scores to determine if samples are out-of-distribution.

Bayesian neural networks (Denker & Lecun, 1990; MacKay, 1992) return a probability distribution over outputs as an alternative way to represent model uncertainty. Gal & Ghahramani (2016) draw a connection between Dropout (Srivastava et al., 2014) and model uncertainty, claiming that sampling models with dropped nodes is a way to estimate the probability distribution over all possible models for a given sample. Kendall & Gal (2017) combine this approach with a model that outputs a predictive mean and variance for each data point. This notion of uncertainty is not restricted to classification problems. Additionally, neural networks can be used in conjunction with Bayesian models that output complete distributions. For example, deep kernel learning (Wilson et al., 2016a;b; Al-Shedivat et al., 2016) combines deep neural networks with Gaussian processes on classification and regression problems. In contrast, our framework, which does not augment the neural network model, returns a confidence score rather than returning a distribution of possible outputs.

5. Results

We apply the calibration methods in Section 4 to image classification and document classification neural networks. For image classification we use 6 datasets:

1. Caltech-UCSD Birds (Welinder et al., 2010): 200 bird species. 5994/2897/2897 images for train/validation/test sets.
2. Stanford Cars (Krause et al., 2013): 196 classes of cars by make, model, and year. 8041/4020/4020 images for train/validation/test.
3. ImageNet 2012 (Deng et al., 2009): Natural scene images from 1000 classes. 1.3 million/25,000/25,000 images for train/validation/test.
4. CIFAR-10/CIFAR-100 (Krizhevsky & Hinton, 2009): Color images (32×32) from 10/100 classes. 45,000/5,000/10,000 images for train/validation/test.
5. Street View House Numbers (SVHN) (Netzer et al., 2011): 32×32 colored images of cropped out house numbers from Google Street View. 598,388/6,000/26,032 images for train/validation/test.

We train state-of-the-art convolutional networks: ResNets (He et al., 2016), ResNets with stochastic depth (SD) (Huang et al., 2016), Wide ResNets (Zagoruyko & Komodakis, 2016), and DenseNets (Huang et al., 2017). We use the data preprocessing, training procedures, and hyperparameters as described in each paper. For Birds and Cars, we fine-tune networks pretrained on ImageNet.

For document classification we experiment with 4 datasets:

1. 20 News: News articles, partitioned into 20 categories by content. 9034/2259/7528 documents for train/validation/test.
2. Reuters: News articles, partitioned into 8 categories by topic. 4388/1097/2189 documents for train/validation/test.
3. Stanford Sentiment Treebank (SST) (Socher et al., 2013): Movie reviews, represented as sentence parse trees that are annotated by sentiment. Each sample includes a coarse binary label and a fine grained 5-class label. As described in (Tai et al., 2015), the training/validation/test sets contain 6920/872/1821 documents for binary, and 544/1101/2210 for fine-grained.

On 20 News and Reuters, we train Deep Averaging Networks (DANs) (Iyyer et al., 2015) with 3 feed-forward layers and Batch Normalization. On SST, we train TreeLSTMs (Long Short Term Memory) (Tai et al., 2015). For both models we use the default hyperparmaeters suggested by the authors.

Calibration Results. Table 1 displays model calibration, as measured by ECE (with $M = 15$ bins), before and after applying the various methods (see Section S3 for MCE, NLL, and error tables). It is worth noting that most datasets and models experience some degree of miscalibration, with ECE typically between 4 to 10%. This is not architecture specific: we observe miscalibration on convolutional networks (with and without skip connections), recurrent networks, and deep averaging networks. The two notable exceptions are SVHN and Reuters, both of which experience ECE values below 1%. Both of these datasets have very low error (1.98% and 2.97%, respectively); and therefore the ratio of ECE to error is comparable to other datasets.

Our most important discovery is the surprising effectiveness of temperature scaling despite its remarkable simplicity. Temperature scaling outperforms all other methods on the vision tasks, and performs comparably to other methods on the NLP datasets. What is perhaps even more surprising is that temperature scaling outperforms the vector and matrix Platt scaling variants, which are strictly more general methods. In fact, vector scaling recovers essentially the same solution as temperature scaling – the learned vector has nearly constant entries, and therefore is no different than a scalar transformation. In other words, network miscalibration is intrinsically low dimensional.

The only dataset that temperature scaling does not calibrate is the Reuters dataset. In this instance, only one of the above methods is able to improve calibration. Because this dataset is well-calibrated to begin with (ECE ≤ 1%), there is not much room for improvement with any method, and post-processing may not even be necessary to begin with. It is also possible that our measurements are affected by dataset split or by the particular binning scheme.

