Hidden Heterogeneity: When to Choose Similarity-Based Calibration

Anonymous authors
Paper under double-blind review

Abstract

Trustworthy classifiers are essential to the adoption of machine learning predictions in many real-world settings. The predicted probability of possible outcomes can inform high-stakes decision making, particularly when assessing the expected value of alternative decisions or the risk of bad outcomes. These decisions require well-calibrated probabilities, not just the correct prediction of the most likely class. Black-box classifier calibration methods can *improve the reliability* of a classifier's output without requiring retraining. However, these methods are unable to detect subpopulations where calibration could also *improve prediction accuracy*. Such subpopulations are said to exhibit "hidden heterogeneity" (HH), because the original classifier did not detect them. The paper proposes a quantitative measure for HH. It also introduces two similarity-weighted calibration methods that can address HH by adapting locally to each test item: SWC weights the calibration set by similarity to the test item, and SWC-HH explicitly incorporates hidden heterogeneity to filter the calibration set. Experiments show that the improvements in calibration achieved by similarity-based calibration methods correlate with the amount of HH present and, given sufficient calibration data, generally exceed calibration achieved by global methods. HH can therefore serve as a useful diagnostic tool for identifying when local calibration methods would be beneficial.

1 Introduction

How do we know when to trust a prediction? A classifier is said to be *well calibrated* or *reliable* with respect to a distribution specified by $P(Y|X)$ if the probability it assigns to an outcome matches the true probability of that outcome according to $P(Y|X)$, for observation X and class label Y. Well-calibrated predictions improve the trustworthiness of systems and support downstream cost-sensitive decisions (e.g., medical diagnosis, autonomous driving, financial decisions). Likewise, calibration is necessary when combining or comparing predictions from different sources (Bella et al., 2013) or in classifier cascades that use a low-cost but less accurate classifier's output to decide whether to apply a higher-cost but more accurate secondary classifier (Enomoto & Eda, 2021). Good calibration is beneficial in any decision making setting in which uncertainty matters (e.g., active learning or classification with a rejection or abstention option).

We focus on an increasingly common use case in which we would like to apply a pre-trained, possibly proprietary, model $\mathcal{M}$ to our own data set $\mathcal{D}$ with corresponding distribution $P_D(Y|X)$. In this scenario, the original training data set is unavailable and $P(Y|X)$, the distribution for which $\mathcal{M}$ was trained (and possibly calibrated), is unknown. Any domain shift between P and P_D could prevent $\mathcal{M}$ from generating reliable predictions on $\mathcal{D}$. Moreover, even in the absence of domain shift, $\mathcal{M}$ may perform poorly on D due to "hidden heterogeneity", which occurs when $\mathcal{M}$ assigns the same posterior probability to items with different true probabilities.

Concerns about poorly calibrated classifiers are not new (e.g., Zadrozny & Elkan, 2001; Niculescu-Mizil & Caruana, 2005), and several post-training calibration correction methods have been developed (e.g., Guo et al., 2017; Kumar et al., 2019; Kull et al., 2019; Alexandari et al., 2020). In general, these methods devise a *calibration map* Φ that transforms the original predicted probabilities. Let the output of a classifier $f(x_i)$ applied to item x_i be a probability vector $\hat{p}_i$ of length K (number of classes) that sums to 1 (i.e., resides in

the simplex Δ_{K-1}). The calibration map $\Phi : \Delta_{K-1} \mapsto \Delta_{K-1}$ is derived from an independent calibration set $\mathcal{C}$ to transform $\hat{p}_i$ to a more reliable $\hat{q}_i = \Phi(\hat{p}_i)$.

A key limitation of these calibration maps is the implicit assumption that all items with the same predicted probability $\hat{p}$ should be given the same correction. Such maps cannot accommodate hidden heterogeneity, which manifests as distinct subpopulations to which the classifier has erroneously assigned the same $\hat{p}$ value. For example, in predicting cancer risk, there could be many different reasons (age, lifestyle, family medical history, etc.) that a given individual is predicted to have $\hat{p} = 0.9$. For some conditions, this probability could be an over-estimate, while for others, it could be an under-estimate. No global calibration map can address this heterogeneity, because all items with the same $\hat{p}$ are mapped to the same $\hat{q}$.

We propose a method to quantify hidden heterogeneity (HH) as a signal for when global calibration may be inadequate. Once HH is detected, we face a choice of either (a) training a new classifier on the available calibration data or (b) improving the existing classifier using the calibration data. Because HH is a local phenomenon, the natural way to improve the classifier is to apply a local, similarity-based calibration technique. We introduce two local calibration methods that leverage the location of x_i in feature space to yield $\hat{q}_i = \Phi(\hat{p}_i|x_i)$. These methods determine the calibrated probability $\hat{q}_i$ by taking a weighted vote of data points in the calibration set $\mathcal{C}$. The first method, Similarity-Weighted Calibration (SWC), assigns weights to every point in the calibration set based on similarity to x_i. The second method, SWC-HH, uses only items within a local neighborhood defined by the estimated HH. We refer to the weighted number of calibration data points as the "calibration support" for x_i, which indicates how much calibration data is available for estimating $\hat{q}_i$. This measure of the calibration quality of $\hat{q}_i$ for each x_i is a unique advantage of local calibration.

We note that any post-hoc calibration method can be viewed as a form of model stacking (Wolpert, 1992), in which the output of the original classifier is transformed via Φ, a model itself. Our SWC and SWC-HH methods are stacking methods that focus on improving local calibration. As a consequence, they also reduce or eliminate HH and can thereby improve classifier accuracy.

The major contributions of this paper are

1. The identification of hidden heterogeneity as a property of classifier predictions that thwarts global calibration methods (Section 3),
2. A method for quantifying hidden heterogeneity to indicate when local calibration is needed (Section 3),
3. Two local calibration methods based on Similarity-Weighted Calibration (Section 4), and
4. Results of experiments that assess the relationship between hidden heterogeneity and calibration, yielding useful guidance for practitioners (Section 5).

We provide context from previous work in Section 2. Key conclusions and limitations of local, similarity-based calibration are discussed in Section 6.

2 Related Work

There are several methods for improving the reliability (calibration) of classifier predictions. Many recent advances were inspired by the recognition that deep neural networks in particular may sacrifice calibration to achieve higher generalization accuracy (Guo et al., 2017). Strategies include using calibration-sensitive training methods, if the original training set is available (e.g., via modifications to the loss function proposed by Kumar et al., 2018; Mukhoti et al., 2020; Enomoto & Eda, 2021; Tomani & Buettner, 2021), using domain-specific representations that lead to improved calibration (Kalmady et al., 2021), or adopting network architectures that do not use convolutions (Minderer et al., 2021). In contrast, post-hoc calibration correction methods that directly modify the classifier's predictions on new observations, without re-training, can be employed even when the training data (or model) are proprietary or when the data distribution has changed and we wish to recalibrate an existing classifier to extend its applicability.

Global, parametric calibration methods re-map the predicted probabilities output by the classifier, $\hat{p}$, by fitting a chosen functional form (e.g., logistic curve) from the probabilities to the labels to compute $\hat{q} = \Phi(\hat{p})$.

For binary classifiers, Platt scaling (Platt, 1999) transforms $\hat{p}_i$ into a value between 0 and 1 using a sigmoid function with two parameters, A and B: $\hat{q}_i = \frac{1}{1+e^{A\hat{p}_i+B}}$. The parameters A and B are chosen to optimize the negative log-likelihood of predictions made on the calibration set. Platt scaling was generalized to multi-class problems for neural networks (Guo et al., 2017) via a method called temperature scaling, which operates on the logits z_i (not the probabilities) by optimizing a temperature parameter T in $u_i[k] = e^{z_i[k]/T}$, where $z_i[k]$ is the logit for item i and class k, and $u_i[k]$ is the corresponding unnormalized probability. These values are normalized as $\hat{q}_i[k] = \frac{u_i[k]}{\sum_j u_j[k]}$. The same T value is used for all classes. Bias-Corrected Temperature Scaling (Alexandari et al., 2020) adds a bias term for each class.

