On the Role of Dataset Quality and Heterogeneity in Model Confidence

Yuan Zhao[†]   Jiasi Chen[†]   Samet Oymak[⋆]

†: Department of Computer Science and Engineering
⋆: Department of Electrical and Computer Engineering
University of California, Riverside

February 22, 2020

Abstract

Safety-critical applications require machine learning models that output accurate and calibrated probabilities. While uncalibrated deep networks are known to make over-confident predictions, it is unclear how model confidence is impacted by the variations in the data, such as label noise or class size. In this paper, we investigate the role of the dataset quality by studying the impact of dataset size and the label noise on the model confidence. We theoretically explain and experimentally demonstrate that, surprisingly, label noise in the training data leads to under-confident networks, while reduced dataset size leads to over-confident models. We then study the impact of dataset heterogeneity, where data quality varies across classes, on model confidence. We demonstrate that this leads to heterogenous confidence/accuracy behavior in the test data and is poorly handled by the standard calibration algorithms. To overcome this, we propose an intuitive heterogenous calibration technique and show that the proposed approach leads to improved calibration metrics (both average and worst-case errors) on the CIFAR datasets.

1 Introduction

State-of-the-art classifiers such as deep neural networks typically output probability estimates of class categories. Particularly in safety-critical applications, these estimates should reflect the true probability of an accurate prediction. For example, if a collision detection machine learning module on a self-driving car outputs a 0.3 probability of a pedestrian being present, there should indeed be a 30% chance of a pedestrian being there. Similarly, machine learning has growing importance in healthcare, and classifier confidence is of utmost importance for reliable and safe clinical diagnosis. There is a need for self-aware models that know when they are likely to fail.

Typical neural networks today provide uncalibrated confidence estimates, so various calibration schemes have been proposed in the literature [10, 6]. However, such calibration schemes may not perform well with imperfect datasets such as those contaminated with label noise or those with few samples. Such imperfect datasets can occur in practice; for example, manual labelling by humans could result in errors and noise in the training set, or insufficient data collection from battery-powered sensors may lead to fewer than desired samples for training. We posit that a classifier

(a) Standard CIFAR-100 model trained with clean data

(b) CIFAR-100 model trained with noisy data (30% chance of label corruption)

(c) CIFAR-100 model trained with small class sizes (50 labels per class rather than the default 500)
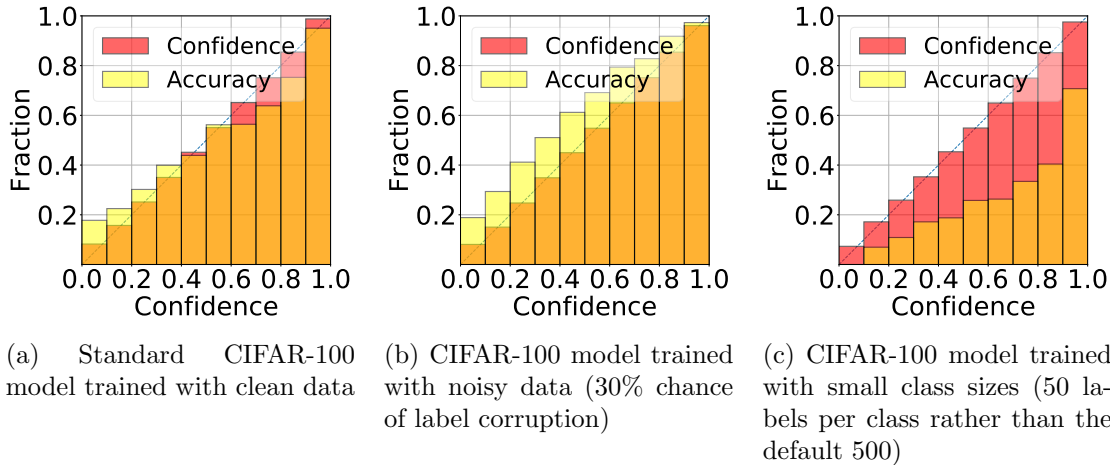
Figure 1: Reliability diagrams of CIFAR-100, generated by binning the test examples by confidence values, and plotting the average confidence (red), average accuracy (yellow), and their overlap (orange). With perfect calibration, the accuracy should align exactly with the confidence along the diagonal. The model (a) trained with clean data is over-confident (higher confidence than accuracy); (b) trained with noisy data is under-confident (lower confidence than accuracy); and (c) trained with small data size is more over-confident than (a).

trained under such scenarios should be aware of these imperfections and able to output calibrated probabilities reflecting these imperfections.

