the random choice of the hold-out sample is bounded by:

$$p_{\boldsymbol{S} \sim \mathcal{D}^N}\Big\{\mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{D}}\big[f(\beta_{temp}, \boldsymbol{x}, \boldsymbol{y})\big] \leq \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{D}}\big[f(\beta_{probe}, \boldsymbol{x}, \boldsymbol{y})\big] - \epsilon\Big\} \leq \frac{1}{\epsilon}\big(\lambda||\beta^\star||^2 + \frac{2\rho^2}{\lambda N}\big), \tag{4}$$

where $\beta^\star$ is the optimal probe scaling parameters at the limit of an infinite validation data size $N \to \infty$. Consequently, by setting $\lambda = 1/\sqrt{N}$, the probability bound $(||\beta^\star||^2 + 2\rho^2)/(\epsilon\sqrt{N})$ can be made arbitrarily small for a sufficiently large N for any fixed $\epsilon > 0$.

In addition, one informal justification for why probe scaling improves calibration (and accuracy) is that early layers tend to learn general-purpose representations whereas later layers specialize. This has been observed in several works, such as when studying transfer learning (Raghu et al., 2019; Yosinski et al., 2014), memorization (Arpit et al., 2017; Cohen et al., 2018), and pretraining with random labels (Maennel et al., 2020). In particular, "easy" examples tend to be classified correctly by all layers of the neural network while difficult examples tend to be classified correctly by the last layers alone (Cohen et al., 2018; Baldock et al., 2021). By injecting linear probes on the intermediate representations of the neural network, the model improves its estimate of its own uncertainty by comparing the predictions at different layers.

2 Contribution and Related Work

Measures of Calibration To evaluate calibration, several scores have been introduced in the literature. One popular score is the expected calibration error (ECE), which in its confidence-based version is given by (Naeini et al., 2015; Guo et al., 2017; Nixon et al., 2019):

$$\mathrm{ECE} = \sum_{j=1}^{M} \frac{|B_j|}{n}\,\big|\mathrm{acc}(B_j)\;-\;p(B_j)\big|, \tag{5}$$

where $\{B_j\}_{j=1,\ldots,M}$ are the bins, $|B_j|$ is the size of the bin, $acc(B_j)$ is the model's accuracy within the bin, and $p(B_j)$ is the model's average confidence. While often used as a default measure of calibration, ECE has its own limitations including high sensitivity to the number of chosen bins (Nixon et al., 2019). As a result, proper scoring rules are used as well. As discussed earlier, a scoring function is called proper if it is optimized by the true probability distribution, and it is called *strictly proper* if it has a unique optima (Gneiting & Raftery, 2007). One commonly used proper scoring function is the Brier score (Brier et al., 1950):

$$\mathrm{Brier} = \mathbb{E}_{\mathbf{x},\mathbf{y}}\big[\sum_{y \in \mathcal{Y}} p(y \,|\, \mathbf{x})^2\;-\;2p(\mathbf{y} \,|\, \mathbf{x})\big], \tag{6}$$

where $\mathbf{y}$ is the true label and $p(y \,|\, \mathbf{x})$ is the probability assigned by the model to the class $y \in \mathcal{Y}$ when the instance is $\mathbf{x}$. Note that the Brier score is an affine transformation of the square loss and can be negative when Equation 6 is used. Besides, the negative log-likelihood (NLL), a.k.a. cross entropy, is a second proper score, which assigns a more aggressive penalty than the Brier score for events that occur when their predicted probabilities were quite small (Bickel, 2007). In our evaluation, we use the implementation of the ECE, Brier score, and NLL in Tensorflow (Abadi et al., 2015).

Calibration Methods. Several methods have been proposed to improve the calibration of neural networks. These include regularization-based methods, such as the Maximum Mean Calibration Error (MMCE), which adds a penalty term to the loss that is analogous to the Maximum Mean Discrepancy (MMD) method for testing the difference between two probability distributions (Kumar et al., 2018). They also include Bayesian methods, such as Stochastic Weight Average Gaussian (SWAG), which estimates uncertainty by sampling from the posterior distribution of the weights

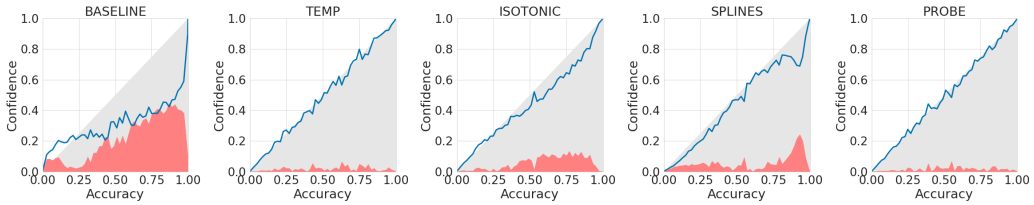

Figure 2: Reliability diagrams are plotted for all five calibration methods when ResNet50 is trained on CIFAR100. Red shaded regions indicate the discrepancy between the confidence of the classifier and its accuracy. In general, both temperature scaling and probe scaling perform better than the other post-hoc methods with probe scaling performing best overall (see details in Section 3).

(Maddox et al., 2019). Other popular choices include ensemble-based approaches such as by training multiple networks with different seeds (Lakshminarayanan et al., 2017), augmentation-based methods such as Mix-Up (Thulasidasan et al., 2019), or variational inference based methods such as the Monte Carlo dropout (Gal & Ghahramani, 2016).