There are also several approaches that construct probability bins and assign the average (or other aggregate) accuracy within bin B_b as its calibrated probability, $\hat{q}_i := Acc_b, \forall i \in B_b$. Histogram binning (Zadrozny & Elkan, 2001) assigns items to bins based on their uncalibrated predictions $\hat{p}_i$, often using equally-spaced bin boundaries or divided so that each bin has the same number of items ("equal frequency") from the calibration set. Kumar et al. (2019) found that the latter strategy, as well as using a larger number of bins, yields better results. Isotonic regression (Zadrozny & Elkan, 2002) adds further flexibility by optimizing the bin boundaries to minimize the squared loss between $\hat{q}_i$ and y_i. Recently, Patel et al. (2021) proposed selecting the bin boundaries to maximize the mutual information between bin predictions $\hat{q}_i$ and y_i.

To date, very few calibration methods have leveraged the location of items in feature space, $\mathcal{X}$. Zhao et al. (2020) introduced "individual" (per-item) calibration for regression problems and confidence intervals. Partial specialization for classification problems can be achieved by estimating a different T per subpopulation (unlabeled cluster (Gong et al., 2021) or labeled "domain" (Yu et al., 2022)), then employing linear regression to estimate a new T' for each test item. Our approach operates at a finer (per-item) granularity and is not restricted to probability rescaling. The LoRe calibration method computes a "confidence" that is weighted by item similarity but is confined to a pre-specified probability bin (Luo et al., 2021). Our approach uses the whole calibration set (not just those in a probability bin), and it estimates the full posterior distribution across classes $\hat{q}_i[k], \forall k$ (not just the highest probability class).

One calibration approach that employs similarity to compute the complete $\hat{q}_i[k]$ vector is Similarity-Binning Averaging or SBA-10 (Bella et al., 2009), which creates bins (neighborhoods) based on an item's 10 nearest neighbors in an "augmented" feature space $\mathcal{X}^+ = \mathcal{X} \times \Delta_{K-1}$ defined by the item's feature vector x_i of dimension d concatenated with its probability vector $\hat{p}_i \in \Delta_{K-1}$. SBA-10 computes the calibrated probability $\hat{q}_i[k]$ as the probability of class k (in the calibration set) within item i's assigned bin, with each item contributing equally (no weighting). In contrast, our approach uses a similarity-weighted contribution from each item in the calibration set, not just the 10 nearest neighbors.

3 Hidden Heterogeneity

Global post-hoc calibration methods, such as Platt scaling and temperature scaling, perform very well for some data sets and algorithms and less well for others. Similarly, local methods like SBA-10 do not always improve upon these global methods. What causes the failure of global methods, and under what conditions can local methods do better? Our hypothesis is that global post-hoc calibration fails when the data exhibits *hidden heterogeneity* (HH) with respect to the predicted probabilities $\hat{p}$. HH characterizes situations where there are subpopulations in the feature space $\mathcal{X}$ to which the classifier assigns the same $\hat{p}$ but that require different calibration corrections.

Definition 1 A classifier f exhibits *hidden heterogeneity* with respect to a feature space $\mathcal{X}$ if there exists a subregion $\mathcal{U} \subseteq \mathcal{X}$ such that $f(x) \approx \hat{p}$ for all $x \in \mathcal{U}$ and yet $\mathcal{U}$ can be partitioned into M disjoint subregions $\mathcal{U} = \mathcal{U}_1 \coprod \cdots \coprod \mathcal{U}_M$ such that the true class probabilities $P(y|x \in \mathcal{U}_m) \neq P(y|x \in \mathcal{U}_{m'})$ for all distinct pairs $m, m' \in \{1, \ldots, M\}, m \neq m'$.

An extreme example of HH occurs for a classifier that ignores all features and predicts the majority class for all items. Imagine a data set composed of 60% cats and 40% birds, for which a classifier predicts $P(y =$ "cat"$) = \hat{p} = 0.6$ for all items (i.e., $\mathcal{U} = \mathcal{X}$). If cats and birds are not separable in the feature space, this may be the best one can do. However, if the items have a feature such as "number of legs", then

Algorithm 1 Hidden Heterogeneity (HH)

Input: Test item t, calibration data $\mathcal{C}$, predicted probabilities $\hat{p}$, and radius r
Output: Hidden heterogeneity in neighborhood around t

1: Construct probability neighborhood around t: $\mathcal{U}_t = \{x_i \in \mathcal{C} | D_H(\hat{p}_t, \hat{p}_i) < r\}$ (using Eqn. 2).
2: Train g_t using labeled data in $\mathcal{U}_t$.
3: Collect model predictions for the neighborhood: $f(\mathcal{U}_t) = \{\hat{p}_i | x_i \in \mathcal{U}_t\}$.
4: Collect g_t predictions for the neighborhood: $g_t(\mathcal{U}_t) = \{g_t(x_i) | x_i \in \mathcal{U}_t\}$.
5: Collect labels for the neighborhood: $Y_{\mathcal{U}_t} = \{y_i | x_i \in \mathcal{U}_t\}$.
6: Calculate $HH_{\mathcal{U}_t}$ using $f(\mathcal{U}_t)$, $g_t(\mathcal{U}_t)$, and $Y_{\mathcal{U}_t}$ (Eqn. 3).

two subregions—$\mathcal{U}_1$ for animals with two legs and $\mathcal{U}_2$ for animals with four legs—can be defined with true conditional probabilities of 1 (for "cats") and 0 (for "birds"). This heterogeneity is hidden in the classifier's predictions.

This extreme situation (complete HH) could happen for a number of reasons (majority-class classifier, classifier only trained on cats, overly limited hypothesis space, etc.). More commonly, any classifier may have one or more subregions $\mathcal{U}$ in its predicted probabilities that likewise obscure informative heterogeneity. Detecting HH can alert the practitioner to limitations of the classifier and provide opportunities for improvement via local calibration methods. While global methods that map $\hat{p}$ to $\hat{q}$ cannot address HH, local calibration could model the subregions separately and assign $\hat{q}_i$ differently for each $\mathcal{U}_i$.

Algorithm 1 provides a method to compute the *detectable* hidden heterogeneity for a region $\mathcal{U} \subseteq \mathcal{C}$ given a labeled calibration data set $\mathcal{C}$ sampled from the same distribution as the test set. HH is calculated as the potential improvement (compared to the original classifier) achieved by training a specialized classifier on only the items in $\mathcal{U}$.

In step 1, we define item t's probability neighborhood $\mathcal{U}_t$ to contain calibration items that are close to t in the probability simplex Δ_{K-1}. More precisely, $\mathcal{U}_t$ contains the items within radius r of item t in Δ_{K-1}. We employ Hellinger distance D_H to calculate the distance between probability vectors $\hat{p}_i$ and $\hat{p}_j$. Hellinger distance is the probabilistic equivalent of Euclidean distance and, as a metric, is more suitable in this case than e.g., KL divergence.

$$D_H(\hat{p}_i, \hat{p}_j) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{K} \left(\sqrt{\hat{p}_i[k]} - \sqrt{\hat{p}_j[k]} \right)^2}. \tag{1}$$

Conveniently, this can be expressed as the Euclidean norm of the difference of the element-wise square root of each probability vector (Krstovski et al., 2013):

$$D_H(\hat{p}_i, \hat{p}_j) = \frac{1}{\sqrt{2}} \left\| \sqrt{\hat{p}_i} - \sqrt{\hat{p}_j} \right\|_2. \tag{2}$$

This in turn allows the use of efficient methods (e.g., k-d tree) for populating neighborhood $\mathcal{U}_t$.

For each test item t, a new (local) classifier g_t is trained using only the nearby calibration items in $\mathcal{U}_t$ (step 2). This classifier g_t can be any classifier type. We employed an ensemble method that can perform internal generalization estimates without an additional validation set. We trained a bagged ensemble of 50 decision trees with no depth limit and no limit on the number of features searched for each split. We used out-of-bag error to determine how much pruning to employ to achieve good generalization and avoid overfitting to the calibration set. We searched over 7 values of the α pruning complexity parameter, evenly spaced between 0.0 (no pruning) and 0.03, as input to the minimal cost-complexity pruning method (Breiman et al., 1984).

