Measuring Calibration in Deep Learning

Abstract

The reliability of a machine learning model's confidence in its predictions is critical for high-risk applications. Calibration—the idea that a model's predicted probabilities reflect true probabilities—formalizes this notion. While analyzing the calibration of deep neural networks, we've identified core problems with the way calibration is currently measured. For example, trained networks often predict a class label with very high confidence. This causes existing metrics to measure calibration error only within a small probability interval, which ultimately leads to misleading conclusions about whether a model is well-calibrated. The Thresholded Adaptive Calibration Error (TACE) metric is designed to resolve these pathologies. We show that while ECE becomes a worse approximation of true calibration error as class predictions beyond the maximum prediction matter more, TACE continues to perform well.

1 Introduction

The reliability of a machine learning model's confidence in its predictions is critical for high risk applications, such as deciding whether to trust a medical diagnosis prediction (Crowson et al., 2016; Jiang et al., 2011; Raghu et al., 2018). One mathematical formulation of the reliability of confidence is calibration (Murphy and Epstein, 1967; Dawid, 1982). Intuitively, for class predictions, calibration means that if a model assigns a class with 90% probability, that class should appear 90% of the time.

Recent work proposed Expected Calibration Error (ECE; Naeini et al., 2015), a measure of calibration error which has lead to a surge of works developing methods for calibrated deep neural networks (e.g., Guo et al., 2017; Kuleshov et al., 2018). We show that ECE has numerous pathologies, and that recent calibration methods, which have been shown to successfully recalibrate models according to ECE, do not lead to truly calibrated models. We examine these pathologies and propose the Thresholded Adaptive Calibration Error (TACE) metric, which is designed to resolve them. We also propose a dynamic programming based metric for estimating the calibration error of specific predictions.

2 Background & Related Work

Assume the dataset of features and outcomes $\{(x, y)\}$ are i.i.d. realizations of the random variables $X, Y \sim \mathbb{P}$. We focus on class predictions. Suppose a model predicts a class y with probability $\hat{p}$. The model is *calibrated* if $\hat{p}$ is always the true probability. Formally,

$$\mathbb{P}(Y = y \mid \hat{p} = p) = p$$

for all probability values $p \in [0, 1]$ and class labels $y \in \{0, \ldots, K-1\}$. The left-hand-side denotes the true data distribution's probability of a label given that the model predicts $\hat{p} = p$; the right-hand-side denotes that value.

Expected Calibration Error (ECE). To measure the deviation from calibration, ECE discretizes the probability interval into a fixed number of bins and assigns predicted probabilities to the bin that encompasses it. The calibration error is the difference between the fraction of predictions in the bin that are correct (accuracy) and the mean of the probabilities in the bin (confidence). Intuitively, the accuracy estimates $\mathbb{P}(Y = y \mid \hat{p} = p)$ and the average confidence is a setting of p. ECE computes a weighted average of this error across bins:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|,$$

where n_b is the number of predictions in bin b, N is the total number of data points, and $acc(b)$ and $conf(b)$ are the accuracy and confidence of bin b, respectively. ECE as framed in Naeini et al. (2015) leaves ambiguity in both its binning implementation and how to compute calibration for multiple classes. In Guo et al. (2017), they bin the probability interval $[0, 1]$ into equally spaced subintervals, and they take the maximum probability output for each datapoint (i.e., the predicted class's probability). We use this for our ECE implementation.

Other measures of probabilistic accuracy. Many classic methods exist to measure the accuracy of predicted probabilities. For example, Brier score measures the mean squared difference between the predicted probability and the actual outcome (Gneiting and Raftery, 2007). This score can be shown to decompose into a sum of metrics, including calibration error. The Hosmer-Lemeshow test is a popular hypothesis test for assessing whether a model's predictions significantly deviate from perfect calibration (Hosmer and Lemesbow, 1980). The reliability diagram provides a visualization of how well-calibrated a model is (DeGroot and Fienberg, 1983). Kuleshov et al. (2018) extends ECE to the regression setting. Unlike these methods, we'd like the metric to be scalar-valued in order to easily benchmark methods, and to only measure calibration.

3 Issues With Calibration Metrics

3.1 Not Computing Calibration Across All Predictions

Expected Calibration Error was crafted to mirror reliability diagrams, which are structured around binary classification such as rain vs not rain (DeGroot and Fienberg, 1983). A consequence is that the error metric is reductive in a multi-class setting. In particular, ECE is computed using only the predicted class's probability, which implies the metric does not assess how accurate a model is with respect to the $K - 1$ other probabilities.