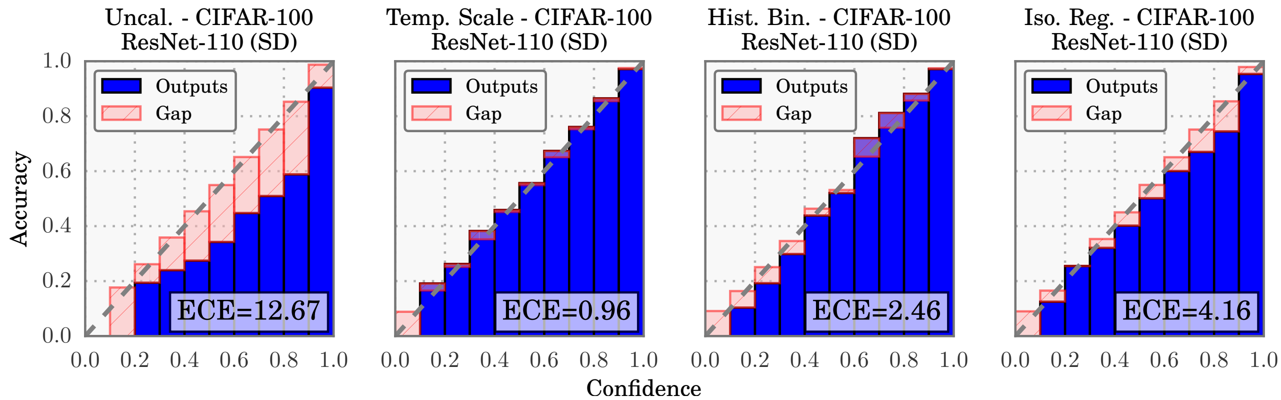

Figure 4. Reliability diagrams for CIFAR-100 before (far left) and after calibration (middle left, middle right, far right).

Matrix scaling performs poorly on datasets with hundreds of classes (i.e. Birds, Cars, and CIFAR-100), and fails to converge on the 1000-class ImageNet dataset. This is expected, since the number of parameters scales quadratically with the number of classes. Any calibration model with tens of thousands (or more) parameters will overfit to a small validation set, even when applying regularization.

Binning methods improve calibration on most datasets, but do not outperform temperature scaling. Additionally, binning methods tend to change class predictions which hurts accuracy (see Section S3). Histogram binning, the simplest binning method, typically outperforms isotonic regression and BBQ, despite the fact that both methods are strictly more general. This further supports our finding that calibration is best corrected by simple models.

Reliability diagrams. Figure 4 contains reliability diagrams for 110-layer ResNets on CIFAR-100 before and after calibration. From the far left diagram, we see that the uncalibrated ResNet tends to be overconfident in its predictions. We then can observe the effects of temperature scaling (middle left), histogram binning (middle right), and isotonic regression (far right) on calibration. All three displayed methods produce much better confidence estimates. Of the three methods, temperature scaling most closely recovers the desired diagonal function. Each of the bins are well calibrated, which is remarkable given that all the probabilities were modified by only a single parameter. We include reliability diagrams for other datasets in Section S4.

Computation time. All methods scale linearly with the number of validation set samples. Temperature scaling is by far the fastest method, as it amounts to a one-dimensional convex optimization problem. Using a conjugate gradient solver, the optimal temperature can be found in 10 iterations, or a fraction of a second on most modern hardware. In fact, even a naive line-search for the optimal temperature is faster than any of the other methods. The computational complexity of vector and matrix scaling are linear and quadratic respectively in the number of classes, reflecting the number of parameters in each method. For CIFAR-100 ($K = 100$), finding a near-optimal vector scaling solution with conjugate gradient descent requires at least 2 orders of magnitude more time. Histogram binning and isotonic regression take an order of magnitude longer than temperature scaling, and BBQ takes roughly 3 orders of magnitude more time.