Finally (step 6), we calculate HH for $\mathcal{U}_t$ by comparing the Brier score (Brier, 1950) of the original predictions by model f on $\mathcal{U}_t$ (step 3) with those generated by the local model g_t (step 4) using true labels $Y_{\mathcal{U}_t}$ (step 5):

$$HH_{\mathcal{U}_t} = Brier(f(\mathcal{U}_t), Y_{\mathcal{U}_t}) - Brier(g_t(\mathcal{U}_t), Y_{\mathcal{U}_t}), \tag{3}$$

Algorithm 2 Similarity-Weighted Calibration (SWC)

Input: Test item t, calibration data $x_i \in \mathcal{C}$ and labels y_i
Output: Calibrated probabilities $\hat{q}_t[k], \forall k$

1: Collect model predictions for item t: $\hat{p}_t[k]$ for $k \in \{1 \dots K\}$.
2: Collect model predictions for the calibration set: $\hat{p}_i[k]$ for $x_i \in \mathcal{C}, k \in \{1, \dots, K\}$.
3: Compute pairwise similarity as $s(t, i)$ for $x_i \in \mathcal{C}$.
4: Compute $\hat{q}_t[k] = \frac{1}{\sum_i s(t,i)} \sum_i s(t, i)\mathbb{1}(y_i = k)$ for $k \in \{1, \dots, K\}$ (Eqn. 5).

where the Brier score is the mean squared error between predictions $\hat{p}_i[k] \in [0, 1]$ and labels y_i, for N items and K possible classes:

$$Brier(\hat{p}, Y) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \Big(\hat{p}_i[k] - \mathbb{1}(y_i = k)\Big)^2. \tag{4}$$

We enforce the condition that g_t is no worse than f by clipping $HH_{\mathcal{U}_t}$ to 0. Regions with large HH values provide both a warning that global calibration methods may not perform well and an opportunity for local specialization by using item similarity during calibration.

4 Similarity-Weighted Calibration

We propose to calibrate classifier predictions by leveraging information in feature space as well as the uncalibrated probabilities $\hat{p}_i$. Given test item t, the goal is to estimate well-calibrated $\hat{q}_t[k] = P(y = k|x_t)$ for each class $k \in \{1 \dots K\}$. This approach to calibration, Similarity-Weighted Calibration (SWC), is described in Algorithm 2. Let

$$s(t, i) = sim(\langle x_t, \hat{p}_t \rangle, \langle x_i, \hat{p}_i \rangle) \in [0, 1]$$

be the similarity between item t and item i measured in the augmented space $\mathcal{X}^+$. A similarity of 1 is perfect identity. We employ a supervised similarity measure known as the random forest *proximity function* (RFprox) so that similarity is informed by the concept to be learned. RFprox trains a random forest on a labeled data set and defines the similarity between items x_i and x_j as the fraction of times they are assigned to the same leaf in each tree of the ensemble (Breiman, 2001; Cutler et al., 2012). Effectively, the random forest encodes a "kernel" defined by those weights (leaf co-occurrences) (Hastie et al., 2009). We employ the calibration data to learn the relevant RFprox measure using a random forest with 100 trees, no depth limit, and considering a random set of $\sqrt{d}$ features for each split, given d total features.

SWC computes the similarity of t to every item in the calibration set (step 3) and uses this information to replace $\hat{p}$ with a similarity-weighted combination of labels from the calibration set (step 4).

$$\hat{q}_t[k] = \frac{1}{\sum_i s(t, i)} \sum_i s(t, i)\mathbb{1}(y_i = k) \tag{5}$$

The similarity-based approach to calibration enables local specialization within the data set, but it does not directly make use of the calculated HH. We also developed the SWC-HH algorithm, which filters the calibration set to restrict which items are used to generate $\hat{q}$. HH, which is computed separately for each test item t, is provided as an additional input for calibration. In step 4, SWC-HH only includes items with a similarity to item t of at least $\frac{1}{2}HH_{\mathcal{U}_t}$. Note that the maximum value for HH is 2.0 since it is the difference in Brier scores, clipped to 0.0, and each Brier score ranges between 0.0 and 2.0. Dividing by 2.0 normalizes the threshold to the range 0.0–1.0, making it suitable as a similarity threshold.

5 Experimental Results

We conducted experiments with a variety of classifiers and data sets to compare local and global calibration methods and to determine the role that hidden heterogeneity plays. Our hypotheses were that (1) local,

similarity-based calibration is more effective at reducing Brier score than global calibration methods, (2) the amount of improvement correlates with the hidden heterogeneity score, and (3) calibration support can serve as an indicator of per-item calibration quality. Our implementations of SWC, SWC-HH, and other calibration methods, along with scripts to replicate the experiments, are available at *URL omitted for blind submission.*

5.1 Methodology

We assessed calibration methods for six tabular data classifiers as implemented in the scikit-learn Python library (Pedregosa et al., 2011), including a decision tree with `min_samples_leaf = 10` (DT), a random forest with 200 trees (RF), an ensemble of 200 gradient-boosted trees (GBT), a linear support vector machine (SVM), a Gaussian kernel ($\gamma = \frac{1}{d\,var(X)}, C = 1.0$) support vector machine (RBFSVM), and a Naive Bayes classifier (NB). Any parameters not explicitly mentioned were set to their default values. We also conducted experiments with pre-trained deep neural networks for classifying images (details in Section 5.4).

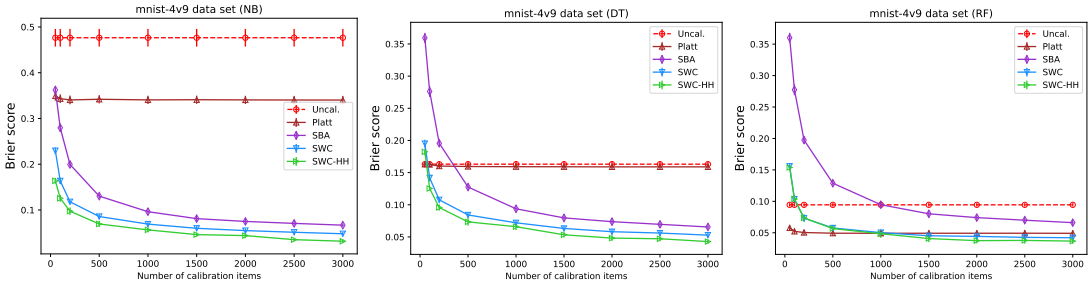
Data sets. We used four tabular and two image data sets:

- moons: a synthetic 2D data set with two partially overlapping classes, chosen to enable visualization of classifier outputs and hidden heterogeneity in feature space. Observations were generated using the scikit-learn `make_moons()` function with `noise` set to 0.3 and `random_state` set to 0.
- letter (letter recognition): a 26-class data set of capital letters from the English alphabet represented by 16 statistical and geometrical features that describe the image of the letter (Frey & Slate, 1991), available at `https://archive.ics.uci.edu/ml/datasets/letter+recognition`.
- mnist: the MNIST handwritten digit data set (LeCun et al., 1998) composed of 28x28 pixel ($d = 784$) images containing a digit from 0 to 9. We used the data set as provided by OpenML; the original source is `http://yann.lecun.com/exdb/mnist/`. In addition to the 10-class data set (mnist10), we created several binary subsets consisting only of two digits, such as "4" and "9" (mnist-4v9). This data set is "tabular" (1-d feature vector); no 2D image structure is preserved.
- fashion-mnist: grayscale images of clothing and accessories (10 classes) that was designed to be more challenging than the MNIST data set yet have the same dimensionality (28x28, $d = 784$) (Xiao et al., 2017), available at `https://github.com/zalandoresearch/fashion-mnist`.
- CIFAR-10: 60,000 images (64 $\times$ 64 pixels) labeled into 10 distinct classes; the test set contains 10,000 images (Krizhevsky, 2009)
- CIFAR-100: a disjoint set of 60,000 images labeled into 100 different classes (50k train, 10k test)

For tabular data sets, we randomly sampled 10,000 items and randomly split them into 500 train, 500 test, and 9000 for a calibration pool. For the "mnist10" and "letter" data sets, we used 1000 items each for training and test, due to their large number of classes (10 and 26, respectively). We calibrated each trained model using a series of nested calibration sets of size { 50, 100, 200, 500, 1000, 1500, 2000, 2500, 3000 } to assess the data efficiency of each calibration method. For image data sets, we generated a class-stratified random split of the standard test set into 5000 test items and reserved the remainder as the calibration set. We report average performance across 10 trials along with the standard error for the observed values.