In Figure 1, we observe that calibration error, when measured in a per-class manner, is non-uniform across classes. Guaranteeing calibration for all class probabilities is an important property for many applications. For example, a physician may care deeply about a secondary or tertiary prediction of a disease. Decision costs along both ethical and monetary axes are non-uniform, and thus clinical decision policies necessitate a comprehensive view of the likelihoods of potential outcomes (e.g., Tsoukalas et al., 2015). Alternatively, if a true data distribution exhibits high data noise (also known as high aleatoric uncertainty, or weak labels), then we'd like models to be calibrated across all predictions as no true class label is likely.

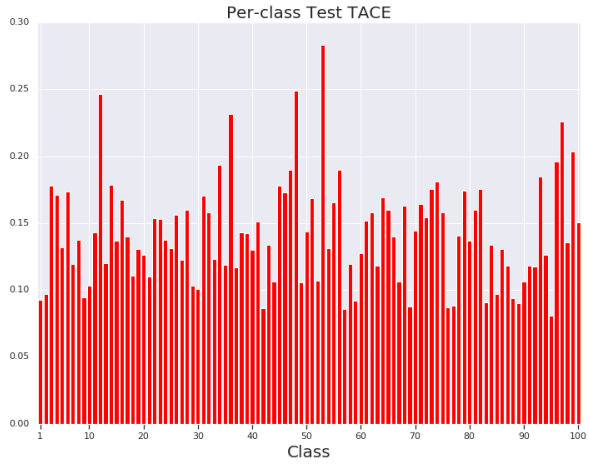

Figure 1: Per-class Thresholded Adaptive Calibration Error (TACE) for a trained 110-layer ResNet on the CIFAR-100 test set. We observe that calibration error is non-uniform across classes, which is difficult to express with a scalar metric and when measuring error only on the maximum probabilities.

3.2 Fixed Calibration Ranges

One major weakness of evenly spaced binning metrics is caused by the dispersion of data across ranges. In computing ECE, there is often a large leftward skew in the output probabilities, with the left end of the region being sparsely populated and the rightward end being densely populated. (That is, network predictions are typically very confident.) This causes only a few bins to contribute the most to expected calibration error—typically one or two as bin sizes are 10-20 in practice (Guo et al., 2017).

More broadly, sharpness is a fundamental property (Gneiting et al., 2007), which is the desire for models to always predict with high confidence, i.e., predicted probabilities concentrate to 0 or 1. Because of the above behavior, ECE conflates calibration and sharpness. An extreme example is if the model overfits, predicted probabilities will be pushed to upwards of 0.99, which places all predictions into one bin. This leads to zero calibration error, even though such sharp models are not necessarily calibrated.

3.3 Bias-Variance Tradeoff

Selecting the number of bins has a bias-variance tradeoff as it determines how many data points fall into each bin

and therefore the quality of the estimate of calibration from that bin's range. In particular, a larger number of bins causes more granular measures of calibration error (low bias) but also a high variance of each bin's measurement as bins become sparsely populated. This trade-off compounds particularly with the problem of fixed calibration ranges, as certain bins have many more data points than others.

4 Challenges in Calibration

Before describing new metrics for calibration, we first outline broad challenges with designing such metrics.

4.1 Ground Truth & Comparing Calibration Metrics

There are many challenges in measuring a network's calibration, beginning with the absence of ground truth. In principle, one can limit comparisons to controlled, simulated experiments where ground truth is available by drawing infinitely many samples from the true data distribution. However, even with ground truth, any estimator property, such as bias, remains difficult to compare across estimators, as "ground truth error" is multi-valued and estimators may make measurements for different elements of these values. Specifically, calibration is a guarantee for all predicted probabilities $p \in [0, 1]$ and class labels $y \in \{0, \ldots, K-1\}$. An estimator may have lower bias and/or variance in estimating the error for specific ranges of p and y but higher bias and/or variance in other ranges. For example, adaptive metrics use different binning strategies and therefore measure error with different settings of p (the average confidence of datapoints in a bin).

4.2 Weighting

The question of how to weight probability values (where one can see thresholding as a 0/1 weighting on datapoints below / above the threshold) creates a set of differences in calibration error metrics that choose to emphasize different aspects of calibration performance. In many contexts what really matters is the rare event - the network's classifications leading up to an accident, the presence of a planet, or the presence of a rare disease. Knowing the difference between whether a class's true probability is .01 and .001 (the difference between 1 in 100 and 1 in 1000) is both extremely difficult to discern and may be much more relevant than a difference between .3 and .301, which these calibration metrics would treat as equivalent. In these contexts, weighting the ends of the interval close to 0 and 1 would be ideal.