Ease of implementation. BBQ is arguably the most difficult to implement, as it requires implementing a model averaging scheme. While all other methods are relatively easy to implement, temperature scaling may arguably be the most straightforward to incorporate into a neural network pipeline. In Torch7 (Collobert et al., 2011), for example, we implement temperature scaling by inserting a `nn.MulConstant` between the logits and the softmax, whose parameter is $1/T$. We set $T = 1$ during training, and subsequently find its optimal value on the validation set.[4]

6. Conclusion

Modern neural networks exhibit a strange phenomenon: probabilistic error and miscalibration worsen even as classification error is reduced. We have demonstrated that recent advances in neural network architecture and training – model capacity, normalization, and regularization – have strong effects on network calibration. It remains future work to understand why these trends affect calibration while improving accuracy. Nevertheless, simple techniques can effectively remedy the miscalibration phenomenon in neural networks. Temperature scaling is the simplest, fastest, and most straightforward of the methods, and surprisingly is often the most effective.

[4] For an example implementation, see `http://github.com/gpleiss/temperature_scaling`.

Acknowledgments

The authors are supported in part by the III-1618134, III-1526012, and IIS-1149882 grants from the National Science Foundation, as well as the Bill and Melinda Gates Foundation and the Office of Naval Research.

References

Al-Shedivat, Maruan, Wilson, Andrew Gordon, Saatchi, Yunus, Hu, Zhiting, and Xing, Eric P. Learning scalable deep kernels with recurrent structure. *arXiv preprint arXiv:1610.08936*, 2016.

Bengio, Yoshua, Goodfellow, Ian J, and Courville, Aaron. Deep learning. *Nature*, 521:436–444, 2015.

Bojarski, Mariusz, Del Testa, Davide, Dworakowski, Daniel, Firner, Bernhard, Flepp, Beat, Goyal, Prasoon, Jackel, Lawrence D, Monfort, Mathew, Muller, Urs, Zhang, Jiakai, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Caruana, Rich, Lou, Yin, Gehrke, Johannes, Koch, Paul, Sturm, Marc, and Elhadad, Noemie. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, 2015.

Collobert, Ronan, Kavukcuoglu, Koray, and Farabet, Clément. Torch7: A matlab-like environment for machine learning. In *BigLearn Workshop, NIPS*, 2011.

Cosmides, Leda and Tooby, John. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *cognition*, 58(1):1–73, 1996.

DeGroot, Morris H and Fienberg, Stephen E. The comparison and evaluation of forecasters. *The statistician*, pp. 12–22, 1983.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Denker, John S and Lecun, Yann. Transforming neural-net output levels to probability distributions. In *NIPS*, pp. 853–859, 1990.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

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

Girshick, Ross. Fast r-cnn. In *ICCV*, pp. 1440–1448, 2015.

Hannun, Awni, Case, Carl, Casper, Jared, Catanzaro, Bryan, Diamos, Greg, Elsen, Erich, Prenger, Ryan, Satheesh, Sanjeev, Sengupta, Shubho, Coates, Adam, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

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

Hendrycks, Dan and Gimpel, Kevin. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. Distilling the knowledge in a neural network. 2015.

Huang, Gao, Sun, Yu, Liu, Zhuang, Sedra, Daniel, and Weinberger, Kilian. Deep networks with stochastic depth. In *ECCV*, 2016.

Huang, Gao, Liu, Zhuang, Weinberger, Kilian Q, and van der Maaten, Laurens. Densely connected convolutional networks. In *CVPR*, 2017.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

Iyyer, Mohit, Manjunatha, Varun, Boyd-Graber, Jordan, and Daumé III, Hal. Deep unordered composition rivals syntactic methods for text classification. In *ACL*, 2015.

Jaynes, Edwin T. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Jiang, Xiaoqian, Osl, Melanie, Kim, Jihoon, and Ohno-Machado, Lucila. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, 2012.

Kendall, Alex and Cipolla, Roberto. Modelling uncertainty in deep learning for camera relocalization. 2016.

Kendall, Alex and Gal, Yarin. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

Krause, Jonathan, Stark, Michael, Deng, Jia, and Fei-Fei, Li. 3d object representations for fine-grained categorization. In *IEEE Workshop on 3D Representation and Recognition (3dRR)*, Sydney, Australia, 2013.

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

Kuleshov, Volodymyr and Ermon, Stefano. Reliable confidence estimation via online learning. *arXiv preprint arXiv:1607.03594*, 2016.

Kuleshov, Volodymyr and Liang, Percy. Calibrated structured prediction. In *NIPS*, pp. 3474–3482, 2015.

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

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

MacKay, David JC. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992.

Naeini, Mahdi Pakdaman, Cooper, Gregory F, and Hauskrecht, Milos. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, pp. 2901, 2015.

Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. Reading digits in natural images with unsupervised feature learning. In *Deep Learning and Unsupervised Feature Learning Workshop, NIPS*, 2011.