The proposed probe scaling method, on the other hand, is a *post-hoc* recipe for improving the calibration of modern neural networks. It extends temperature scaling (Guo et al., 2017) to intermediate layers in the neural network. Besides temperature scaling, an alternative post-hoc recipe for calibrating the predictions of neural networks is isotonic regression, which learns a monotonic transformation of the model's predictions into the true labels (Zadrozny & Elkan, 2002). In Gupta et al. (2021), another post-hoc recipe is proposed that is inspired by the Kolmogorov-Smirnov test, which approximates cumulative distribution functions using splines and estimates uncertainty using derivatives. For illustration, Figure 2 displays the reliability diagrams of these post-hoc rules when applied to ResNet50 (He et al., 2016) on the CIFAR100 dataset (Krizhevsky, 2009). A more thorough evaluation of all three post-hoc methods as well as probe scaling is provided in Section 3.

Statement of Contribution. In summary, we propose extending temperature scaling to intermediate layers in the neural network using linear probes. We provide a comprehensive evaluation showing that "probe scaling" outperforms other post-hoc rules in common benchmark datasets, including under distribution shift. We also demonstrate that probe scaling can complement other existing techniques, such as Monte Carlo dropout.

3 Experimental Setup

3.1 Comparison with Other Post-hoc Methods

Data Splits. To calibrate deep neural networks using post-hoc methods, we split the data into three parts: a training dataset (90% of standard training split), a hold-out dataset (10% of standard training split), and a test dataset (standard test split). In all methods, the original architecture is trained on the training dataset while the post-hoc rule is trained on the hold-out dataset. Final results are reported on the test data. In addition, the linear probes in the proposed probe scaling algorithm are trained on the training dataset as well. Nevertheless, we remark that the latter is by no means necessary since the probes can be applied to architectures that were pretrained on large datasets, such as ImageNet-21k (a superset of ILSVRC2012 that contains 21k classes (Deng et al., 2009)).

Datasets and Architectures. In our experiments, we train the models on CIFAR10/100 (Krizhevsky, 2009), Street View House Numbers (SVHN) (Netzer et al., 2011), and Fashion-MNIST (Xiao et al., 2017). We use five architectures: VGG16 and VGG19 (Simonyan & Zisserman, 2015), ResNet18 and ResNet50 (He et al., 2016), and Wide-ResNet-28-10 (Zagoruyko & Komodakis, 2016). We also conduct experiments on ImageNet ILSVRC2012 (Deng et al., 2009) using ResNet50 (He et al., 2016). To assess the impact of distribution shift, we evaluate ImageNet trained models on both clean images as well as the ImageNet-C corrupted images (Hendrycks & Dietterich, 2019).

Training Procedure. For reproducibility, we use a simple input training pipeline. We implement standard architectures with default ℓ_2 regularization of 10^{-4}. We train the model for 200 epochs on

Algorithm 1 Pseudocode of Probe Scaling

Input: (1) Trained neural network with identified sequence of $d-1 \geq 0$ blocks; (2) Hold-out dataset.

Output: Calibrated model.

Training:

1: Insert $d-1$ linear probes, where probe k is trained to predict the target $\mathbf{y} \in \mathcal{Y}$ given the output (intermediate representation) of block k.

2: Concatenate the logits of all $d-1$ probes in addition to the logits of the original neural network, forming a new representation with $d \times |\mathcal{Y}|$ features. Let:

$$R(\mathbf{x}) = \begin{bmatrix} r_{1,1} & r_{2,1} & \cdots & r_{d,1} \\ r_{1,2} & r_{2,2} & \cdots & r_{d,2} \\ \cdots & \cdots & \cdots & \cdots \\ r_{1,|\mathcal{Y}|} & r_{2,|\mathcal{Y}|} & \cdots & r_{d,|\mathcal{Y}|} \end{bmatrix}$$

be the new representation, where $r_{i,y}$ is the logit of the i-th probe for the class y.

3: Initialize $\beta_0 = (0, 0, \ldots, 1)^T \in \mathbb{R}^d$.

4: Train an aggregator using the projected stochastic gradient descent (SGD) that minimizes:

$$\mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[l\big(\text{softmax}\big(R(\mathbf{x})\,\beta\big),\ \mathbf{y}\big)\Big],$$

over the d parameters $\beta \in \mathbb{R}^d$ subject to the constraint $\beta \geq 0$, and starting from the initial value β_0, for some proper scoring function l such as the cross-entropy. Here, $R(\mathbf{x})\,\beta$ is matrix multiplication, which can be implemented using 1D convolutional layers with strides d.

CIFAR10/100, for 100 epochs on SVHN, and for 50 epochs on Fashion-MNIST, all with a batch size of 128. We use SGD with an initial learning rate of 0.1 and momentum of 0.9. We decay the learning rate with cosine learning rate decay (Loshchilov & Hutter, 2016). On the input pipeline, we standardize the input to have values in the unit interval $[0,\ 1]$ and use only right-left random flipping as data augmentation. For ImageNet experiments, we use the pretrained ResNet50 architecture available in Keras (Chollet et al., 2015). We train on NVIDIA Tesla V100 GPUs for the medium-size datasets and train the probes in ImageNet on 32 TPUs. After training, we freeze the weights.