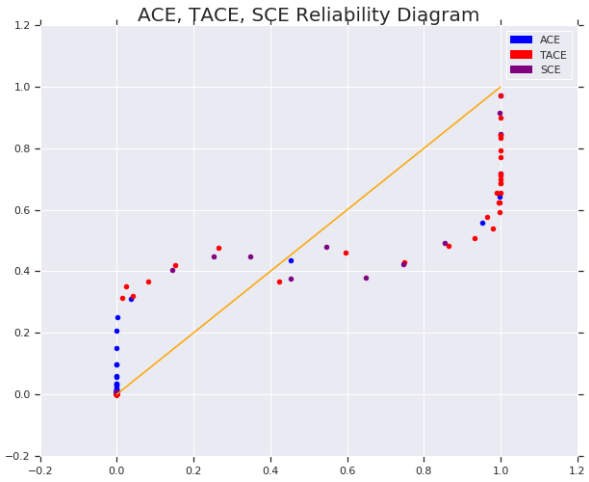

Figure 2: Reliability Diagram over adaptive metrics for a CNN on Fashion-MNIST validation data. TACE and ACE with 100 bins, SCE with 10.

Additionally, in the context of out-of-distribution detection, we would prefer to be well-calibrated in the middle of the spectrum, whereby a prediction of .5 (high uncertainty) is more likely to happen.

4.3 Visualization and Interpretability

Adaptive calibration metrics, due to the variability of their binning scheme, make it difficult to compare two models side-by-side. The calibration ranges that they choose will be different from one another, and those differences create a reliability diagram that is more difficult to interpret than the standard diagram over evenly-spaced calibration metrics.

5 New Calibration Metrics

5.1 Multiclass & Static Calibration Error

There are a few implications to making a multi-class calibration metric. One natural change is in the distribution of the predictions - as more classes are added, in order to remain accurate classifiers need to output small values for those classes' predictions (see sharpness of CNN predictions in Figure 2).

We introduce Static Calibration Error (SCE), which is a standard extension of Expected Calibration Error to every probability in the multiclass setting. SCE is an evenly spaced calibration range scheme that considers all softmax values.

5.2 Adaptivity & Adaptive Calibration Error

Adaptive calibration ranges are motivated by the bias-variance tradeoff in the choice of ranges, suggesting that in order to get the best estimate of the overall calibration error the metric should focus on the regions where the predictions are made (and focus less on regions with few predictions).

We chose an adaptive scheme which spaces the bin intervals so that each contains an equal number of predictions. ACE can be implemented as TACE with the threshold set to 0, and so one can follow the implementation in that section for more details.

5.3 Thresholding & Thresholded Adaptive Calibration Error

One initial challenge is that the vast majority of softmax predictions become infinitesimal. Without thresholding, these tiny values will dominate the binning schemes, introducing huge amounts of bias to the estimate of the correct output for the huge bins generated over space where the softmax has relatively sparse outputs (including the last bin, containing extreme values). These artifacts dominate the calibration score by default.

One natural solution to that challenge is thresholding - only including predictions above some epsilon (ex., .01 or .001) in the calibration score. This allows the metric to focus the preponderance of its calibration ranges on higher values, typically large predictions from the softmax.

In detail, TACE takes as input the predictions p (usually out of a softmax), correct labels ℓ, a number of ranges r and the threshold t value. The first sorted and thresholded values s are computed as

$$s = \text{sort}(p)[p <= t].$$

Given that, we compute the calibration ranges

$$c = s[\lfloor \text{range}(\text{len}(s)) \frac{\text{len}(s)}{r} \rfloor],$$

and use it to assign range indices to our predictions to get a digitized prediction matrix d. For each range we compute the mean probability

$$m = \mathbb{E}_r[(p[d == i])],$$

and compute our fraction correct f, equal to the number of values that are correct in each range divided by the total number of predictions that fall in that range. Finally, we compute our Thresholded Adaptive Calibration Error

$$\text{TACE} = \sum_{i=0}^{r} \frac{|(f_i - m_i)|}{r}.$$

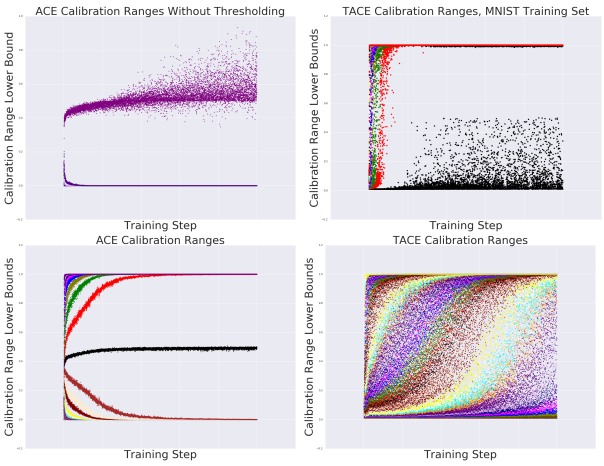

Figure 3: **Top Left:** Lower bounds of calibrations ranges over the course of training for adaptive calibration error on Fashion-MNIST, focusing almost entirely on small ranges and motivating thresholding. **Top Right:** On the MNIST training set with thresholding, so few values are small that the bottom of the lowest range often spikes to .99 and higher due to every datapoint being fit. **Bottom Left:** ACE on Fashion-MNIST validation with 100 calibration ranges. **Bottom Right:** Thresholded adaptive calibration with 50 calibration ranges over the course of training on Fashion-MNIST's validation set.