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

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

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

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

Socher, Richard, Perelygin, Alex, Wu, Jean, Chuang, Jason, Manning, Christopher D., Ng, Andrew, and Potts, Christopher. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pp. 1631–1642, 2013.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Srivastava, Rupesh Kumar, Greff, Klaus, and Schmidhuber, Jürgen. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Tai, Kai Sheng, Socher, Richard, and Manning, Christopher D. Improved semantic representations from tree-structured long short-term memory networks. 2015.

Vapnik, Vladimir N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Wilson, Andrew G, Hu, Zhiting, Salakhutdinov, Ruslan R, and Xing, Eric P. Stochastic variational deep kernel learning. In *NIPS*, pp. 2586–2594, 2016a.

Wilson, Andrew Gordon, Hu, Zhiting, Salakhutdinov, Ruslan, and Xing, Eric P. Deep kernel learning. In *AISTATS*, pp. 370–378, 2016b.

Xiong, Wayne, Droppo, Jasha, Huang, Xuedong, Seide, Frank, Seltzer, Mike, Stolcke, Andreas, Yu, Dong, and Zweig, Geoffrey. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016.

Zadrozny, Bianca and Elkan, Charles. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, pp. 609–616, 2001.

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

Zagoruyko, Sergey and Komodakis, Nikos. Wide residual networks. In *BMVC*, 2016.

Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Supplementary Materials for: On Calibration of Modern Neural Networks

S1. Further Information on Calibration Metrics

We can connect the ECE metric with our exact miscalibration definition, which is restated here:

$$\mathop{\mathbb{E}}_{\hat{P}}\left[\left|\mathbb{P}\left(\hat{Y} = Y \mid \hat{P} = p\right) - p\right|\right]$$

Let $F_{\hat{P}}(p)$ be the cumulative distribution function of $\hat{P}$ so that $F_{\hat{P}}(b) - F_{\hat{P}}(a) = \mathbb{P}(\hat{P} \in [a, b])$. Using the Riemann-Stieltjes integral we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\hat{P}}\left[\left|\mathbb{P}\left(\hat{Y} = Y \mid \hat{P} = p\right) - p\right|\right] \\
&= \int_0^1 \left|\mathbb{P}\left(\hat{Y} = Y \mid \hat{P} = p\right) - p\right| dF_{\hat{P}}(p) \\
&\approx \sum_{m=1}^{M} \left|\mathbb{P}(\hat{Y} = Y \mid \hat{P} = p_m) - p_m\right| \mathbb{P}(\hat{P} \in I_m)
\end{aligned}$$

where I_m represents the interval of bin B_m. $\left|\mathbb{P}(\hat{Y} = Y \mid \hat{P} = p_m) - p_m\right|$ is closely approximated by $|\mathrm{acc}(B_m) - \hat{p}(B_m)|$ for n large. Hence ECE using M bins converges to the M-term Riemann-Stieltjes sum of $\mathbb{E}_{\hat{P}}\left[\left|\mathbb{P}\left(\hat{Y} = Y \mid \hat{P} = p\right) - p\right|\right]$.

S2. Further Information on Temperature Scaling

Here we derive the temperature scaling model using the entropy maximization principle with an appropriate balanced equation.

Claim 1. *Given n samples' logit vectors $\mathbf{z}_1, \ldots, \mathbf{z}_n$ and class labels $y_1, \ldots, y_n$, temperature scaling is the unique solution q to the following entropy maximization problem:*

$$\begin{aligned}
\max_{q} \quad & -\sum_{i=1}^{n} \sum_{k=1}^{K} q(\mathbf{z}_i)^{(k)} \log q(\mathbf{z}_i)^{(k)} \\
\text{subject to} \quad & q(\mathbf{z}_i)^{(k)} \geq 0 \quad \forall i, k \\
& \sum_{k=1}^{K} q(\mathbf{z}_i)^{(k)} = 1 \quad \forall i \\
& \sum_{i=1}^{n} z_i^{(y_i)} = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^{(k)} q(\mathbf{z}_i)^{(k)}.
\end{aligned}$$

The first two constraint ensure that q is a probability distribution, while the last constraint limits the scope of distributions. Intuitively, the constraint specifies that the average true class logit is equal to the average weighted logit.