In probe scaling, each probe comprises of a fully connected classifier head layer with softmax activations trained using the cross-entropy loss. Before the classifier head, we flatten the intermediate representation that the probe is connected to, and we don't use any non-linearity between the backbone layers and the probe. For simplicity and reproducibility, we place a linear probe after every convolutional block in the ResNet and Wide-ResNet architectures, and after every convolutional layer in the VGG architectures. This results in 9 probes in ResNet18 (including the logits of the original classifier), 19 probes in ResNet50, 13 probes in Wide-ResNet-28-10, 17 probes in VGG16, and 20 probes in VGG19. Since the backbone network is frozen, training the probes can be carried out quickly. We train each probe with a learning rate of 0.005 and momentum 0.9 for 50 epochs, although we observe that they converge much faster. After each probe is trained, we also freeze it.

The final step in probe scaling is to train the final model, which aggregates the logits of all probes. It takes as input the concatenated output of each probe and performs a strided 1D convolution on it (See Figure 1 and Algorithm 1), which is equivalent to using one parameter per probe. We train the aggregator model for 50 epochs with a fixed learning rate of 0.005, using the hold-out part of the dataset. We repeat all experiments five times and report averages. Section 4.1 presents the results.

3.2 Complementing Existing Algorithms

The proposed probe scaling algorithm can also be combined with other existing techniques to improve calibration further. We illustrate this using the popular Monte Carlo dropout method (Gal & Ghahramani, 2016). In Monte Carlo dropout, multiple predictions are generated for the same instance $\mathbf{x} \in \mathcal{X}$ by adding dropout layers to the model and using them at *both* training and test time. Then, the model's uncertainty can be estimated by averaging the predictions.

In our experiments, we use the implementation of the Monte Carlo dropout available at the Uncertainty Baselines Benchmark (Nado et al., 2021). We use the same implementation details and

Table 1: Empirical evaluation of post-hoc calibration methods on CIFAR100 (Krizhevsky, 2009) for the five metrics: ECE, negative log-likelihood (NLL), Brier score, accuracy (ACC), and AUC.

		ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
VGG16	BASELINE	0.201 ± 0.006	1.759 ± 0.044	-0.513 ± 0.011	0.698 ± 0.007	0.923 ± 0.002
	TEMP	0.041 ± 0.003	1.238 ± 0.026	-0.582 ± 0.008	0.698 ± 0.007	0.969 ± 0.002
	ISOTONIC	0.069 ± 0.005	1.392 ± 0.039	-0.565 ± 0.009	0.687 ± 0.006	0.959 ± 0.003
	SPLINES	0.049 ± 0.010	3.382 ± 0.231	-0.273 ± 0.030	0.459 ± 0.025	0.847 ± 0.016
	PROBE	**0.025 ± 0.004**	**0.981 ± 0.020**	**-0.628 ± 0.007**	**0.726 ± 0.006**	**0.984 ± 0.001**
VGG19	BASELINE	0.217 ± 0.005	2.002 ± 0.040	-0.497 ± 0.010	0.697 ± 0.006	0.908 ± 0.002
	TEMP	0.046 ± 0.005	1.303 ± 0.022	-0.581 ± 0.008	0.697 ± 0.006	0.963 ± 0.001
	ISOTONIC	0.057 ± 0.007	1.436 ± 0.029	-0.568 ± 0.008	0.686 ± 0.007	0.956 ± 0.002
	SPLINES	0.052 ± 0.012	3.929 ± 0.313	-0.221 ± 0.033	0.412 ± 0.035	0.818 ± 0.020
	PROBE	**0.026 ± 0.003**	**0.969 ± 0.018**	**-0.633 ± 0.006**	**0.731 ± 0.005**	**0.986 ± 0.001**
RESNET18	BASELINE	0.119 ± 0.004	1.183 ± 0.033	-0.604 ± 0.009	0.728 ± 0.006	0.960 ± 0.002
	TEMP	0.037 ± 0.004	1.047 ± 0.025	-0.627 ± 0.007	0.728 ± 0.006	0.979 ± 0.001
	ISOTONIC	0.060 ± 0.005	1.245 ± 0.031	-0.597 ± 0.008	0.706 ± 0.006	0.968 ± 0.002
	SPLINES	0.048 ± 0.011	2.975 ± 0.285	-0.328 ± 0.035	0.486 ± 0.029	0.870 ± 0.020
	PROBE	**0.025 ± 0.003**	**0.925 ± 0.021**	**-0.651 ± 0.007**	**0.746 ± 0.005**	**0.985 ± 0.001**
RESNET50	BASELINE	0.185 ± 0.006	1.675 ± 0.033	-0.561 ± 0.009	0.731 ± 0.006	0.933 ± 0.002
	TEMP	0.021 ± 0.003	0.965 ± 0.015	-0.634 ± 0.006	0.731 ± 0.006	0.983 ± 0.001
	ISOTONIC	0.058 ± 0.006	1.353 ± 0.028	-0.545 ± 0.008	0.670 ± 0.007	0.967 ± 0.002
	SPLINES	0.041 ± 0.007	2.769 ± 0.202	-0.343 ± 0.025	0.499 ± 0.021	0.885 ± 0.014
	PROBE	**0.018 ± 0.003**	**0.949 ± 0.015**	**-0.637 ± 0.006**	**0.733 ± 0.007**	**0.984 ± 0.001**
WIDE-RESNET-28-10	BASELINE	0.054 ± 0.004	0.890 ± 0.017	**-0.691 ± 0.005**	**0.785 ± 0.004**	0.981 ± 0.001
	TEMP	0.054 ± 0.004	0.890 ± 0.017	**-0.691 ± 0.005**	**0.785 ± 0.004**	0.981 ± 0.001
	ISOTONIC	**0.046 ± 0.004**	0.980 ± 0.024	-0.686 ± 0.006	0.775 ± 0.005	0.972 ± 0.002
	SPLINES	0.054 ± 0.011	3.312 ± 0.263	-0.300 ± 0.035	0.470 ± 0.029	0.839 ± 0.020
	PROBE	0.053 ± 0.004	**0.887 ± 0.017**	**-0.691 ± 0.005**	**0.785 ± 0.004**	**0.982 ± 0.001**

Table 2: This table lists the p-values using the non-parametric Wilcoxon signed-rank test for each post-hoc method compared against probe scaling. In all cases, statistics are in favor of probe scaling. The p-values shown in **bold** remain statistically significant at the 95% confidence level even after correcting for multiple hypothesis testing using Holm's step-down procedure (Demšar, 2006).