Calibration methods. We compared three similarity-based calibration methods (SBA-10, SWC, and SWC-HH) to standard calibration methods including Platt scaling (Platt, 1999) and histogram binning (Zadrozny & Elkan, 2001). Our implementation of SBA-10 employs Euclidean distance to identify the nearest neighbors, which we believe to be the method employed by Bella et al. SWC and SWC-HH used the RFprox similarity measure. When computing hidden heterogeneity, we used a probability radius of $r = 0.1$ in the probability simplex.

Our implementation of Platt scaling optimizes the negative log-likelihood of predictions against the target probabilities rather than discrete $\{0, 1\}$ labels (Platt, 1999; Niculescu-Mizil & Caruana, 2005). With n_+ as the number of calibration items in the positive class and n_- as the number of negative items, the target probabilities are $y_+ = \frac{n_+ + 1}{n_+ + 2}$ and $y_- = \frac{1}{n_- + 2}$. For multi-class problems, we applied temperature scaling (Guo et al., 2017). For classifiers that output probabilities instead of logits, we first transformed $\hat{p}_i$ into logits as



(a) Naive Bayes (b) Decision tree (c) Random forest

Figure 1: Calibration performance for three classifiers on the MNIST 4-vs-9 data set. Each plot shows Brier score (lower is better) with increasing calibration set size. Error bars show one standard error over 10 trials.

$z_i[k] = \ln \frac{\hat{p}_i[k]}{1-\hat{p}_i[k]}$. To avoid dividing by zero (or taking its logarithm), we clipped $\hat{p}_i[k]$ to the range $[\epsilon, 1-\epsilon]$, where $\epsilon = 1 \times 10^{-12}$. We applied the histogram binning method implemented by Kumar et al. (2019)[1] and followed their recommendation to use equal-mass bins (100 bins).

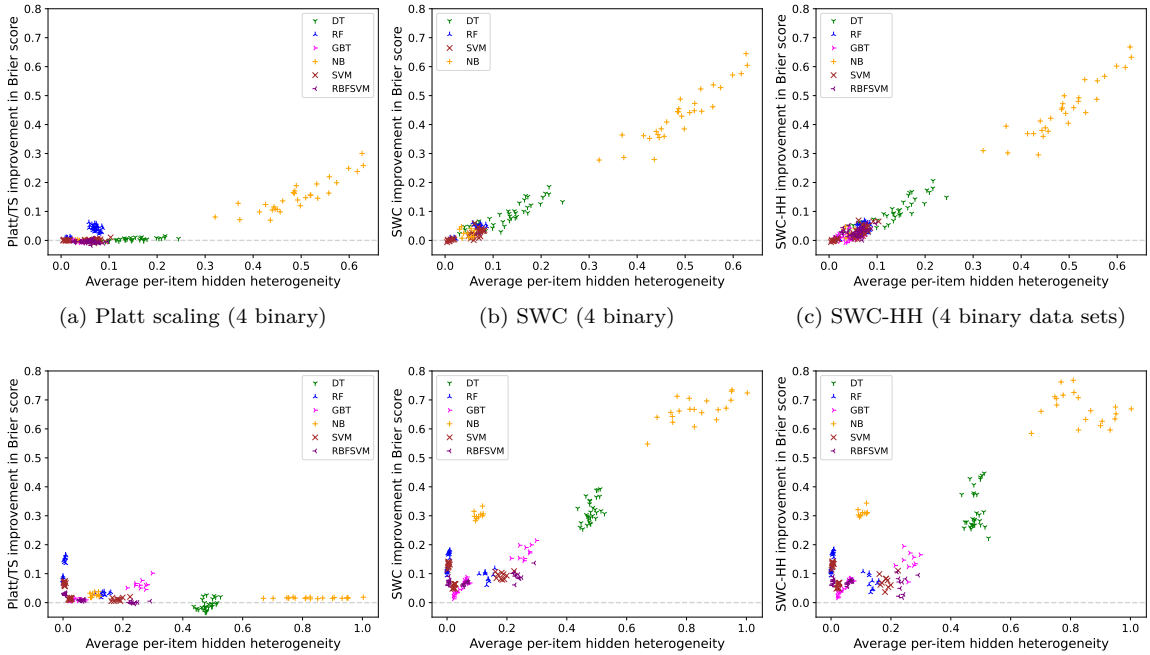
Metrics. We employ Brier score (Equation 4) to measure prediction quality, following earlier researchers (e.g., Zadrozny & Elkan, 2001; 2002; Niculescu-Mizil & Caruana, 2005). It has several advantages over a commonly used calibration metric called the Expected Calibration Error (ECE) (Naeini et al., 2015), which assigns predictions to bins to compare the average "confidence" to the accuracy of predictions within the bin. ECE can be trivially minimized to 0 by always predicting the empirical average probability of a given class, yielding perfectly calibrated but uninformative predictions (Kull et al., 2017; Widmann et al., 2019; Ovadia et al., 2019). The Brier score incorporates not just calibration error (or "reliability") but also "resolution" or sharpness, which rewards predictions that deviate from the average probability (Ferro & Fricker, 2012). In addition, ECE is sensitive to the number and choice of bins (Vaicenavicius et al., 2019; Kumar et al., 2019; Patel et al., 2021), it exhibits undesirable edge effects (discontinuities at bin boundaries), and it only assesses calibration of the predicted class. Brier score characterizes prediction quality across all classes and avoids artificial discretization and edge effects, since it does not employ binning.

5.2 Similarity-based calibration for tabular data sets

To test our first hypothesis, we compared similarity-based calibration to global methods. Figure 1 shows Brier score as a function of available calibration data for the binary classification task of distinguishing two handwritten MNIST digits ("4" vs. "9"). The uncalibrated predictions yielded different starting Brier scores for each classifier (red dashed lines). Platt scaling improved the Brier score for the Naive Bayes (NB) and random forest (RF) classifiers but only marginally for the decision tree (DT). No further improvements were achieved beyond 500 calibration items. Similarity-based calibration (SBA-10, SWC, and SWC-HH) achieved much larger benefits for NB and DT when at least 500 items were used for calibration, and Brier score continued to improve as more calibration data was provided. The random forest, which had the best initial Brier score and therefore less room for improvement, showed an advantage for similarity-based calibration after at least 1500 items were used. SWC-HH yielded a clear advantage at all calibration set sizes for NB and DT, and it also provided a small advantage over SWC for RF with calibration set sizes larger than 1500 items.

SWC and SWC-HH consistently out-performed SBA-10. Since RFprox uses labels to learn the similarity measure, it can elevate the importance of individual features in $\mathcal{X}^+$ (like $\hat{p}[k]$) by placing them higher in individual decision trees within its ensemble. Also, SWC and SWC-HH include evidence from the entire calibration set rather than only the nearest neighbors. Importantly, SBA-10 showed almost no difference in Brier score for different classifiers, ignoring the differences in $\hat{p}$. This data set is represented by 784 features,

[1] Available at `https://github.com/p-lambda/verified_calibration`.



(a) Platt scaling (4 binary) (b) SWC (4 binary) (c) SWC-HH (4 binary data sets)

(d) Temperature scaling (3 multi-class) (e) SWC (3 multi-class) (f) SWC-HH (3 multi-class data sets)

Figure 2: Brier score improvement versus average hidden heterogeneity for four binary (top row) and three multi-class (bottom row) tabular data sets, for six classifier types and 10 random trials.

so the addition of two dimensions in $\mathcal{X}^+$ has little effect. In contrast, the fact that SWC obtained different absolute results for each classifier indicates that RFprox effectively leveraged the $\hat{p}$ features. Experimental results for all classifiers and all tabular data sets, reporting Brier score and accuracy results, are provided in Appendix A (Figures 7 and 8).