Furthermore, popular calibration schemes such as temperature scaling [26] typically try to find optimal global parameters that are used to calibrate all samples uniformly (e.g. using a single calibration parameter). However, complex multi-modal datasets are often heterogeneous and thus a *one size fits all* approach may be ineffective. For instance, we observe that a classifier may be over-confident in certain classes and under-confident in some others (discussed in Sections 3 and 5). Global calibration may fail to treat such heterogeneities, leading to worse calibration performance and uncertainty quantification. The per-class calibration differences become particularly evident when the individual classes have varying data quality, frequency, and importance [20, 22, 15]. For example, if the collision detection model has few samples of pedestrian collision but many samples of trash collision, the error of the uncertainty estimate may be high for the pedestrian class, which is undesirable as pedestrian collision is a very important class to predict.

In this work, we explore heterogeneous datasets with varying levels of noise or sample sizes across classes, and investigate per-class calibration both empirically and theoretically. In addition to standard calibration error metrics used in the literature, we also evaluate the calibration success through an intuitive fairness-inspired metric, which captures the worst-case (maximum) calibration error across classes. Our goals are to (a) understand the role of data quality in model confidence via theory and experiments, (b) empirically demonstrate the challenges of calibrating classifiers trained on class-wise heterogenous data, and (c) propose intuitive and effective solutions to correct for this imbalanced calibration. Specifically, our contributions are as follows:

• **Understanding the role of the data quality.**
Noise imbalance: We find that label noise in the training data leads to under-confident classifiers (Figure 1b), and we provide a theoretical justification explaining this observation. This is in con-

2

trast, surprisingly, to the over-confidence of deep networks trained with noiseless data previously observed [10, 12].

Sample imbalance: Training sample size similarly has a major effect on classifier confidence. Specifically, in CIFAR100 experiments, a smaller sample size leads to more overconfident classifiers due to lower accuracy (Figure 1c).

Both of these observations, interestingly, also apply to classifiers trained on heterogenous data. For instance, if the label noise is unbalanced across classes (e.g. some classes have more noise compared to others), the classifier tends to be under-confident on noisier classes and over-confident on cleaner classes (see Figures 3 and 4).

• **Calibration for heterogenous data.** These observations motivate us to investigate class-wise calibration algorithms. We propose an intuitive and general approach that allows for individually calibrating each class. Specifically, we slice the validation set by predicted class assignments and calibrate each slice separately. Our approach, coupled with temperature scaling method (TS), leads to class-wise temperature scaling (CTS) as a special case. We establish validation sample complexity bounds and demonstrate the benefit of this approach when the classes exhibit noise and sample size imbalances. We also show the benefits of CTS over an alternative approach, vector scaling [10].

1.1 Related Works

Uncertainty quantification has a long history in several fields spanning medical diagnosis, meteorology, and risk analysis [23, 24, 14, 26]. Due to widespread use of deep learning, there is a growing algorithmic need for training properly calibrated deep networks that are aware of their limitations and can know when they are likely to fail.

It has been empirically observed that modern deep networks tend to be over-confident, i.e. they are more confident than they are accurate, and there is a growing theoretical understanding of this phenomena. Specifically, [8, 21, 2, 4, 3] show that large capacity deep networks have the ability to fit any training dataset and achieve 100% accuracy (a.k.a. interpolation), leading to very high test confidence [12, 10]. Several works [10, 19, 22, 6, 29, 17, 13, 25, 18] aim to address this calibration challenge. [27, 11] relates machine learning fairness to the calibration problem and explores the associated tradeoffs. [28] studies model uncertainty under dataset shift and provides large-scale empirical comparison of different calibration techniques. [16] and [17] provides alternative methods improving over Platt scaling.