	ECE	NLL	Brier	ACC	AUC
BASELINE	$\mathbf{1 \times 10^{-4}}$	$\mathbf{8 \times 10^{-5}}$	$\mathbf{9 \times 10^{-5}}$	$\mathbf{2 \times 10^{-4}}$	$\mathbf{9 \times 10^{-5}}$
TEMP	$\mathbf{8 \times 10^{-3}}$	$\mathbf{3 \times 10^{-4}}$	$\mathbf{9 \times 10^{-5}}$	$\mathbf{2 \times 10^{-4}}$	$\mathbf{9 \times 10^{-4}}$
ISOTONIC	3×10^{-1}	$\mathbf{3 \times 10^{-4}}$	$\mathbf{5 \times 10^{-4}}$	$\mathbf{1 \times 10^{-3}}$	$\mathbf{2 \times 10^{-4}}$
SPLINES	1×10^{-1}	$\mathbf{9 \times 10^{-5}}$	$\mathbf{1 \times 10^{-4}}$	$\mathbf{1 \times 10^{-4}}$	$\mathbf{9 \times 10^{-5}}$

hyper-parameters suggested by the benchmark, which uses the Wide-ResNet-28-10 architecture, and follow it by probe scaling as described earlier in Section 3.1. We also test on different values of the dropout rate. The final results are presented in Section 4.2.

4 Results and Discussion

4.1 Post-hoc Caliberation Results

Tables 1-5 present the empirical results of using probe scaling to calibrate neural networks in a post-hoc manner. Five metrics are used in this evaluation: expected calibration error (ECE), negative log-likelihood (NLL), Brier score (Brier), accuracy (ACC), and the area under the ROC curve (AUC). We compare probe scaling to vanilla training (baseline), temperature scaling (Guo et al., 2017), isotonic regression (Zadrozny & Elkan, 2002), and splines (Gupta et al., 2021). We observe that probe scaling consistently outperforms other post-hoc methods in all metrics except on ECE in which no method seems to dominate. Also, probe scaling is the only post-hoc method that consistently improves the accuracy of the model and its AUC. For ImageNet and ImageNet-C, on the other hand, probe scaling seems to offer a modest gain over temperature scaling as shown in Table 6.

Statistical Significance. To determine if the improvement of probe scaling over other post-hoc methods is statistically significant, we used the non-parametric Wilcoxon signed-rank test and cor-

Table 3: Empirical evaluation of post-hoc calibration methods on CIFAR10 (Krizhevsky, 2009) for the five metrics: ECE, negative log-likelihood (NLL), Brier score, accuracy (ACC) and AUC.

		ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
VGG16	BASELINE	0.060 ± 0.002	0.423 ± 0.018	-0.869 ± 0.004	0.926 ± 0.002	0.979 ± 0.001
	TEMP	0.025 ± 0.003	0.263 ± 0.009	-0.885 ± 0.004	0.926 ± 0.002	0.994 ± 0.000
	ISOTONIC	0.018 ± 0.003	0.260 ± 0.009	-0.887 ± 0.003	0.924 ± 0.003	0.993 ± 0.001
	SPLINES	**0.013 ± 0.002**	0.298 ± 0.011	-0.886 ± 0.003	0.924 ± 0.003	0.992 ± 0.001
	PROBE	0.016 ± 0.003	**0.226 ± 0.007**	**-0.892 ± 0.004**	**0.927 ± 0.002**	**0.996 ± 0.000**
VGG19	BASELINE	0.061 ± 0.003	0.444 ± 0.021	-0.869 ± 0.005	0.927 ± 0.002	0.977 ± 0.001
	TEMP	0.023 ± 0.003	0.269 ± 0.009	-0.885 ± 0.004	0.927 ± 0.002	0.994 ± 0.001
	ISOTONIC	0.017 ± 0.004	0.269 ± 0.013	-0.886 ± 0.004	0.926 ± 0.003	0.993 ± 0.001
	SPLINES	**0.012 ± 0.002**	0.313 ± 0.018	-0.885 ± 0.003	0.924 ± 0.003	0.991 ± 0.001
	PROBE	0.017 ± 0.002	**0.226 ± 0.006**	**-0.892 ± 0.003**	**0.928 ± 0.002**	**0.996 ± 0.000**
RESNET18	BASELINE	0.038 ± 0.002	0.265 ± 0.016	-0.904 ± 0.004	**0.941 ± 0.003**	0.989 ± 0.001
	TEMP	0.013 ± 0.003	0.191 ± 0.010	**-0.912 ± 0.004**	**0.941 ± 0.003**	**0.996 ± 0.001**
	ISOTONIC	0.015 ± 0.004	0.240 ± 0.026	-0.893 ± 0.011	0.928 ± 0.008	0.994 ± 0.001
	SPLINES	**0.011 ± 0.004**	0.247 ± 0.036	-0.903 ± 0.014	0.933 ± 0.014	0.994 ± 0.001
	PROBE	0.015 ± 0.002	**0.184 ± 0.009**	**-0.912 ± 0.004**	**0.941 ± 0.002**	**0.996 ± 0.000**
RESNET50	BASELINE	0.046 ± 0.002	0.341 ± 0.016	-0.893 ± 0.003	**0.936 ± 0.002**	0.986 ± 0.001
	TEMP	**0.010 ± 0.002**	0.190 ± 0.006	**-0.907 ± 0.002**	**0.936 ± 0.002**	**0.997 ± 0.000**
	ISOTONIC	0.028 ± 0.008	0.440 ± 0.065	-0.791 ± 0.031	0.857 ± 0.022	0.986 ± 0.003
	SPLINES	0.046 ± 0.016	0.740 ± 0.146	-0.718 ± 0.049	0.769 ± 0.041	0.961 ± 0.014
	PROBE	**0.010 ± 0.002**	**0.189 ± 0.006**	**-0.907 ± 0.002**	**0.936 ± 0.002**	**0.997 ± 0.000**
WIDE-RESNET-28-10	BASELINE	0.026 ± 0.002	0.176 ± 0.010	-0.928 ± 0.003	**0.956 ± 0.002**	0.994 ± 0.001
	TEMP	0.012 ± 0.002	0.149 ± 0.007	**-0.932 ± 0.003**	**0.956 ± 0.002**	**0.997 ± 0.000**
	ISOTONIC	0.012 ± 0.002	0.157 ± 0.008	**-0.932 ± 0.003**	0.955 ± 0.002	**0.997 ± 0.001**
	SPLINES	**0.009 ± 0.002**	0.192 ± 0.014	-0.929 ± 0.003	0.953 ± 0.003	0.995 ± 0.001
	PROBE	0.011 ± 0.001	**0.147 ± 0.006**	**-0.932 ± 0.003**	**0.956 ± 0.002**	**0.997 ± 0.000**