5.4 Criticisms & Limitations of Adaptive Calibration Metrics

With a static scheme it can be straightforward to compare the outputs of two very different models through a reliability diagram. With an adaptive scheme the ranges may fall in very different regions from algorithm to algorithm based on the density of its predictions, making comparisons much less intuitive.

These adaptive calibration metrics can create excessively large ranges if there is very sparse output in a region. In those cases, it may be better to have a higher variance estimate of the calibration, but only compare datapoints that are closer together (dropping datapoints that are farther away). This level of granularity can be replicated with an adaptive scheme that has a very large number of ranges, but that will come at the cost of easy interpretability and will increase the within-range variance across all ranges, not just the sparse regions.

This scheme does not accomplish transfer smoothly between all related predictions, or through ensembling / hierarchical calibration ranges approximate that smoothness. This limitation does suggest an ensemble based metric, which could leverage dynamic programming to look at very many range schemes efficiently.

6 Post-Processing Methods

6.1 Standard Approaches for Multiclass Classification

One common approach to calibration is to apply post-processing methods to the output of classifiers without retraining. These methods can be applied to any existing classifier thus freeing model design from the need to account for calibration-related measures at training time.

The two most popular post-processing methods are the parametric approach of Platt scaling (Platt et al., 1999) and the non-parametric approach of isotonic regression (Zadrozny and Elkan, 2002).

Platt scaling (Platt et al., 1999) fits a logistic regression model to the logits of a classifier on the validation set which can be used to compute calibrated predictions at test time. The original formulation of Platt scaling for neural networks (Niculescu-Mizil and Caruana, 2005) involves learning scalar parameters $a, b \in \mathbb{R}$ on a held-out validation set and then computing calibrated probabilities $\hat{p_i}$ given the uncalibrated logits vector z_i on the test set as $\hat{p_i} = \sigma(az_i + b)$. These parameters are typically estimated by minimizing the negative log likelihood.

Platt scaling can be extended to the multiclass setting by considering higher-dimensional parameters. In *matrix scaling* a is replaced with $W \in \mathbb{R}^{K*K}$ while we consider $b \in \mathbb{R}^K$. As for *vector scaling*; $W \in \mathbb{R}^K$. Calibrated probabilities for either of these extensions can then be computed as

$$\hat{p_i} = \max_k \sigma(Wz_i + b).$$

An even simpler extension is *temperature scaling* (Guo et al., 2017) which reduces the set of regression parameters to the inverse of a single scalar $T > 0$ such that

$$\hat{p_i} = \max_k \sigma(z_i/T).$$

On the other hand, isotonic regression (Zadrozny and Elkan, 2002) is a common non-parametric processing method that finds the stepwise-constant non-decreasing (isotonic) function f that best fits the data according to a mean-squared loss function $\sum_i (f(p_i) - y_i)^2$ where p_i are the uncalibrated probabilities and y_i the labels. Refer to (Zadrozny and Elkan, 2002) for the exact formulation of this regression problem.

The standard approach for extending isotonic regression to the multiclass setting is to break the problem into many binary classification problems (e.g. one-versus-all problems), to calibrate each problem separately, and then to combine the calibrated probabilities (Zadrozny and Elkan, 2002).

Method	ECE	TACE	SCE	ACE
Uncalibrated	19.638%	10.093%	0.412%	0.131%
Temp. Scaling	**2.155%**	**0.518%**	0.057%	0.057%
Vector Scaling	2.272%	0.613%	**0.037%**	**0.0133%**
Matrix Scaling	12.112%	3.983%	0.264%	0.257%
Isotonic Regr.	17.851%	2.827%	0.353%	0.118%

Table 1: ECE, TACE, SCE, and ACE (with 15 bins) on a ResNet-110 applied to CIFAR-100 before calibration, and after the application of post-processing methods.

6.2 Evaluation Challenges

When designing post-processing methods, not accounting for the properties detailed in Section 5 during evaluation can lead to misleading conclusions about a post-processing method's success.

For example, temperature scaling (Guo et al., 2017) has been shown to effectively minimize expected calibration error better than alternative techniques such as isotonic regression and Platt scaling (Platt et al., 1999). The question of whether these scaling techniques effectively minimize more sophisticated error metrics is an important standard for their efficacy.