Proof. We solve this constrained optimization problem using the Lagrangian. We first ignore the constraint $q(\mathbf{z}_i)^{(k)}$ and later show that the solution satisfies this condition. Let $\lambda, \beta_1, \ldots, \beta_n \in \mathbb{R}$ be the Lagrangian multipliers and define

$$\begin{aligned}
L = & -\sum_{i=1}^{n} \sum_{k=1}^{K} q(\mathbf{z}_i)^{(k)} \log q(\mathbf{z}_i)^{(k)} \\
& + \lambda \sum_{i=1}^{n} \left[\sum_{k=1}^{K} z_i^{(k)} q(\mathbf{z}_i)^{(k)} - z_i^{(y_i)}\right] \\
& + \sum_{i=1}^{n} \beta_i \sum_{k=1}^{K} \left(q(\mathbf{z}_i)^{(k)} - 1\right).
\end{aligned}$$

Taking the derivative with respect to $q(\mathbf{z}_i)^{(k)}$ gives

$$\frac{\partial}{\partial q(\mathbf{z}_i)^{(k)}} L = -nK - \log q(\mathbf{z}_i)^{(k)} + \lambda z_i^{(k)} + \beta_i.$$

Setting the gradient of the Lagrangian L to 0 and rearranging gives

$$q(\mathbf{z}_i)^{(k)} = e^{\lambda z_i^{(k)} + \beta_i - nK}.$$

Since $\sum_{k=1}^{K} q(\mathbf{z}_i)^{(k)} = 1$ for all i, we must have

$$q(\mathbf{z}_i)^{(k)} = \frac{e^{\lambda z_i^{(k)}}}{\sum_{j=1}^{K} e^{\lambda z_i^{(j)}}},$$

which recovers the temperature scaling model by setting $T = \frac{1}{\lambda}$. □

Figure S1 visualizes Claim 1. We see that, as training continues, the model begins to overfit with respect to NLL (red line). This results in a low-entropy softmax distribution over classes (blue line), which explains the model's overconfidence. Temperature scaling not only lowers the NLL but also raises the entropy of the distribution (green line).

S3. Additional Tables

Tables S1, S2, and S3 display the MCE, test error, and NLL for all the experimental settings outlined in Section 5.

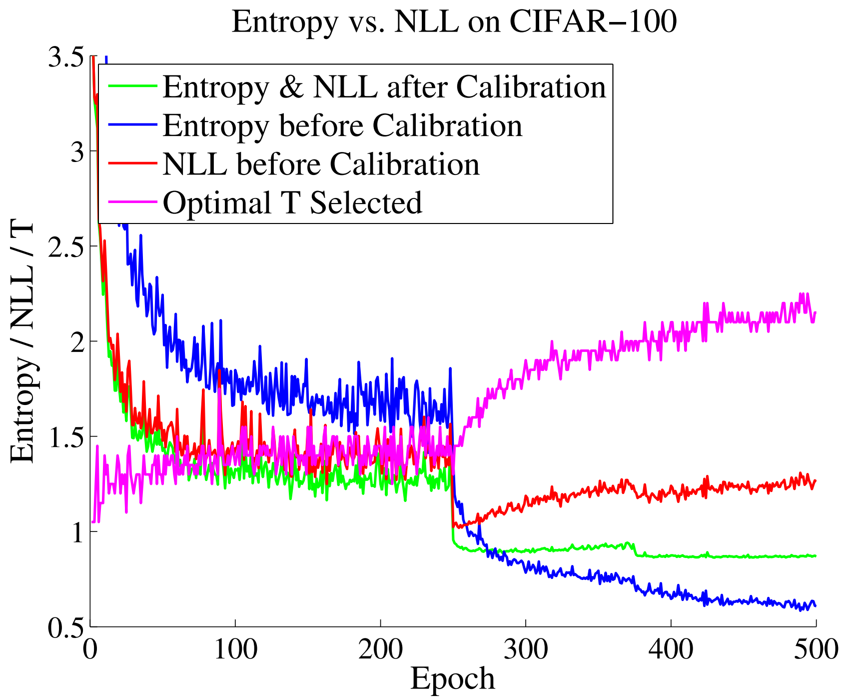

Figure S1. Entropy and NLL for CIFAR-100 before and after calibration. The optimal T selected by temperature scaling rises throughout optimization, as the pre-calibration entropy decreases steadily. The post-calibration entropy and NLL on the validation set coincide (which can be derived from the gradient optimality condition of T).