Calibration is particularly important for detecting rare events and anomalies which is connected to both noisiness and sample imbalance. [20, 22, 7] proposes algorithms for confidently discovering out-of-distribution samples by modifying the training process or input samples. In contrast, we investigate the related problem of class imbalances through a generic approach compatible with standard calibration algorithms.

2 Problem Setup

We consider the supervised classification problem with multiple classes, with an emphasis on deep networks. Denote the joint distribution $\mathcal{D}$ of input/output pairs (X, Y) via

$$P(Y, X) = P(Y|X)P(X).$$

Multiclass setup: Input $X \in \mathcal{X}$ and output $Y \in \{1, 2, \ldots, K\}$ are random variables where Y is the true class assignment and $\mathcal{X}$ is the input space. $\hat{Y}$ is the predicted class. Let $f : \mathcal{X} \to [0, 1]^K$

be a multiclass classifier (e.g. a deep neural network) mapping an input to a K dimensional vector of class distributions. For an input X, f outputs a class decision $\hat{Y} = \arg\max_{1 \leq k \leq K} f(X)_k$ with confidence $\hat{P} = f(X)_{\hat{Y}}$, where $f(X)_k$ denotes the kth entry of the output vector. $\hat{Y}, \hat{P}$ are functions of f and X. $\hat{P}_f(X), \hat{Y}_f(X)$ will explicitly highlight this dependence.

Binary setup: For theoretical analysis, we also consider binary classifiers where $Y \in \{0, 1\}$ and $f(X): \mathcal{X} \to [0, 1]$. We set $\hat{Y} = 1$, $\hat{P} = f(X)$ if $f(X) \geq 0.5$ and $\hat{Y} = 0$, $\hat{P} = 1 - f(X)$ if $f(X) < 0.5$.

Throughout, we assume that the classifier f can be decomposed as a softmax (or logistic) function applied to logits f^{lgt} i.e. $f(X) = \text{sftmx}(f^{\text{lgt}}(X))$. This is a natural assumption for modern classifiers such as deep networks. We will study how well model confidence at a test sample captures the model accuracy. The mismatch between the model confidence and accuracy is known as the calibration error and there are multiple metrics to assess it [17, 10, 25]. Our approach is agnostic to the specific calibration metric; however, below we will focus on the Expected Calibration Error (ECE). ECE measures the distance between the model accuracy $\mathbb{P}(Y = \hat{Y})$ and confidence $\hat{P}$ over fixed confidence levels $\hat{P} = p$ of the predicted label. Its continuous version with respect to the ℓ_1 metric is given by

$$\text{ECE}(f) = \text{ECE}(f, \mathcal{D}) = \mathbb{E}_{\hat{P}}\big[|\mathbb{P}(Y = \hat{Y}|\hat{P} = p) - p|\big].$$

This continuous version operates in infinitesimal confidence intervals. The discrete version of ECE circumvents this by using binned confidences as defined below.

Definition 2.1 (Discrete Expected Calibration Error) *Split the interval $[0, 1]$ into M disjoint intervals $(B_i)_{i=1}^M$. The discrete ECE is given by*

$$ECE(f) = \sum_{i=1}^{M} |\mathbb{P}(Y = \hat{Y}|\hat{P} \in B_i) - p|\mathbb{P}(\hat{P} \in B_i).$$

In our experiments, the ECE bins are chosen to be equally spaced which is the common approach in the related literature. Given a dataset $\mathcal{S} = (X_i, Y_i)_{i=1}^n$ with sample size n, we denote the empirical (finite sample) version of ECE by $\text{ECE}(f, \mathcal{S})$ obtained by averaging over the dataset.

We will work with the negative log-likelihood (NLL, i.e. cross-entropy) as the optimization loss function given by $\text{NLL}(f) = -\mathbb{E}[\log(f(X)_Y)]$. NLL is a popular metric for evaluating uncertainty and is minimized iff f exactly captures the true conditional distribution $P(Y|X)$. In contrast to ECE, NLL is a continuous function of the samples. This makes it a natural candidate for calibration optimization [10] and we will optimize NLL loss over the validation set for calibration.