Table 4: Empirical evaluation of post-hoc calibration methods on SVHN (Netzer et al., 2011) for the five metrics: ECE, negative log-likelihood (NLL), Brier score, accuracy (ACC) and AUC.

		ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
VGG16	BASELINE	0.021 ± 0.003	0.227 ± 0.009	-0.907 ± 0.005	0.940 ± 0.004	0.993 ± 0.001
	TEMP	0.012 ± 0.002	0.218 ± 0.011	-0.909 ± 0.005	0.940 ± 0.004	0.995 ± 0.000
	ISOTONIC	**0.006 ± 0.001**	0.208 ± 0.009	-0.914 ± 0.004	0.944 ± 0.003	0.994 ± 0.001
	SPLINES	0.013 ± 0.004	0.420 ± 0.033	-0.869 ± 0.009	0.917 ± 0.005	0.980 ± 0.002
	PROBE	0.011 ± 0.002	**0.191 ± 0.009**	**-0.920 ± 0.004**	**0.947 ± 0.003**	**0.996 ± 0.000**
VGG19	BASELINE	0.019 ± 0.002	0.205 ± 0.011	-0.918 ± 0.004	0.947 ± 0.003	0.993 ± 0.001
	TEMP	0.012 ± 0.002	0.199 ± 0.009	-0.920 ± 0.004	0.947 ± 0.003	0.995 ± 0.000
	ISOTONIC	**0.006 ± 0.001**	0.194 ± 0.010	-0.922 ± 0.004	0.949 ± 0.003	0.994 ± 0.000
	SPLINES	0.013 ± 0.004	0.408 ± 0.044	-0.877 ± 0.010	0.923 ± 0.005	0.981 ± 0.003
	PROBE	0.010 ± 0.003	**0.181 ± 0.007**	**-0.925 ± 0.003**	**0.951 ± 0.002**	**0.996 ± 0.000**
RESNET18	BASELINE	0.026 ± 0.002	0.250 ± 0.018	-0.901 ± 0.007	0.937 ± 0.005	0.991 ± 0.001
	TEMP	0.012 ± 0.003	0.228 ± 0.016	-0.904 ± 0.007	0.937 ± 0.005	0.995 ± 0.001
	ISOTONIC	**0.007 ± 0.001**	0.218 ± 0.014	-0.908 ± 0.006	0.940 ± 0.004	0.994 ± 0.001
	SPLINES	0.011 ± 0.003	0.409 ± 0.031	-0.868 ± 0.006	0.917 ± 0.004	0.981 ± 0.003
	PROBE	0.013 ± 0.002	**0.205 ± 0.013**	**-0.912 ± 0.006**	**0.942 ± 0.004**	**0.996 ± 0.001**
RESNET50	BASELINE	0.013 ± 0.002	0.209 ± 0.018	-0.914 ± 0.008	0.944 ± 0.006	0.994 ± 0.001
	TEMP	0.016 ± 0.002	0.205 ± 0.018	-0.914 ± 0.008	0.944 ± 0.006	0.995 ± 0.001
	ISOTONIC	**0.006 ± 0.001**	**0.183 ± 0.013**	**-0.924 ± 0.006**	**0.952 ± 0.004**	0.995 ± 0.001
	SPLINES	0.011 ± 0.003	0.397 ± 0.029	-0.879 ± 0.008	0.925 ± 0.005	0.981 ± 0.002
	PROBE	0.014 ± 0.002	0.191 ± 0.014	-0.919 ± 0.007	0.947 ± 0.005	**0.996 ± 0.000**
WIDE-RESNET-28-10	BASELINE	0.020 ± 0.002	0.196 ± 0.010	-0.923 ± 0.005	0.951 ± 0.004	0.993 ± 0.000
	TEMP	0.009 ± 0.001	0.181 ± 0.011	-0.925 ± 0.005	0.951 ± 0.004	**0.996 ± 0.000**
	ISOTONIC	**0.005 ± 0.001**	0.175 ± 0.009	**-0.928 ± 0.004**	**0.954 ± 0.003**	0.995 ± 0.000
	SPLINES	0.012 ± 0.004	0.421 ± 0.042	-0.878 ± 0.008	0.924 ± 0.005	0.980 ± 0.003
	PROBE	0.009 ± 0.001	**0.174 ± 0.008**	-0.927 ± 0.004	0.953 ± 0.003	**0.996 ± 0.000**