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	30.06%	25.35%	16.59%	11.72%	9.08%	9.81%	38.67%
Cars	ResNet 50	41.55%	5.16%	15.23%	9.31%	20.23%	8.59%	29.65%
CIFAR-10	ResNet 110	33.78%	26.87%	7.8%	72.64%	8.56%	27.39%	22.89%
CIFAR-10	ResNet 110 (SD)	34.52%	17.0%	16.45%	19.26%	15.45%	15.55%	10.74%
CIFAR-10	Wide ResNet 32	27.97%	12.19%	6.19%	9.22%	9.11%	4.43%	9.65%
CIFAR-10	DenseNet 40	22.44%	7.77%	19.54%	14.57%	4.58%	3.17%	4.36%
CIFAR-10	LeNet 5	8.02%	16.49%	18.34%	82.35%	5.14%	19.39%	16.89%
CIFAR-100	ResNet 110	35.5%	7.03%	10.36%	10.9%	4.74%	2.5%	45.62%
CIFAR-100	ResNet 110 (SD)	26.42%	9.12%	10.95%	9.12%	8.85%	8.85%	35.6%
CIFAR-100	Wide ResNet 32	33.11%	6.22%	14.87%	11.88%	5.33%	6.31%	44.73%
CIFAR-100	DenseNet 40	21.52%	9.36%	10.59%	8.67%	19.4%	8.82%	38.64%
CIFAR-100	LeNet 5	10.25%	18.61%	3.64%	9.96%	5.22%	8.65%	18.77%
ImageNet	DenseNet 161	14.07%	13.14%	11.57%	10.96%	12.29%	9.61%	-
ImageNet	ResNet 152	12.2%	14.57%	8.74%	8.85%	12.29%	9.61%	-
SVHN	ResNet 152 (SD)	19.36%	11.16%	18.67%	9.09%	18.05%	30.78%	18.76%
20 News	DAN 3	17.03%	10.47%	9.13%	6.28%	8.21%	8.24%	17.43%
Reuters	DAN 3	14.01%	16.78%	44.95%	36.18%	25.46%	18.88%	19.39%
SST Binary	TreeLSTM	21.66%	3.22%	13.91%	36.43%	6.03%	6.03%	6.03%
SST Fine Grained	TreeLSTM	27.85%	28.35%	19.0%	8.67%	44.75%	11.47%	11.78%

Table S1. MCE (%) (with $M = 15$ bins) on standard vision and NLP datasets before calibration and with various calibration methods. The number following a model's name denotes the network depth. MCE seems very sensitive to the binning scheme and is less suited for small test sets.

S4. Additional Reliability Diagrams

We include reliability diagrams for additional datasets: CIFAR-10 (Figure S2) and SST (Figure S3 and Figure S4). Note that, as mentioned in Section 2, the reliability diagrams do not represent the proportion of predictions that belong to a given bin.

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	22.54%	55.02%	23.37%	37.76%	22.54%	22.99%	29.51%
Cars	ResNet 50	14.28%	16.24%	14.9%	19.25%	14.28%	14.15%	17.98%
CIFAR-10	ResNet 110	6.21%	6.45%	6.36%	6.25%	6.21%	6.37%	6.42%
CIFAR-10	ResNet 110 (SD)	5.64%	5.59%	5.62%	5.55%	5.64%	5.62%	5.69%
CIFAR-10	Wide ResNet 32	6.96%	7.3%	7.01%	7.35%	6.96%	7.1%	7.27%
CIFAR-10	DenseNet 40	5.91%	6.12%	5.96%	6.0%	5.91%	5.96%	6.0%
CIFAR-10	LeNet 5	15.57%	15.63%	15.69%	15.64%	15.57%	15.53%	15.81%
CIFAR-100	ResNet 110	27.83%	34.78%	28.41%	28.56%	27.83%	27.82%	38.77%
CIFAR-100	ResNet 110 (SD)	24.91%	33.78%	25.42%	25.17%	24.91%	24.99%	35.09%
CIFAR-100	Wide ResNet 32	28.0%	34.29%	28.61%	29.08%	28.0%	28.45%	37.4%
CIFAR-100	DenseNet 40	26.45%	34.78%	26.73%	26.4%	26.45%	26.25%	36.14%
CIFAR-100	LeNet 5	44.92%	54.06%	45.77%	46.82%	44.92%	45.53%	52.44%
ImageNet	DenseNet 161	22.57%	48.32%	23.2%	47.58%	22.57%	22.54%	-
ImageNet	ResNet 152	22.31%	48.1%	22.94%	47.6%	22.31%	22.56%	-
SVHN	ResNet 152 (SD)	1.98%	2.06%	2.04%	2.04%	1.98%	2.0%	2.08%
20 News	DAN 3	20.06%	25.12%	20.29%	20.81%	20.06%	19.89%	22.0%
Reuters	DAN 3	2.97%	7.81%	3.52%	3.93%	2.97%	2.83%	3.52%
SST Binary	TreeLSTM	11.81%	12.08%	11.75%	11.26%	11.81%	11.81%	11.81%
SST Fine Grained	TreeLSTM	49.5%	49.91%	48.55%	49.86%	49.5%	49.77%	48.51%