3 Theoretical Insights into Model Confidence and Data Quality

Fig. 1 shows empirically that data quality can greatly affect the model confidence. Specifically, in Fig. 1b, a CIFAR-100 model with noisy data (i.e. 30% chance of label corruption) is trained. Compared to the standard CIFAR-100 model with perfect labels (Fig. 1a), the model with noisy data suffers from under-confidence on the test set. On the other hand, a CIFAR-100 model trained with small class sizes (only 50 labels per class rather than the standard 500 labels) results in over-confident models (Fig. 1c), especially compared to the default CIFAR-100 model (Fig. 1a). Towards explaining these observations, in this section we provide theoretical insights on how model uncertainty changes as a function of the noise level and sample size. Our discussion focuses on binary classification with linear classifiers and minimizes binary NLL for training. Specifically, our classifier f will be parameterized by a vector $\boldsymbol{a}$ and intercept b via

$$f_{\boldsymbol{a},b}(X) = \mathrm{sftmx}(\boldsymbol{a}^T X + b) = \frac{\mathrm{e}^{\boldsymbol{a}^T X + b}}{1 + \mathrm{e}^{\boldsymbol{a}^T X + b}}.$$

3.1 The Role of the Sample Size

First, we explore the role of the dataset size (Fig. 1c) and its impact on model confidence. Deep networks are often trained until they achieve 100% training accuracy [30] which leads to over-confident models. Once a network $f = \mathrm{sftmx}(f^{\mathrm{lgt}})$ achieves 100% accuracy, it will still attempt to push NLL to zero. Loss can be pushed to 0 by scaling up the logits i.e. by letting $\alpha \to \infty$ in the class of functions $\mathrm{sftmx}(\alpha f^{\mathrm{lgt}})$. This eventually leads to classifiers with 100% confidence in the training data as well as the test data. The reason is that as soon as one entry of f^{lgt} is favorable over the others (which is guaranteed to happen except for degenerate distributions/classifiers), letting $\alpha \to \infty$ will lead to 100% prediction confidence. On the other hand, test accuracy highly depends on the sample size and quickly degrades for small datasets. Indeed, the over-confidence in Fig. 1c compared to Fig. 1a (also see Fig. 2 for details on accuracy/confidence) mostly arises from the lackluster test accuracy. The following result formalizes this observation and states that small sample size can provably lead to further over-confidence.

Theorem 3.1 *There exists a distribution $\mathcal{D}$ (with unit ℓ_2 norm input set $\mathcal{X}$) as follows. Generate datasets $\mathcal{S}_1 = (X_i, Y_i)_{i=1}^{n} \overset{i.i.d.}{\sim} \mathcal{D}$ and $\mathcal{S}_2 = (X_i, Y_i)_{i=1}^{30n} \overset{i.i.d.}{\sim} \mathcal{D}$ and fix $R > 0$. Minimize the empirical NLL loss on these datasets to find linear classifiers f_1, f_2 as follows.*

$$f_i = \arg \min_{f \in \{f_{\boldsymbol{a},b} \ | \ \|\boldsymbol{a}\|_{\ell_2} \leq R\}} NLL(f, \mathcal{S}_i).$$

Given precision $\varepsilon > 0$, choose $R \geq 6 \log(50n + \varepsilon^{-1})$. With probability at least $9/10$ (over the proper set $\mathcal{S}_1$ or $\mathcal{S}_2$), we have the following accuracy and confidence behavior.
- *For all inputs $X \in \mathcal{X}$ and $i \in \{1, 2\}$: $\hat{P}_{f_i}(X) \geq 1 - \varepsilon$.*
- *$\mathbb{P}_{\mathcal{D}}(\hat{Y}_{f_1}(X) = Y) \leq 1 - \frac{1}{20n}$ and $\mathbb{P}_{\mathcal{D}}(\hat{Y}_{f_2}(X) = Y) = 1$.*

In the setup above, both large dataset ($\mathcal{S}_2$) and small dataset ($\mathcal{S}_1$) problems lead to arbitrarily high confidence classifiers (over all viable inputs in $\mathcal{D}$); however, the model trained on the small dataset is provably less accurate. The proof idea is constructing a distribution where certain features have low probability, thus requiring more data to learn them.