In fact, by design of having only a single scalar parameter which uniformly scales predicted probabilities across all classes, temperature scaling is likely perform worse when accounting for, for example, calibration errors across all predictions. We hypothesize that this would be the case with respect to adaptive calibration metrics (e.g. ACE) as well as calibration metrics that account for the probabilities of all classes instead of that of only the top predicted class (e.g. SCE).

In the experiments section, we validate this hypothesis by comparing, from Table 1, the three extensions of Platt scaling and isotonic regression on ECE as well as SCE, ACE, and TACE.

7 Datapoint Specific Calibration Estimates via Dynamic Programming

In many contexts the calibration of a specific prediction is much more interesting to the consumer of a machine learning model than the model's overall calibration. A physician using a model to diagnose a patient cares about the calibration of that prediction in particular, rather than the model's average calibration error in general.

With that motivation, we propose a metric that uses a

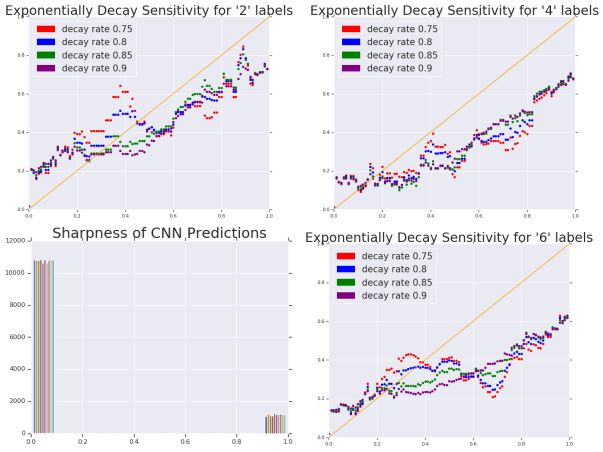

Figure 4: **Top Left**: Exponential Decay DP Metric Reliability Diagram for Fashion-MNIST Pullover Class **Top Right**: Exponential Decay Sensitivity to decay rates for Fashion-MNIST's predictions of coat class **Bottom Left**: Sharpness of predictions on Fashion-MNIST across entire softmax **Bottom Right**: As above, for shirt class

weighted exponential decay on the calibration of nearby datapoints on the predicted class to estimate the calibration of a prediction at a particular point. We use dynamic programming to efficiently compute the weighted impact of every datapoint's calibration error on the calibration at a particular point, where due to the exponential decay the contribution of a datapoint to the calibration error dies out smoothly with distance from the point at which the calibration is being measured (Figure 4).

One challenge with a method like this is that near the edges of the prediction space, there are more datapoints on the side that favors the rest of the dataset. This leads to a calibration score that is biased towards larger values near 0 and smaller values near 1. The sharpness of the network's predictions ameliorates this issue in our experiments.

8 Experiments

We designed and ran experiments to measure and explore the various calibration metrics on a variety of tasks and data, including simulations, MNIST (LeCun et al., 2010), Fashion MNIST (Xiao et al., 2017), CIFAR-10/CIFAR-100 (Krizhevsky and Hinton, 2009), and ImageNet 2012 (Deng et al., 2009). For the MNIST and Fashion MNIST experiments, we trained two-layer convolutional models. For the CIFAR-10/CIFAR-100 experiments, we trained 110-layer ResNet models (He et al., 2016). Finally, for the ImageNet experiments, we trained ResNet-50 models (He et al., 2016). Additional details beyond those found here and in the following subsections can be found in the supplementary material.

8.1 Comparisons of the Calibration Metrics

On the training data, the network trained on Fashion-MNIST overfits (Figure 9). For most of our calibration metrics this means comparing a mean prediction that is near 1 to a correct prediction that is exactly 1, leading to near zero calibration error (Figure 5). On the validation data, as the network overfits (making predictions that are more and more confident), the error tends to diverge from zero given that the network's true prediction accuracy is less than zero.

8.2 Per-Class Calibration Error

To explore per-class calibration error, we trained 110-layer ResNet models on CIFAR-10 and CIFAR-100. Figure 1 shows the calibration error of each of the 100 classes on the CIFAR-100 test set. We observe that calibration error is non-uniform across classes, which is difficult to express with a scalar metric and when measuring error only on the maximum probabilities.

8.3 Variance of the Calibration Metrics

We first use simulated data in order to understand the calibration metrics' properties in comparison to the true calibration error, estimated via draws from the true data distribution (Figure 7). Note that the average value of ECE is higher than the average value of TACE and ACE, incdicating that the latter two metrics are able to converge to the true calibration error faster with less variance.

8.4 Initialization

There is a desire for well calibrated output from the softmax of a neural network. That softmax processes inputs from a previous layer, and the magnitude of that layers outputs are strong determinants of the range of outputs that will be seen in the softmax.