Table S2.Test error (%) on standard vision and NLP datasets before calibration and with various calibration methods. The number following a model's name denotes the network depth. Error with temperature scaling is exactly the same as uncalibrated.

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	0.9786	1.6226	1.4128	1.2539	0.8792	0.9021	2.334
Cars	ResNet 50	0.5488	0.7977	0.8793	0.6986	0.5311	0.5299	1.0206
CIFAR-10	ResNet 110	0.3285	0.2532	0.2237	0.263	0.2102	0.2088	0.2048
CIFAR-10	ResNet 110 (SD)	0.2959	0.2027	0.1867	0.2159	0.1718	0.1709	0.1766
CIFAR-10	Wide ResNet 32	0.3293	0.2778	0.2428	0.2774	0.2283	0.2275	0.2229
CIFAR-10	DenseNet 40	0.2228	0.212	0.1969	0.2087	0.1750	0.1757	0.176
CIFAR-10	LeNet 5	0.4688	0.529	0.4757	0.4984	0.459	0.4568	0.4607
CIFAR-100	ResNet 110	1.4978	1.4379	1.207	1.5466	1.0442	1.0485	2.5637
CIFAR-100	ResNet 110 (SD)	1.1157	1.1985	1.0317	1.1982	0.8613	0.8655	1.8182
CIFAR-100	Wide ResNet 32	1.3434	1.4499	1.2086	1.459	1.0565	1.0648	2.5507
CIFAR-100	DenseNet 40	1.0134	1.2156	1.0615	1.1572	0.9026	0.9011	1.9639
CIFAR-100	LeNet 5	1.6639	2.2574	1.8173	1.9893	1.6560	1.6648	2.1405
ImageNet	DenseNet 161	0.9338	1.4716	1.1912	1.4272	0.8885	0.8879	-
ImageNet	ResNet 152	0.8961	1.4507	1.1859	1.3987	0.8657	0.8742	-
SVHN	ResNet 152 (SD)	0.0842	0.1137	0.095	0.1062	0.0821	0.0844	0.0924
20 News	DAN 3	0.7949	1.0499	0.8968	0.9519	0.7387	0.7296	0.9089
Reuters	DAN 3	0.102	0.2403	0.1475	0.1167	0.0994	0.0990	0.1491
SST Binary	TreeLSTM	0.3367	0.2842	0.2908	0.2778	0.2739	0.2739	0.2739
SST Fine Grained	TreeLSTM	1.1475	1.1717	1.1661	1.149	1.1168	1.1085	1.1112

Table S3.NLL (%) on standard vision and NLP datasets before calibration and with various calibration methods. The number following a model's name denotes the network depth. To summarize, NLL roughly follows the trends of ECE.

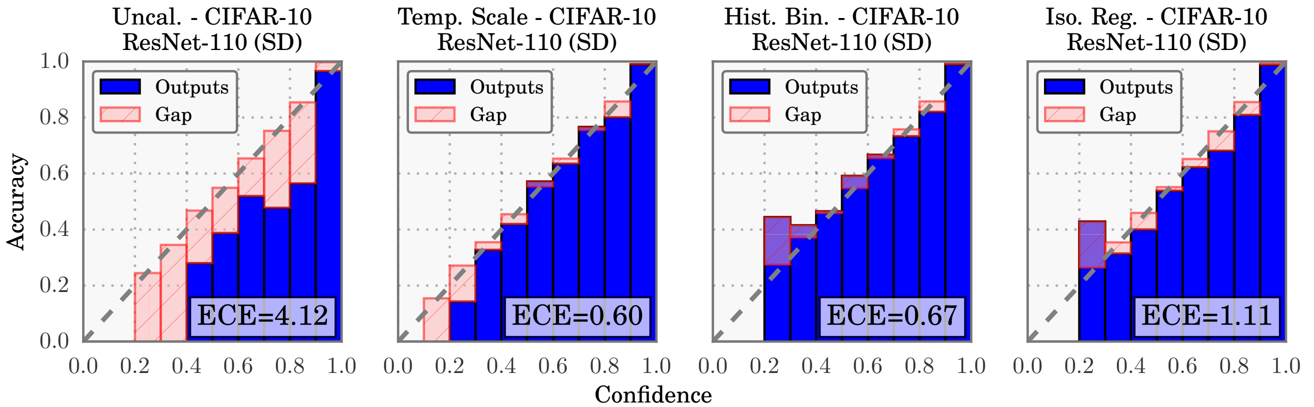

Figure S2.Reliability diagrams for CIFAR-10 before (far left) and after calibration (middle left, middle right, far right).