rected it for multiple hypothesis testing using Holm's step down procedure (Demšar, 2006). Our analysis reveals that probe scaling outperforms both vanilla training (baseline) and temperature scaling in ECE, and outperforms all post-hoc methods on all remaining four metrics: NLL, Brier, accuracy, and AUC with a statistically significant evidence at the 95% confidence level. Details are provided in Table 2.

Table 5: Empirical evaluation of post-hoc calibration methods on Fashion MNIST (Xiao et al., 2017) for the five metrics: ECE, negative log-likelihood (NLL), Brier score, accuracy (ACC) and AUC.

		ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
VGG16	BASELINE	0.049 ± 0.007	0.323 ± 0.039	-0.858 ± 0.010	0.910 ± 0.005	0.988 ± 0.003
	TEMP	0.017 ± 0.003	0.274 ± 0.021	-0.866 ± 0.009	0.910 ± 0.005	0.994 ± 0.001
	ISOTONIC	0.016 ± 0.003	0.273 ± 0.018	-0.872 ± 0.008	0.914 ± 0.005	0.993 ± 0.001
	SPLINES	0.016 ± 0.005	0.340 ± 0.077	-0.857 ± 0.028	0.895 ± 0.031	0.991 ± 0.003
	PROBE	**0.010 ± 0.002**	**0.199 ± 0.007**	**-0.898 ± 0.004**	**0.931 ± 0.003**	**0.997 ± 0.000**
VGG19	BASELINE	0.040 ± 0.012	0.298 ± 0.064	-0.864 ± 0.026	0.912 ± 0.015	0.990 ± 0.003
	TEMP	0.015 ± 0.003	**0.267 ± 0.048**	-0.869 ± 0.023	0.912 ± 0.015	**0.994 ± 0.002**
	ISOTONIC	0.016 ± 0.003	0.269 ± 0.028	-0.873 ± 0.014	0.915 ± 0.010	**0.994 ± 0.001**
	SPLINES	**0.014 ± 0.004**	0.320 ± 0.070	-0.862 ± 0.030	0.902 ± 0.031	0.992 ± 0.003
	PROBE	0.023 ± 0.025	0.280 ± 0.178	**-0.889 ± 0.026**	**0.928 ± 0.010**	0.992 ± 0.010
RESNET18	BASELINE	0.053 ± 0.006	0.358 ± 0.043	-0.863 ± 0.011	0.915 ± 0.006	0.986 ± 0.002
	TEMP	**0.014 ± 0.003**	0.261 ± 0.019	-0.875 ± 0.009	0.915 ± 0.006	0.994 ± 0.001
	ISOTONIC	0.020 ± 0.005	0.323 ± 0.030	-0.847 ± 0.016	0.898 ± 0.010	0.991 ± 0.001
	SPLINES	0.030 ± 0.015	0.568 ± 0.200	-0.773 ± 0.072	0.812 ± 0.063	0.975 ± 0.017
	PROBE	**0.014 ± 0.004**	**0.210 ± 0.005**	**-0.897 ± 0.003**	**0.931 ± 0.002**	**0.996 ± 0.000**
RESNET50	BASELINE	0.064 ± 0.006	0.433 ± 0.021	-0.823 ± 0.008	0.887 ± 0.004	0.984 ± 0.001
	TEMP	0.014 ± 0.002	0.331 ± 0.012	-0.836 ± 0.007	0.887 ± 0.004	0.992 ± 0.001
	ISOTONIC	0.017 ± 0.004	0.318 ± 0.017	-0.845 ± 0.008	0.896 ± 0.004	0.992 ± 0.001
	SPLINES	0.026 ± 0.014	0.480 ± 0.160	-0.801 ± 0.057	0.849 ± 0.054	0.982 ± 0.012
	PROBE	**0.012 ± 0.002**	**0.282 ± 0.021**	**-0.859 ± 0.010**	**0.904 ± 0.006**	**0.994 ± 0.001**
WIDE-RESNET-28-10	BASELINE	0.043 ± 0.004	0.300 ± 0.014	-0.873 ± 0.005	0.920 ± 0.003	0.989 ± 0.001
	TEMP	0.013 ± 0.002	0.252 ± 0.009	-0.879 ± 0.004	0.920 ± 0.003	0.994 ± 0.001
	ISOTONIC	0.013 ± 0.002	**0.242 ± 0.008**	**-0.886 ± 0.003**	**0.925 ± 0.003**	0.994 ± 0.001
	SPLINES	0.013 ± 0.002	0.270 ± 0.011	-0.884 ± 0.003	0.923 ± 0.003	0.994 ± 0.001
	PROBE	**0.010 ± 0.001**	**0.242 ± 0.007**	-0.882 ± 0.003	0.922 ± 0.002	**0.995 ± 0.000**

Table 6: Empirical evaluation on ResNet50 trained on ImageNet ILSVRC2012 (Deng et al., 2009) and ImageNet-C (Hendrycks & Dietterich, 2019). Here, the splines-based method is excluded from the comparison because it failed to perform better than the baseline.

		ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
ImageNet	BASELINE	0.0640	1.1481	-0.6173	0.7276	0.6659
	ISOTONIC	0.0447	1.3849	-0.5978	0.7093	**0.6947**
	TEMP	0.0247	**1.1066**	**-0.6220**	**0.7276**	0.6735
	PROBE	**0.0239**	**1.1065**	**-0.6220**	0.7272	0.6736
ImageNet-C	BASELINE	0.0846	1.6544	-0.5005	**0.6316**	0.6548
	ISOTONIC	0.0402	1.8061	-0.4965	0.6249	**0.6858**
	TEMP	0.0271	**1.5903**	**-0.5102**	**0.6316**	0.6646
	PROBE	**0.0259**	**1.5903**	**-0.5101**	**0.6316**	0.6646

4.2 Complementing Methods Results

Table 7 presents the empirical results when combining Monte Carlo dropout with probe scaling. We evaluate on the Uncertainty Baselines Benchmark (Nado et al., 2021) as mentioned earlier, which uses the Wide-ResNet-28-10 architecture. The results indicate that in most of the cases, probe scaling improves the calibration and uncertainty estimation of the model when combined with Monte Carlo dropout, sometimes quite significantly such as in the Fashion MNIST dataset. Out of the four datasets, only SVHN does not reveal a notable gain by combining probe scaling with Monte Carlo dropout.

4.3 Contribution of Probes.

In Figure 3, we plot the contribution of the probes as a function of their relative depths for models trained on CIFAR100. One stark difference appears between the ResNet architectures with skip connections and the classical VGG architectures. In VGG16/19, probes assigned to the *early* layers in the neural network play a significant role in estimating the model's uncertainty. In ResNet architectures, on the other hand, only the probes assigned to the upper layers play a significant role.

Table 7: Empirical evaluation on Wide-ResNet-28-10 with Monte Carlo dropout. Probe scaling can complement other techniques to improve calibration further.

			ECE ↓	NLL ↓	Brier ↓	ACC ↑	AUC ↑
			CIFAR10				
Dropout rate	0.1	MC Dropout	0.023 ± 0.001	0.160 ± 0.004	-0.932 ± 0.001	0.958 ± 0.001	0.995 ± 0.000
		Probes + MC Dropout	**0.010 ± 0.001**	**0.144 ± 0.003**	**-0.935 ± 0.001**	0.957 ± 0.001	**0.997 ± 0.000**
Dropout rate	0.2	MC Dropout	0.019 ± 0.001	0.151 ± 0.002	-0.929 ±0.001	0.954 ± 0.001	0.996 ± 0.000
		Probes + MC Dropout	**0.005 ± 0.000**	**0.144 ± 0.001**	-0.930 ± 0.000	0.953 ± 0.001	**0.998 ± 0.000**
			CIFAR100				
Dropout rate	0.1	MC Dropout	0.044 ± 0.002	0.754 ± 0.002	-0.717 ± 0.002	0.802 ± 0.002	0.985 ± 0.000
		Probes + MC Dropout	**0.033 ± 0.002**	**0.751 ± 0.002**	**-0.719 ± 0.002**	0.802 ± 0.002	**0.988 ± 0.000**
Dropout rate	0.2	MC Dropout	0.035 ± 0.003	0.709 ± 0.005	-0.718 ± 0.002	0.798 ± 0.002	0.988 ± 0.000
		Probes + MC Dropout	**0.016 ± 0.002**	**0.716 ± 0.004**	-0.716 ± 0.002	**0.794 ± 0.002**	**0.991 ± 0.001**
			SVHN				
Dropout rate	0.1	MC Dropout	**0.003 ± 0.001**	0.166 ± 0.010	-0.929 ± 0.005	0.953 ± 0.004	0.996 ± 0.000
		Probes + MC Dropout	0.009 ± 0.001	0.165 ± 0.010	-0.930 ± 0.005	0.954 ± 0.004	0.996 ± 0.000
Dropout rate	0.2	MC Dropout	0.016 ± 0.004	0.168 ± 0.007	-0.929 ± 0.004	0.955 ± 0.003	0.997 ± 0.000
		Probes + MC Dropout	0.014 ± 0.002	0.166 ± 0.008	-0.930 ± 0.004	0.955 ± 0.003	0.997 ± 0.000
			FASHION MNIST				
Dropout rate	0.1	MC Dropout	0.038 ± 0.005	0.295 ± 0.033	-0.857 ± 0.015	0.905 ± 0.010	0.992 ± 0.001
		Probes + MC Dropout	**0.011 ± 0.001**	**0.237 ± 0.015**	**-0.881 ± 0.006**	**0.918 ± 0.004**	**0.996 ± 0.001**
Dropout rate	0.2	MC Dropout	0.018 ± 0.003	0.233 ± 0.012	-0.879 ± 0.005	0.917 ± 0.004	0.996 ± 0.001
		Probes + MC Dropout	**0.009 ± 0.001**	**0.212 ± 0.006**	**-0.890 ± 0.003**	**0.923 ± 0.002**	0.996 ± 0.000