5.3 Similarity-based calibration exploits hidden heterogeneity

Our second hypothesis was that HH helps explain why and when similarity-based calibration is effective. We examined Brier score improvement across a large combination of data sets, classifiers, and random trials. We found that large HH can be present even in the absence of domain shift, which creates a large opportunity for local calibration. Figure 2 shows the improvement (reduction) in Brier score after calibration as a function of the average HH across the test set. The four binary data sets were MNIST "1" vs. "7" (relatively easy), "4" vs. "9" and "3" vs. "8" (intermediate), and "3" vs. "5" (difficult). The three multi-class data sets were "mnist10", "fashion-mnist", and "letter". We compared Platt (for binary) or temperature scaling (multi-class) calibration to SWC and SWC-HH for six classifiers, using 10 trials per combination of data set and classifier. SWC and SWC-HH achieved Brier score improvements that correlate strongly with the amount of HH present. The relationship was far weaker for the global Platt/temperature scaling methods, which cannot detect or exploit HH. These results show that HH, which can be computed prior to calibration, is a useful diagnostic indicator that can guide the choice of calibration strategy. When HH is high, it is advisable to employ a similarity-based calibration method like SWC. When there is not much calibration data, global methods such as Platt scaling may be recommended.

To better understand how SWC and SWC-HH improve Brier score, we visualize the calibration process in Figure 3 for the two-dimensional "moons" data set. Each row corresponds to results for a different classifier (linear SVM, decision tree, and random forest). The third column shows the computed HH values. The classifiers were trained on 500 points, calibrated using 1000 points, and Brier score evaluated on 500 points. The linear SVM cannot model the nonlinear decision boundary very well. The diagonal region where the

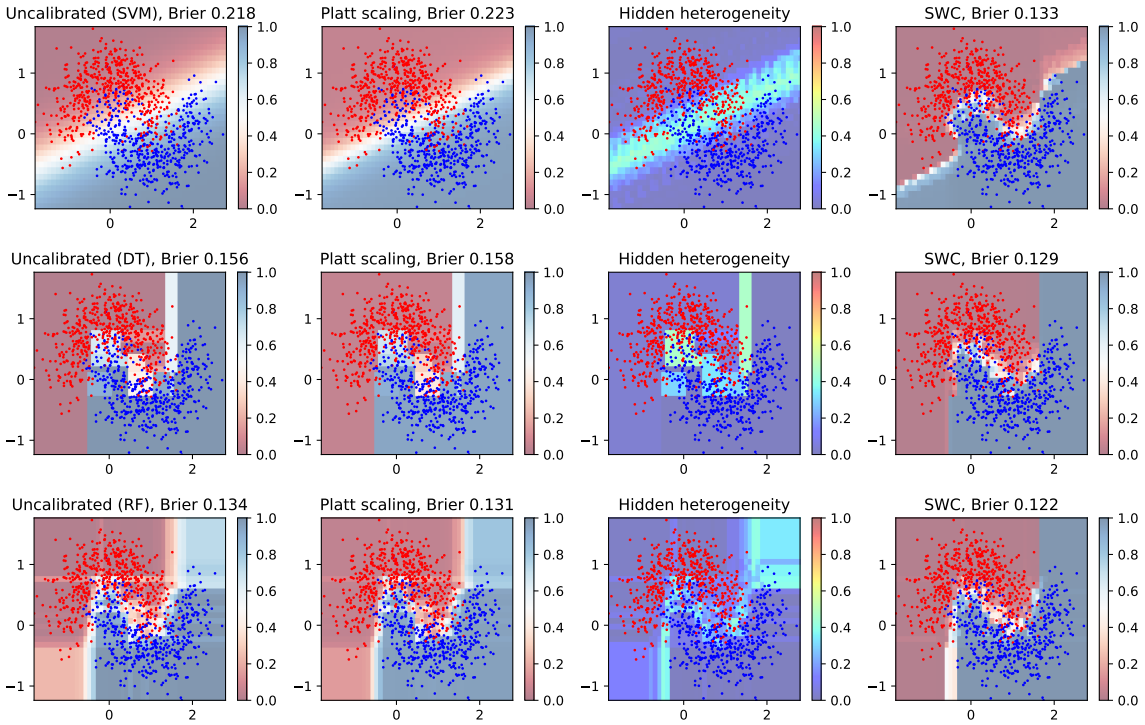


Figure 3: Visualization of uncalibrated predictions (first column) for the "moons" data set, Platt scaling (second column), and SWC (fourth column). The third column shows hidden heterogeneity values that highlight areas of potential improvement, which align with SWC improvements.

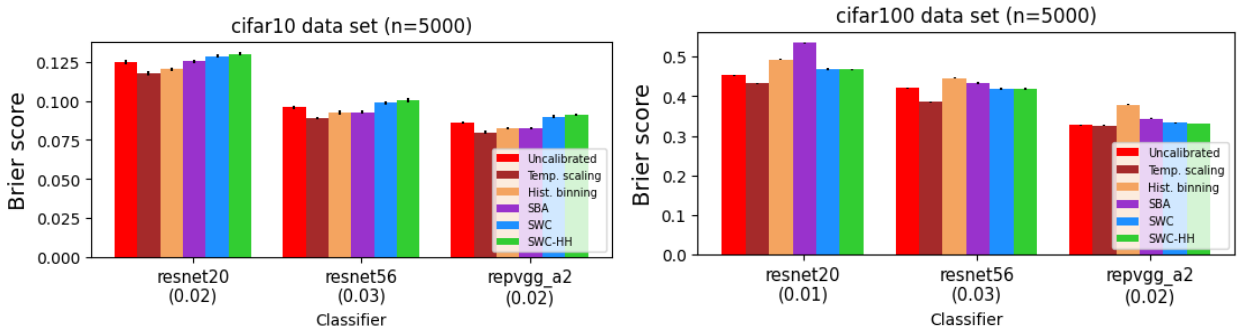
classes are mixed yet separable in the feature space has a high value for HH. When SWC is applied, it is the $\hat{p}$ values in this band that receive the biggest modifications. These changes reduce (improve) the Brier score from 0.218 to 0.133. Platt scaling increases (worsens) the Brier score slightly. The decision tree (second row of Figure 3) exhibits less hidden heterogeneity on the same data set, because it is able to model the nonlinear decision boundary more effectively. SWC improves the Brier score from 0.156 to 0.129. Finally, the random forest (bottom row of Figure 3), somewhat to our surprise, exhibits more areas with hidden heterogeneity. SWC creates smoother regions as the calibration data informs updates to the posterior probabilities and improves the Brier score from 0.134 to 0.122.

Importantly, the difference in results for the rightmost column in Figure 3 demonstrates that SWC adapts (calibrates) most where the classifier exhibits hidden heterogeneity, yielding a result that is customized to the original classifier and more flexible than global calibration.

5.4 Similarity-based calibration for image classifiers

We also conducted calibration experiments with the CIFAR-10 and CIFAR-100 data sets (Krizhevsky, 2009) using three pre-trained neural networks of increasing complexity[2]. ResNet20 (He et al., 2016) has 20 layers and 0.27M parameters, ResNet56 (He et al., 2016) has 56 layers and 0.85M parameters, and RepVGG_A2 (Ding et al., 2021) has 22 layers and 25.49M parameters. For these data sets, the similarity measure *sim* used by SWC and SWC-HH operates in the latent space learned by each network. Specifically, we used the output activations of the `avgpool` (for ResNet models) and `gap` (for RepVGG_A2) layers as a feature vector (dimensionality 64, 64, and 1408 respectively) for similarity computations. We again used a learned RF proximity function to compute similarity. We found that a probability radius of 0.05 yielded reasonably sized neighborhoods for computing HH.

[2]Pre-trained models were obtained from `https://github.com/chenyaofo/pytorch-cifar-models` .



(a) CIFAR-10

(b) CIFAR-100

Figure 4: Calibration performance on CIFAR-10 and CIFAR-100 using three pre-trained neural network over 10 trials (error bars show standard error). HH is shown in parentheses.