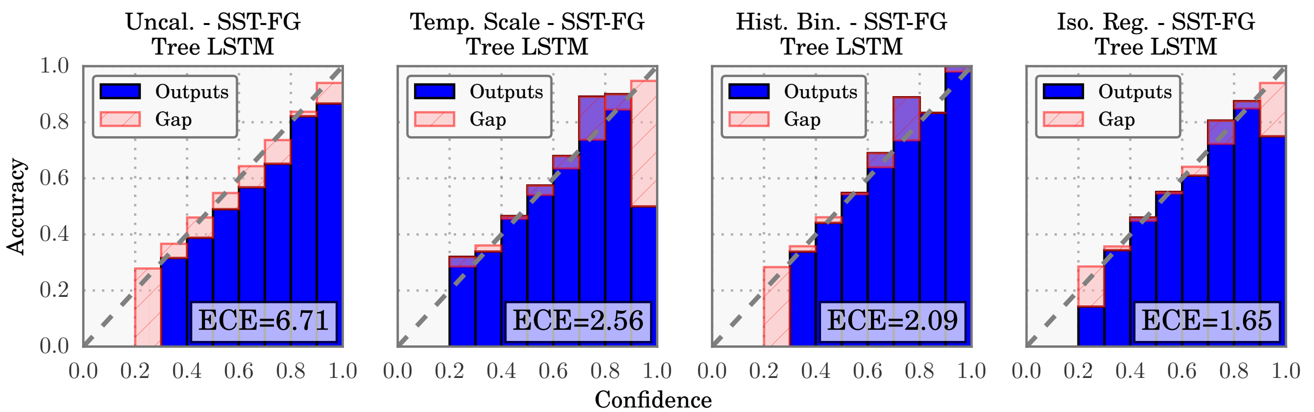

Figure S3.Reliability diagrams for SST Binary and SST Fine Grained before (far left) and after calibration (middle left, middle right, far right).

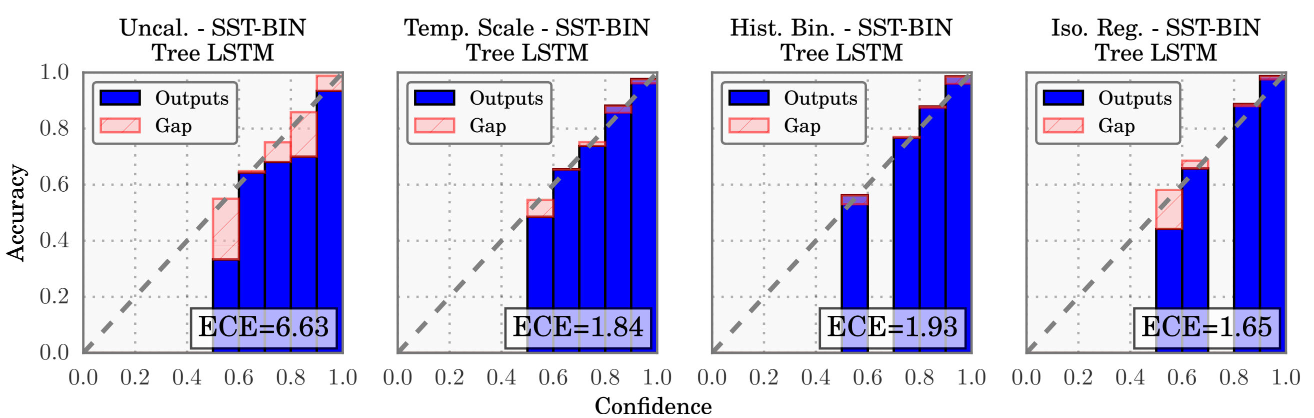

Figure S4.Reliability diagrams for SST Binary and SST Fine Grained before (far left) and after calibration (middle left, middle right, far right).