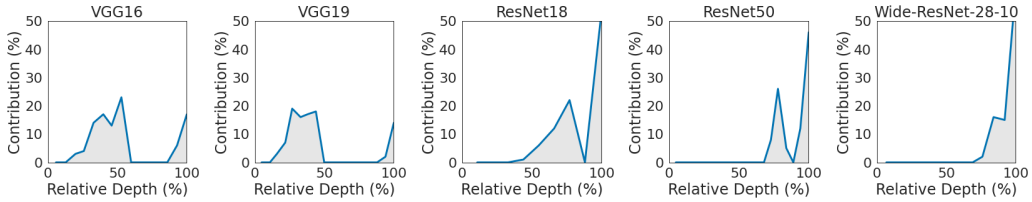

Figure 3: The relative contribution of each probe is plotted as a function of its relative depth. In classical VGG architectures with no skip connections, probes assigned to the early layers play a significant role in estimating the model's uncertainty. This is not the case in ResNet architectures with skip connections, where only the probes assigned to the upper layers are used for calibration.

5 Conclusion

In this paper, we develop "probe scaling", a simple, effective post-hoc technique to improve the calibration of deep neural networks. We empirically demonstrate that using the representations learned by the intermediate layers of a neural network can be utilized to estimate its uncertainty. Besides the theoretical argument for the advantage of probe scaling over temperature scaling, we conduct an empirical evaluation against several post-hoc calibration methods across several architectures and show that probe scaling outperforms other post-hoc methods in standard calibration metrics, such as the Brier score and the negative log-likelihood with statistically significant evidence. In addition, probe scaling is the only post-hoc method that exhibits a consistent improvement over the baseline in both accuracy and AUC. Finally, we show that probe scaling can complement other techniques, such as Monte Carlo dropout, to improve calibration further.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker,

Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Ibrahim Alabdulmohsin and Mario Lucic. A near-optimal algorithm for debiasing trained machine learning models. *arXiv preprint arXiv:2106.12887*, 2021.

Devansh Arpit, Stanisław Jastrzkebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, 2017.

Robert JN Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *arXiv preprint arXiv:2106.09647*, 2021.

J Eric Bickel. Some comparisons among quadratic, spherical, and logarithmic scoring rules. *Decision Analysis*, 4(2):49–65, 2007.

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

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *FAccT*, 2019.

François Chollet et al. Keras. `https://keras.io`, 2015.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. DNN or k-NN: That is the generalize vs. memorize question. *arXiv:1805.06822*, 2018.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *SIGKDD*, 2017.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*. PMLR, 2016.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *NeurIPS*, 2017.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

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

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *ICLR*, 2021.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *ICLR*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

Benjamin Kompa, Jasper Snoek, and Andrew L Beam. Second opinion needed: communicating uncertainty in medical machine learning. *NPJ Digital Medicine*, 4(1):1–6, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, 2019.

Aviral Kumar and Sunita Sarawagi. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*, 2019.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

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

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *NeurIPS*, 2019.

Hartmut Maennel, Ibrahim Alabdulmohsin, Ilya Tolstikhin, Robert JN Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. What do neural networks learn when trained with random labels? In *NeurIPS*, 2020.

Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *FAccT*, 2018.

Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2901–2907. AAAI Press, 2015. ISBN 0262511290.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *NeurIPS*, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *SIGKDD*, 2002.

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

A Proof of Proposition 1

The proof of this proposition uses Corollary 13.8 in Shalev-Shwartz & Ben-David (2014). Let $L_S(w)$ be the empirical loss w.r.t. f and write $L_{\mathcal{D}}(w)$ for the population loss of w. If f is a convex function that is ρ-Lipschitz continuous, then the solution to the regularized empirical risk minimization problem:

$$\hat{\mathbf{w}} = \arg\min_{w} \big\{\lambda||w||^2 + L_S(w)\big\},$$

satisfies the oracle inequality:

$$\forall \mathbf{w}: \quad \mathbb{E}_S\big[L_{\mathcal{D}}(\hat{\mathbf{w}}\big] \leq L_{\mathcal{D}}(\mathbf{w}) + \lambda||\mathbf{w}||^2 + \frac{2\rho^2}{\lambda N}.$$

Choosing $\mathbf{w} = \mathbf{w}^\star \doteq \arg\min_w\{L_{\mathcal{D}}(w)\}$:

$$\mathbb{E}_S\big[L_{\mathcal{D}}(\hat{\mathbf{w}}\big] \leq L_{\mathcal{D}}(\mathbf{w}^\star) + \lambda||\mathbf{w}^\star||^2 + \frac{2\rho^2}{\lambda N}.$$

Using the definition of $\mathbf{w}^\star$ and applying Markov's inequality, we have for any $\mathbf{w}$:

$$\begin{aligned} p_S\Big\{L_{\mathcal{D}}(\hat{\mathbf{w}}) - L_{\mathcal{D}}(\mathbf{w}) \geq \epsilon\Big\} &\leq p_S\Big\{L_{\mathcal{D}}(\hat{\mathbf{w}}) - L_{\mathcal{D}}(\mathbf{w}^\star) \geq \epsilon\Big\} \\ &\leq \frac{1}{\epsilon}\Big(\lambda\,||\mathbf{w}^\star||^2 + \frac{2\rho^2}{\lambda N}\Big) \end{aligned}$$

Setting $\hat{\mathbf{w}}$ for the solution learned by probe scaling and $\mathbf{w}$ for temperature scaling (which is valid because temperature scaling belongs to the search space of probe scaling), we obtain the statement of the proposition.