Improvements for CIFAR-10 and CIFAR-100 are evident as more complex neural networks are trained; the Brier score consistently decreases from ResNet20 to ResNet56 to RepVGG_A2 (see Figure 4). However, HH values were very low for all three networks ($0.01 - 0.03$), leaving little room for improvement with local calibration. Indeed, we found that global calibration (temperature scaling) yielded the best results for these data sets. Consistent with the results on tabular data shown in section 5.2, calculating HH in advance provides guidance as to which method will be most useful.

There is room for additional improvement. Recent studies of deep network latent spaces suggest that distances computed in neural network latent spaces often do not work well. For example, nearest neighbor classifiers using latent space distances perform substantially worse than the standard multinomial logistic regression (softmax) classifiers (Garrepalli et al., 2022). These latent spaces are also not able to represent dimensions of variation that were poorly represented in the training data (Dietterich & Guyer, 2022). Likewise, decision tree classifiers do not perform well on these learned representations (Garrepalli et al., 2022). Consequently, using random forests to measure hidden heterogeneity in neural network latent spaces may fail to detect HH even though a better representation (or a better classifier) could detect that it is present. In addition, even if HH is large, similarity-based local calibration may not always be able to improve the Brier score due to limitations of the representation. Exploring the connection between choice of representation and calibration efficacy is an important future direction.

5.5 Calibration support highlights calibration data gaps and domain shift

Our third hypothesis is that measuring calibration support, which is a unique capability of similarity-based calibration methods, can provide useful information about the relevance of the calibration set to each item being calibrated. We define the *calibration support* S for item t that informs $\Phi(\hat{p}|t)$ as the sum of similarity weights for items drawn from calibration set $\mathcal{C}$:

$$S_{t,\mathcal{C}} = \sum_i s(t,i), x_i \in \mathcal{C}. \tag{6}$$

Identifying items with low values for $S_{t,\mathcal{C}}$ can draw attention to observations that are not well represented by the calibration set. These could be individual outliers or, if there is a large number of such items, they could indicate distribution shift between the calibration and test sets. Low $S_{t,\mathcal{C}}$ values signal the need for more data (or more representative data) to be added to $\mathcal{C}$. While previous studies focus solely on calibration performance as a function of the total calibration set *size*, similarity-based calibration can characterize the *relevance* of the calibration set to individual test items.

Consider a linear SVM trained on 500 MNIST "1" and "7" digits and calibrated using SWC with 3000 digits. For most items in the test set, calibration improves. However, examining items with low calibration

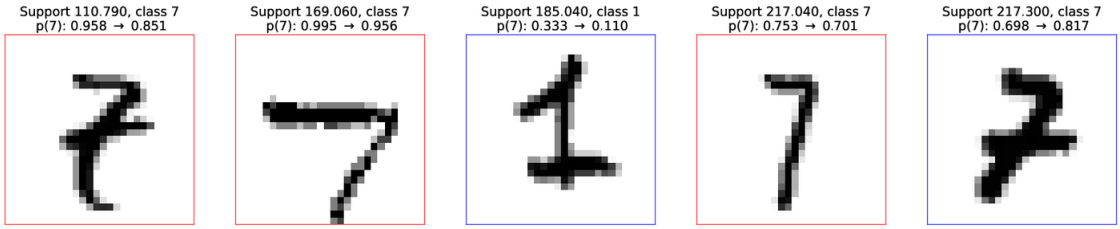
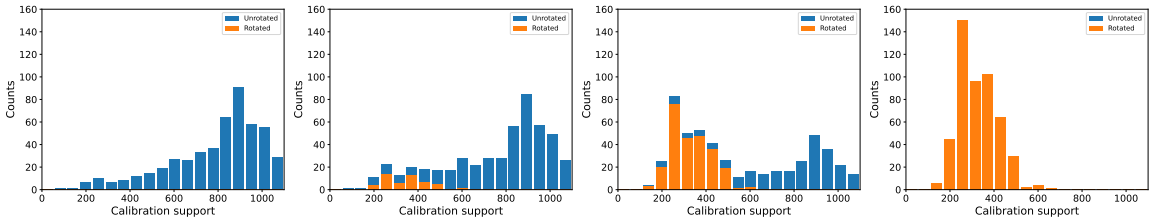


Figure 5: Test items from "mnist-1v7" with the lowest calibration support for a linear SVM classifier. Items framed in blue (red) indicate whether SWC calibration helped (hurt).



(a) 0% rotated (b) 10% rotated (c) 50% rotated (d) 100% rotated

Figure 6: Distribution of calibration support values for 500 test items ("mnist-1v7") classified by a linear SVM with progressively more of the test set items rotated. Bar plots are stacked.

support helps us understand failures. Figure 5 shows the five test items (of 500 total) with the lowest calibration support. Calibration with SWC was beneficial ($\hat{q}[y]$ increased for the true label y) for two of the five and detrimental ($\hat{q}[y]$ decreased) for three of them. These items are not necessarily ambiguous in an objective sense, but the fact that they have low representation in the calibration set signals (correctly) that the calibrated output $\hat{q}$ may be less reliable.

Likewise, low calibration support can provide a warning when domain shift is present between the calibration and test sets (or any new prediction). We simulated domain shift (covariate shift) by rotating a subset of the test items 90 degrees counter-clockwise. Figure 6(a) shows the distribution of calibration support values without any rotation; the peak value is around 900, and items with low support are rare. With 10% of the test set rotated (Figure 6(b)), the overall histogram changes a little, and the rotated items (orange bars) tend to have low calibration support, signaling a need for inspection. With 50% (partial domain shift) rotated (Figure 6(c)), distinct populations for the rotated and unrotated items are clear, and with 100% rotated (complete domain shift), the whole histogram shifts to lower values (peak around 300).

This result suggests that in a deployment setting, it is useful to monitor the calibration support values that are reported by SWC. Knowledge about typical support values (or better, their distribution) could enable early detection of domain shift when new items originate from a changed distribution. That signal indicates that the calibration set, and likely the trained model as well, require revision. Platt, temperature scaling, histogram binning, and other methods provide a fixed mapping $\Phi(\hat{p})$ without regard to the item being calibrated; there is no signal to indicate whether Φ is still relevant. SWC provides an intrinsic measure of calibration set relevance through the calibration support values obtained by each new item.

6 Conclusions, Limitations, and Future work

In this work, we explored the benefits of local, individual calibration for each test item, in contrast to widely used global classifier calibration methods. We identified hidden heterogeneity (HH) as a strong indicator of the need for local calibration, when there are subpopulations within a data set that have the same uncalibrated predicted probability $\hat{p}$ yet require different corrections to achieve a well-calibrated probability $\hat{q}$. We provided a method for calculating HH before calibration to inform selection of the calibration method.

Experiments with tabular data sets and diverse machine learning classifiers indicate that local calibration improves Brier score in proportion to the average hidden heterogeneity (HH) value in the data set. We highlight this finding as an important step towards not only correcting miscalibration but also explaining and understanding it. When HH is very low (as we found with several deep neural networks), or little calibration data is available, global methods such as temperature scaling are sufficient, but otherwise, local calibration is preferred.

We proposed a similarity-based approach to local calibration (SWC) that weights evidence from the calibration set according to its similarity to the test item in an "augmented" feature space that includes both the input features and the predicted class probabilities. This concept goes beyond prior work such as Similarity-Binning Averaging (SBA-10), which calibrates (without weighting) using the 10 nearest neighbors based on Euclidean distance in the augmented feature space (Bella et al., 2009). In most cases, we found that SWC out-performs SBA-10. Additional benefits can be obtained by incorporating HH directly into the SWC algorithm (SWC-HH). A final and unique benefit of similarity-based calibration is that the explicit measurement of calibration support can serve to warn when a given test item lacks good representation in the calibration set. This can also be an indicator when distribution shift or domain shift is present.

Limitations: Runtime. The computational cost of local calibration methods tends to be higher than that of global methods, since each item is independently modeled rather than constructing a single model to apply to all items. However, this also means that calibration can be conducted lazily, as needed, given a similarity measure. The computation of hidden heterogeneity requires (1) the identification of an item's nearest neighbors in the probability simplex Δ_{K-1}, which can be costly with a large number of classes, and (2) training a specialized classifier to estimate the potential Brier score improvement (Equation 3).