3.2 The Role of Label Noise

For label noise, we work with a noisy dataset model with a discrete distribution over $\mathcal{X} = \{\boldsymbol{v}, -\boldsymbol{v}\}$.

Definition 3.2 ($\mathcal{D}_{\mathbf{noisy}}(p_+, p_-)$) *Fix a vector $\boldsymbol{v} \in \mathbb{R}^d$ with unit ℓ_2 norm and let $\mathcal{X} = \{\boldsymbol{v}, -\boldsymbol{v}\}$. Fix the noise levels $0 \leq p_-, p_+ \leq 1/2$. Suppose that $\mathbb{P}(X = \boldsymbol{v}) = 1/2$ and the conditional class distributions obey*

$$\mathbb{P}(Y = 1 | X = \boldsymbol{v}) = 1 - p_+ \ and \ \mathbb{P}(Y = 0 | X = -\boldsymbol{v}) = 1 - p_-.$$

The next lemma is a straightforward result that captures the properties of the linear classifier on this noisy data model.

5

(a) 5% sampling rate, 65.8% confidence, 29.5% accuracy

(b) 10% sampling rate, 72.6% confidence, 42.2% accuracy

(c) 60% sampling rate, 83.2% confidence, 76.3% accuracy

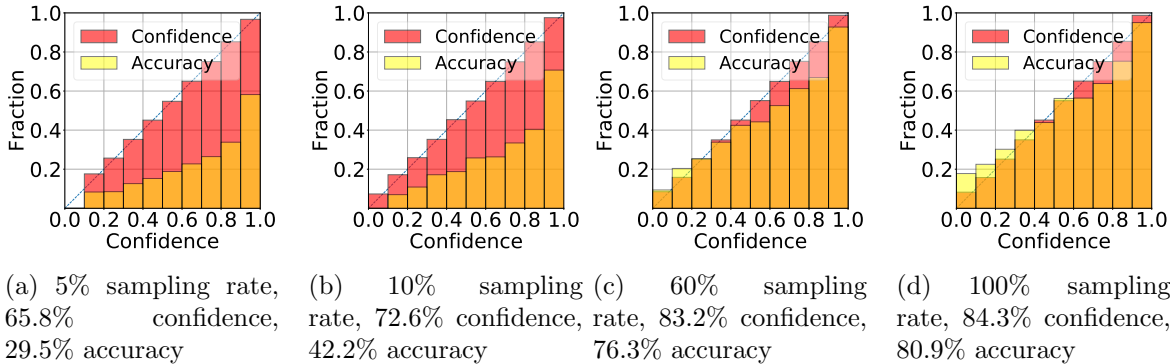(d) 100% sampling rate, 84.3% confidence, 80.9% accuracy

Figure 2: Reliability diagram and over-confidence for different training dataset sizes for CIFAR-100. A smaller training dataset size leads over-confidence. This is because the model accuracy degrades rapidly with fewer samples.

Lemma 3.3 *Fix $1/2 > p_+, p_-, p_{test} \geq 0$. Suppose the data is distributed with $\mathcal{D} = \mathcal{D}_{noisy}(p_{test}, p_{test})$, but the training set is corrupted by label noise in an unbalanced way with distribution $\mathcal{D}_{noisy}(p_+, p_-)$. A linear classifier f minimizing population (infinite sample) training NLL loss obeys*

$$\text{Test confidence over } +: \quad \hat{P}(\boldsymbol{v}) = f(\boldsymbol{v}) = 1 - p_+$$

$$\text{Test confidence over } -: \quad \hat{P}(-\boldsymbol{v}) = 1 - f(-\boldsymbol{v}) = 1 - p_-$$

$$\text{Test accuracy over either: } \quad \mathbb{P}(\hat{Y} = Y | X) = 1 - p_{test}.$$

This lemma highlights that if the training data is noisier than the test (e.g. $p_+, p_- > p_{test}$), the classifier will be under-confident at test time, explaining the behavior in Fig. 1b. It also shows that individual classes or inputs can have different confidence levels as a function of noise.