One intervention that leads to large differences in the magnitude of the values in the network throughout training is the quality of the initialization. It has been shown that initialization can have a dramatic impact on training speed and stability. Here we show the impact on initialization quality on the calibration of the resulting network.

Figure 9 vizualizes the effects of model initialization. To the degree that the magnitude of this matrix is in a range amenable to the softmax, we see substantially different calibration scores. With an initialization of weights with

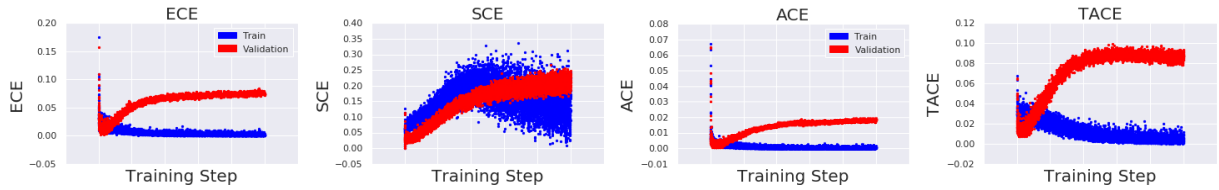

Figure 5: Calibration metrics measured over the course of training a convolutional model on Fashion-MNIST. Blue is training set, Red is validation set. **Left:** Expected Calibration Error. **Middle Left:** Static Calibration Error. **Middle Right:** Adaptive Calibration Error. **Right:** Thresholded Adaptive Calibration Error.

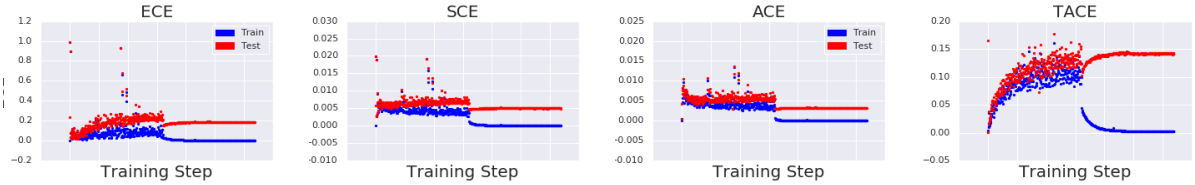

Figure 6: Calibration metrics measured over the course of training a 110-layer ResNet on CIFAR-100. Blue is training set, Red is test set. **Left:** Expected Calibration Error. **Middle Left:** Static Calibration Error. **Middle Right:** Adaptive Calibration Error. **Right:** Thresholded Adaptive Calibration Error.

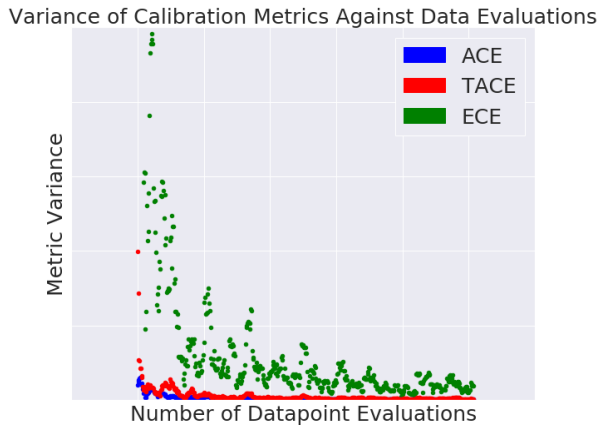

Figure 7: Convergence speed and stability of each metric, measuring variance while increasing the number of datapoints over which each metric is measured. Note that the average value of ECE is higher than the average value of TACE and ACE.

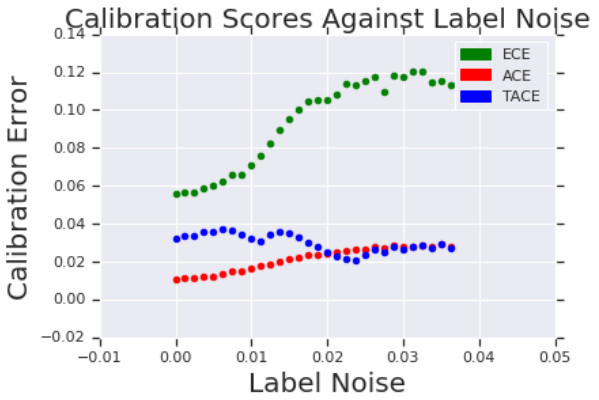

Figure 8: As label noise increases, ECE is outperformed by ACE and TACE. This shows that ECE becomes a worse approximation of true calibration error as class predictions beyond the predicted one matter more. (The $x = 0$ extreme has a true data distribution with deterministic y; $x \to \infty$ extreme has a true data distribution with uniform y.)

smaller values, the quality of network calibration scores increases dramatically.