Limitations: Preservation of accuracy. SWC and SWC-HH are not rank-preserving calibration methods. This means that in addition to modifying the calibration properties of the predictions, they can also change the predicted class and therefore the accuracy of predictions. Improvements in accuracy are reflected in improved Brier scores. Temperature scaling, in contrast, does preserve rank and accuracy because, without a bias term, it cannot move items to the other side of the decision threshold. Some researchers favor rank-preserving methods (Zhang et al., 2020; Patel et al., 2021), since they seek to improve calibration without sacrificing accuracy. However, this constraint also prevents them from *increasing* accuracy, which is an outcome available to Platt scaling, histogram binning, SWC, etc. On the whole, we agree with Bella et al. (2013) that there is no need to preserve item rankings given the opportunity to improve both accuracy and calibration. However, we acknowledge that in some applications, there could be a need to choose a rank-preserving calibration method to ensure accuracy is unchanged (up or down) for user acceptance (Srivastava et al., 2020).

Future work. There are several important directions for future work. It is possible that within the same data set, some items are best calibrated with global methods while others (where HH is present) benefit from local calibration. A hybrid method that selectively applies global/local calibration, or some combination of the two, for each test item could potentially out-perform either one. In addition, SWC is well designed to naturally accommodate domain shift, if the calibration data set is drawn from the shifted distribution. Sampling bias in the training set, whether intentional or not, induces a particular kind of domain shift that is especially important to address to meet fairness goals when predictions are made in a deployment setting.

References

Amr M. Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 222–232, 2020.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Similarity-binning averaging: A generalisation of binning calibration. In *Intelligent Data Engineering and Automated Learning - IDEAL 2009, Lecture Notes in Computer Science*, volume 5788, pp. 341–349, 2009.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. On the effect of calibration in classifier combination. *Applied Intelligence*, 38:566–585, 2013.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

Leo Breiman, Jerome H. Friedman, R. A. Olshen, and Charles J. Stone. *Classification and regression trees*. Chapman and Hall, New York, NY, 1984.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1): 1–3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

Adele Cutler, D. Richard Cutler, and John R. Stevens. *Ensemble machine learning: Methods and applications*, chapter Random forests. Springer, 2012.

Thomas G. Dietterich and Alex Guyer. The familiarity hypothesis: Explaining the behavior of deep open set methods. *Pattern Recognition*, 132:108931, 2022. doi: 10.1016/j.patcog.2022.108931.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making VGG-style ConvNets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13733–13742, Jun 2021.

Shohei Enomoto and Takeharu Eda. Learning to cascade: Confidence calibration for improving the accuracy and computational cost of cascade inference systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(8), pp. 7331–7339, 2021.

C. A. T. Ferro and T. E. Fricker. A bias-corrected decomposition of the Brier score. *Quarterly Journal of the Royal Meteorological Society*, 138(668):1954–1960, 2012.

Peter W. Frey and David J. Slate. Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, 6:161–182, 1991. doi: 10.1007/BF00114162.

Risheek Garrepalli, Alan Fern, and Thomas G. Dietterich. Oracle analysis of representations for deep open set detection. arXiv, version as of September 22, 2022. URL `https://arxiv.org/abs/2209.11350`.

Yunye Gong, Xiao Lin, Yi Yao, Thomas G. Dietterich, Ajay Divakaran, and Melinda Gervasio. Confidence calibration for domain generalization under covariate shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8958–8967, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017. doi: 10.5555/3305381.3305518.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Sunil Kalmady, Weijie Sun, Justin Ezekowitz, Nowell Fine, Jonathan Howlett, Anamaria Savu, Russ Greiner, and Padma Kaul. Improving the calibration of long term predictions of heart failure rehospitalizations using medical concept embedding. In *Proceedings of AAAI Spring Symposium on Survival Prediction - Algorithms, Challenges, and Applications*, volume 146 of *PMLR*, pp. 70–82, 2021.

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

Kriste Krstovski, David A. Smith, Hanna M. Wallach, and Andrew McGregor. Efficient nearest-neighbor search in the probability simplex. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, pp. 101–108, 2013. doi: 10.1145/2499178.2499189.

Meelis Kull, Telmo M. Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11:5052–5080, 2017.

Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Obtaining well-calibrated multiclass probabilities with Dirichlet calibration. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Rachel Luo, Aadyot Bhatnagar, Yu Bai, Shengjia Zhao, Huan Wang, Caiming Xiong, Silvio Savarese, Stefano Ermon, Edward Schmerling, and Marco Pavone. Localized calibration: Metrics and recalibration. arXiv, version as of December 15, 2021. URL `https://arxiv.org/abs/2102.10809`.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H.S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

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*, pp. 2901–2907, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 625–632, 2005.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An empirical analysis of backward compatibility in machine learning systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3272–3280, 2020. doi: 10.1145/3394486.3403379.

Christian Tomani and Florian Buettner. Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(11), pp. 9886–9896, 2021.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B. Schön. Evaluating model calibration in classification. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

David H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv, version as of August 28 2017. URL `https://arxiv.org/abs/2102.10809`.

Yaodong Yu, Stephen Bates, Yi Ma, and Michael I. Jordan. Robust calibration with multi-domain temperature scaling. In *Proceedings of the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability*, 2022.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the International Conference on Machine Learning*, pp. 609–616, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699, 2002.

Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 11117–11128, 2020.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 11387–11397, 2020.

A Appendix

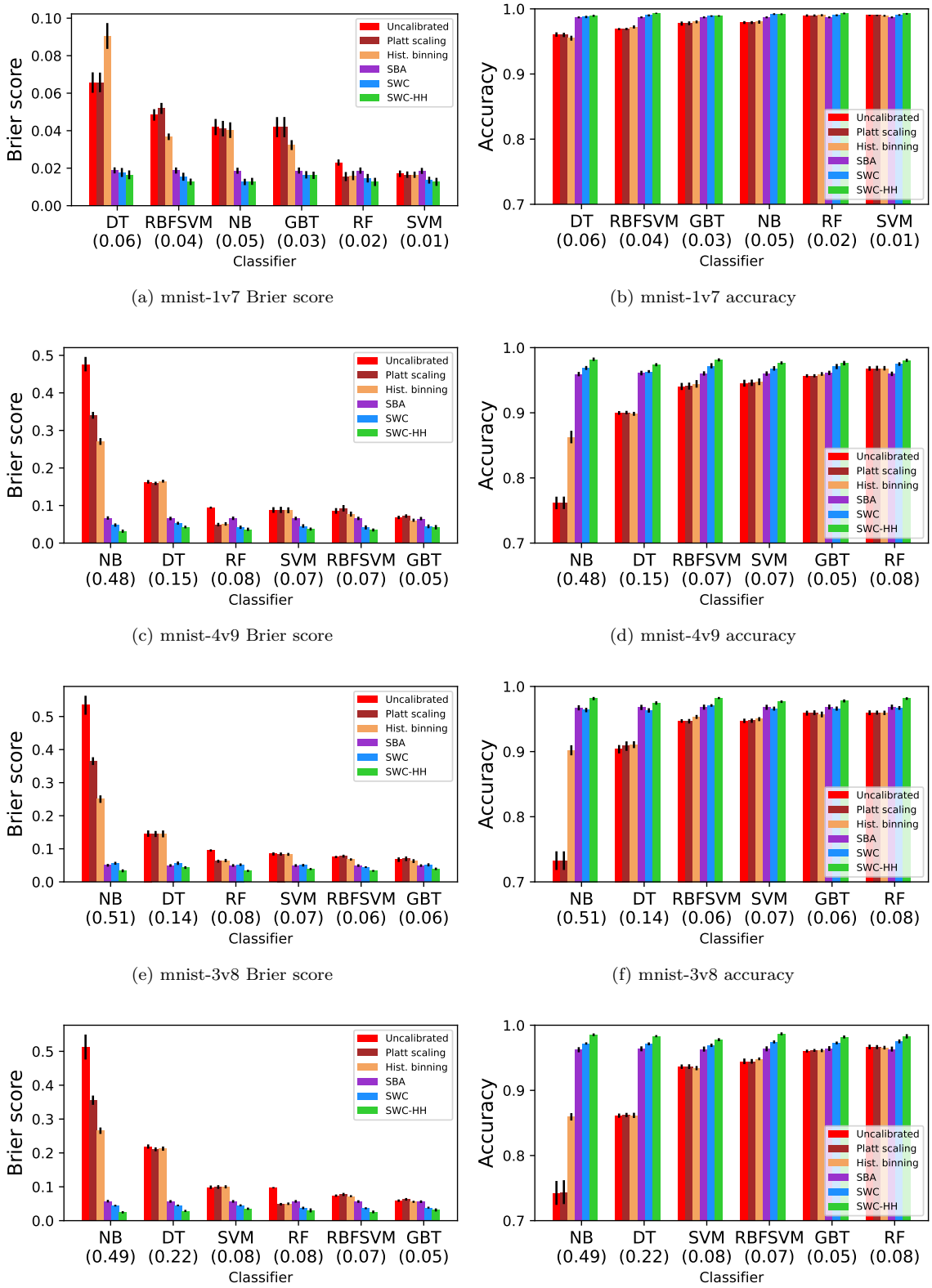
This appendix provides experimental results for seven data sets and six classifiers, comparing similarity-based calibration to other methods. In these experiments, we randomly sampled 10,000 items from each data set and randomly split them into 500 train (for binary problems) or 1000 train (for multi-class problems), 500 test, and 9000 for a calibration pool.