8.5 Post-Processing Method Evaluation

We are interested in comparing the performance of temperature scaling, claimed to perform the best in (Guo et al., 2017), to that of vector scaling, matrix scaling, and isotonic regression. This performance is measured by the standard ECE metric as well as the novel metrics

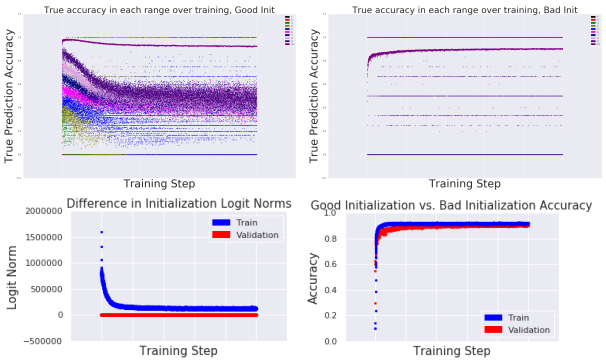

Figure 9: **Top Left:** Calibration scores for each of 10 bins (0.0-0.1, ..., 0.9-1.0) for a CNN over the course of training on Fashion-MNIST.The 90% bracket is well calibrated, but the 80% bracket and below become less calibrated over the course of training, as softmax prediction probabilities become overconfident. **Top Right:** With poor initialization, the diversity of probabilities output degrade dramatically. **Bottom Left:** The logit norm for the poor initialization is orders of magnitude larger than for the quality initialization. **Bottom Right:** Accuracy differences between the initializations are minor, despite dramatic differences in calibration.

presented in Section 5: SCE, ACE, and TACE.

We hypothesized, in Section 6, that having more degrees of freedom than temperature scaling would allow a post-processing method to more flexibly calibrate for probabilities across all classes. However, the ECE metric only captures the calibration for the top predicted class and is thus a suboptimal criterion for selecting a post-processing method in the multiclass setting.

We trained a ResNet-110 model (He et al., 2016) on CIFAR-100 (Krizhevsky and Hinton, 2009). We then estimated the regression parameters on the validation held-out set. Finally, we used the regression parameter estimates to compute calibrated probabilities on the test set. The results are reported in Table 1. A similar experiment with a ResNet-50 model on ImageNet 2012 is reported in Table 2 in the supplements and echoes the findings of Table 1.

As seen in Table 1, temperature scaling outperforms the other 3 methods on the ECE metric as reported in (Guo et al., 2017). However, temperature scaling falls short in comparison to vector scaling on the SCE and ACE metrics. In fact, the relative gap on SCE is the largest, among all metrics and across methods. This is unsurprising given that SCE is a multiclass extension of ECE which considers all probabilities, instead of the top one.

On the other hand, the performance of temperature scaling on the TACE metric could be attributed to thresholding leading to the consideration of only a fraction of the classes per datapoint. In this case, one would expect a closer result to ECE in terms of the relative performance of each post-processing method.

9 Discussion

We show that current ways to measure calibration can be problematic and investigate these pathologies. We design several calibration metrics to overcome them, and analyze architectures and recalibration methods with respect to the new metrics.

Ideally, machine learning models would be calibrated by default, without requiring post-processing on an additional data set as current recalibration methods do. One idea is to explicitly add a term to the training loss penalizing calibration error. For training, the aforementioned calibration measurement are poor choices as they are not differentiable. In future work, we aim to investigate the role of existing regularizers such as label smoothing, as well as proper scoring rules and broader methods such as Bayesian neural networks.

References

Crowson, C. S., Atkinson, E. J., and Therneau, T. M. (2016). Assessing calibration of prognostic risk scores. *Statistical methods in medical research*, 25(4):1692–1706.

Dawid, A. P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610.

DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268.

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

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hosmer, D. W. and Lemesbow, S. (1980). Goodness of fit tests for the multiple logistic regression model. *Communications in statistics-Theory and Methods*, 9(10):1043–1069.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2011). Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

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

Kuleshov, V., Fenner, N., and Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18.

Murphy, A. H. and Epstein, E. S. (1967). Verification of probabilistic predictions: A brief review. *Journal of Applied Meteorology*, 6(5):748–755.

Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM.

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

Raghu, M., Blumer, K., Sayres, R., Obermeyer, Z., Kleinberg, R., Mullainathan, S., and Kleinberg, J. (2018). Direct uncertainty prediction for medical second opinions.

Tsoukalas, A., Albertson, T., and Tagkopoulos, I. (2015). From Data to Optimal Decision Making: A Data-Driven, Probabilistic Machine Learning Approach to Decision Support for Patients With Sepsis. *JMIR Medical Informatics*, 3(1):e11.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM.

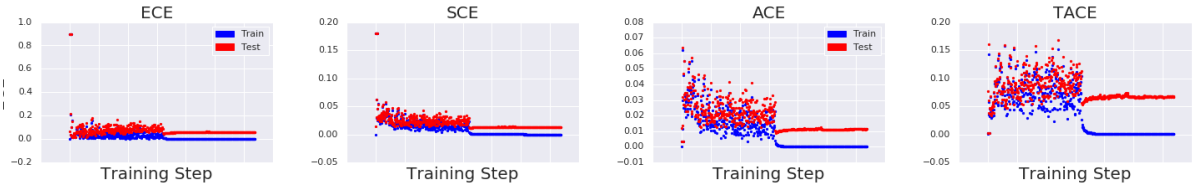

Figure 10: Calibration metrics measured over the course of training a 110-layer ResNet on CIFAR-10. Blue is training set, Red is test set. **Top Left:** Expected Calibration Error. **Top Right:** Static Calibration Error. **Bottom Left:** Adaptive Calibration Error. **Bottom Right:** Thresholded Adaptive Calibration Error.

A Replication Details

MNIST & Fashion-MNIST. Experiments on MNIST and Fashion-MNIST are run with a 2-layer CNN with max pooling after each layer (Filter counts are 32 followed by 64 on $5x5$ filters and $2x2$ pooling layers), followed by two affine transformations (neuron counts are $3136 - 1024$ followed by $1024 - 10$). Relu activation functions are used throughout. A softmax is run over the output layer to produce the predictions.

The good initialization is Glorot which we use for all experiments other than the initialization comparison, where the poor initialization is a normal distribution with mean 0 and standard deviation 1 Over the weights and biases. The model is optimized with Adam, using a learning rate of 0.001, a momentum parameter of 0.9 and a RMS parameter of 0.999. The epsilon term is $1e-08$. Our loss function is a cross entropy loss. Runs go to $100,000$ steps with calibration metrics run at every 10th step. Minibatch size was 64.

CIFAR-100 & CIFAR-10. We follow the training setup presented in He et al. (2016) for training a 110-layer ResNet for CIFAR-10/CIFAR-100. For experiments measuring calibration error, we train on the full training split, and report on the test set. For the recalibration experiments, we perform an 80%/20% train/validation split of the training data, train the model on the training split, train the recalibration method on the validation dataset, and report results on the test set.

ImageNet. We train a ResNet-50 as presented in He et al. (2016) along with label smoothing. In all experiments, models are trained on the full training split, and the validation dataset is split into 25k/25k validation/test splits. Recalibration methods are trained on the 25k validation split and results are reported on the 25k test split.

Specific Calibration Estimates via Dynamic Programming For well-calibration of a prediction of class c at probability p, we can look at all the predictions of c near p, weighted by their distance away from p. Our equation for actual accuracy at prediction p is then simply (weighted times the actual class was c near p)/(weighted times we made a prediction near p). Our weightings is by decay rate away from p, where a data point .01 away from p is weighted decay rate as much as a datapoint at p (and something .02 away is worth $decayrate^2$, etc).

This can easily get computationally expensive if we query the calibration at many different points, so we use dynamic programming to get the "canonical calibration" at 101 evenly spaced points from 0.00 to 1.00 (initial experiments show that this is not sensitive to the number of buckets). A further optimization is that we make two passes (left-to-right and right-to-left), and reuse earlier work (eg, to get the canonical left-to-right calibration of .51, we only need to calculate the calibration from .5 to .51, and then multiply the left-to-right of .5 by the decay rate).

With the canonical calibration, we can then query any given probability by looking at the two points of canonical calibration nearby, and then linearly interpolating.

A final (unprincipled) design choice we made is to place 50-50 weights for points below and above a canonical calibration threshold, instead of weighting by total number of points. This is to help ameliorate the issue with having very high or very low probabilities.

B Additional Figures

Method	ECE	TACE	SCE	ACE
Uncalibrated	6.626%	2.511%	0.016%	0.0146%
Temp. Scaling	5.420%	2.637%	0.010%	0.0007%
Vector Scaling	**1.443%**	**1.249%**	**0.002%**	0.0044%
Matrix Scaling	5.061%	1.980%	0.010%	0.0005%
Isotonic Regr.	3.474%	1.862%	0.008%	**0.0000 %**

Table 2: ECE, TACE, SCE, and ACE (with 15 bins) on a ResNet-50 applied to ImageNet before calibration, and after the application of various extensions to Platt scaling and Isotonic regression.

Figure 11: Per-class calibration error for a trained 110-layer ResNet on the CIFAR-100 test set. We observe that calibration error is non-uniform across classes, which is difficult to express with a scalar metric and when measuring error only on the maximum probabilities. **Top:** Per-class Static Calibration Error. **Middle:** Per-class Adaptive Calibration Error. **Bottom:** Per-class Thresholded Adaptive Calibration Error.