Binary tabular data. Figure 7 presents results across all classifiers and calibration methods for the four binary MNIST data sets after 3000 calibration items were employed. Classifiers appear in order of improving (decreasing) Brier score for the uncalibrated predictions.

Platt scaling improved performance for the Naive Bayes and random forest classifiers, but it yielded little benefit for the others. Histogram binning performed about the same as Platt/TS, except for Naive Bayes models where it was much better except on the (relatively easy) "mnist-1v7" data set. In contrast, similarity-based methods were highly effective for all classifiers in reducing Brier score, with SWC out-performing SBA-10, and SWC-HH providing small additional improvements, especially for the harder MNIST "3v8" and "3v5" problems. SWC/SWC-HH improvements correlate with the average HH value, shown in parenthesis under the x axis labels. In addition, similarity-based calibration also increased test accuracy (see right column of Figure 7). SWC-HH consistently achieved the highest accuracy.

Multi-class tabular data. Figure 8 presents results for the multi-class data sets "mnist10", "fashion-mnist", and "letter", after 5000 calibration items were employed. Temperature scaling improved Brier score for the tree-based methods (RF, GBT) the most, although not consistently. Histogram binning was again beneficial for the Naive Bayes models, but it often made calibration worse for decision trees. In all cases, similarity-based calibration provided the best results. SWC and SWC-HH out-performed SBA-10 in most cases. SWC-HH usually improved over SWC, except for the more challenging "fashion-mnist" data set. In this data set, the filtering employed by SWC-HH (to ignore calibration items with insufficient similarity) often left no calibration items remaining. We handle this case by using the single nearest neighbor, even if its similarity is below the threshold. This leads to values for $\hat{q}$ that are based only on one calibration item. In many cases, the single nearest neighbor belongs to the correct class, yielding good accuracy, but when it is from an incorrect class, the Brier score penalty is large. However, the SWC-HH results were still comparable or better than SBA-10 and the global calibration methods on this data set.

In addition, SWC-HH yielded the best accuracy for all classifiers and data sets with the single exception of Naive Bayes on the "letter" data set, in which case SBA-10 achieved the best accuracy.



(a) mnist-1v7 Brier score

(b) mnist-1v7 accuracy

(c) mnist-4v9 Brier score

(d) mnist-4v9 accuracy

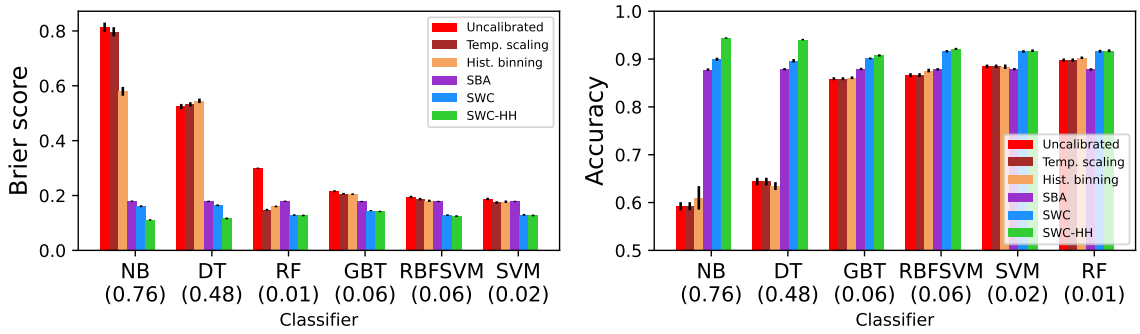
(e) mnist-3v8 Brier score

(f) mnist-3v8 accuracy

(g) mnist-3v5 Brier score

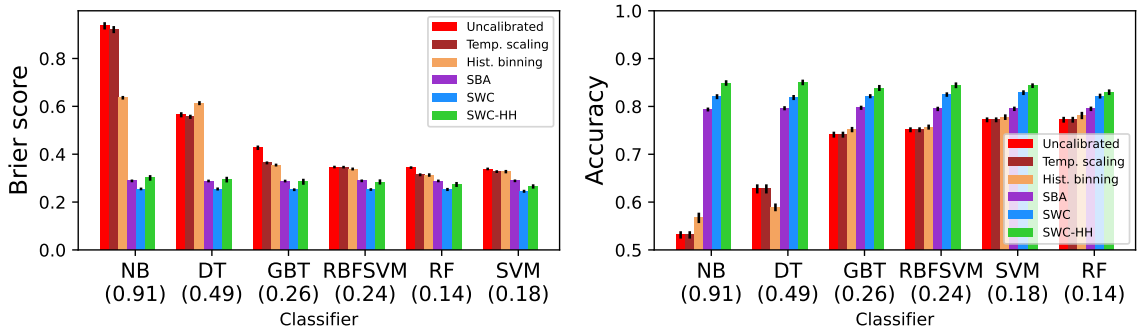
(h) mnist-3v5 accuracy

Figure 7: Calibration performance (left) and accuracy (right) for the binary MNIST data sets (10 trials; error bars indicate one standard error). Classifiers are sorted in order of improvement based on the uncalibrated classifier's score, and average HH values are below each classifier.



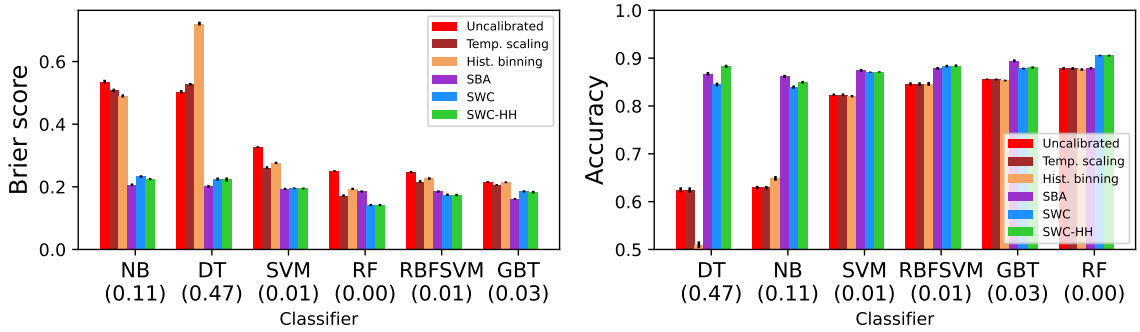
(a) mnist10 Brier score

(b) mnist10 accuracy



(c) fashion-mnist Brier score

(d) fashion-mnist accuracy



(e) letter Brier score

(f) letter accuracy

Figure 8: Calibration performance (left) and accuracy (right) for the multi-class "mnist10", "fashion-mnist", and "letter" data sets (10 trials; error bars indicate one standard error). Classifiers are sorted in order of improvement based on the uncalibrated classifier's score, and average HH values are below each classifier.