Remarks on under-confidence: Recall that once a network achieves 100% training accuracy further training will eventually lead to 100% confidence. Thus large capacity and sufficiently trained networks should not be under-confident. On the other hand, for noisy datasets (e.g. Fig. 1b), the training stops before model achieves 100% training accuracy which is the key source of under-confidence[1].

4   Approach for Heterogenous Calibration

To address the above challenges of sample size and label noise resulting in different confidence levels, we propose a new calibration method for heterogeneous datasets. Our approach is summarized in Alg. 1 and applies post-processing on a given classifier f. It can use an arbitrary calibration function $\mathcal{C}$ (chosen from a set $\mathcal{F}_{cal}$) which takes a classifier f and outputs a calibrated classifier $\mathcal{C}(f)$ (e.g. $\mathcal{C}$ applies Platt scaling on f) . The core idea is splitting a heterogenous dataset $\mathcal{S}$ into homogenous subsets so that $\mathcal{C}$ can calibrate each subset individually. The appropriate splitting is a function of the dataset (i.e. its size and type of heterogeneity), and prior information can guide the subset selection. A good example is related to fair machine learning where a dataset may be heterogenous with respect to a sensitive input feature (e.g. race, sex) [27]. We can create the sub-datasets, (e.g. corresponding to different demographic groups) based on the distinct values of the sensitive feature. While our approach can apply to any general splitting policy, in this work, we

---

[1]Maximum validation accuracy is at epoch 61 for Fig. 1b compared to epoch 172 for Fig. 1a.

Algorithm 1 Class-wise Calibration

Inputs: Classifier f, validation dataset $\mathcal{S}$, regularization Γ

Calibration loss function $\mathcal{L}_{\text{calib}}(\cdot)$ (e.g. NLL, ECE), search space $\mathcal{F}_{\text{cal}}$ (e.g. Platt scalings)

Outputs: Calibrated classifier f_{cal}

$\quad \mathcal{S}_k = \{(X,Y) \in \mathcal{S} \mid \hat{Y} = k\}, \ \forall \ 1 \le k \le K.$

$\quad$ Solve the calibration optimization

$$\mathcal{C}_k^\star = \min_{\mathcal{C}_k} \ \sum_{i=1}^{K} \mathcal{L}_{\text{calib}}(\mathcal{C}_k(f), \mathcal{S}_k) \quad \text{s.t.} \tag{CC}$$

$$\|\mathcal{C}_k - \mathcal{C}_0\| \le \Gamma, \ \mathcal{C}_k \in \mathcal{F}_{\text{cal}} \quad \forall \ 0 \le k \le K.$$

$\quad$ For a fresh input sample X, f_{cal} returns $f_{\text{cal}}(X) = \mathcal{C}_{\hat{Y}}^\star(f(X))$
---

restrict our attention to the heterogeneity across different classes and focus on class-wise splitting to address unbalanced class distributions.

Specifically, $\mathcal{S}$ is split into K subsets $(\mathcal{S}_k)_{k=1}^K$ where $\mathcal{S}_k$ is the set of samples whose predicted labels $\hat{Y}$ are class k. Note that, we use *predicted labels* for calibration rather than the actual labels, because at the time of inference, we won't have access to the labels and have to infer them.

Our algorithm takes a calibration loss (e.g. NLL, ECE) and solves the Class-wise Calibration problem (CC). The key idea is individually calibrating each class to obtain $\mathcal{C}_k^\star(f)$ from the base function f. (CC) admits a regularization parameter Γ which quantifies the level of multi-task learning. $\Gamma = 0$ reduces to standard (non-class-wise) calibration whereas $\Gamma = \infty$ means each class is calibrated by themselves which may be more prone to over-fitting. Finally, for inference in test time, the final calibrated classifier f_{cal} calls the sub-classifier $\mathcal{C}_k^\star(f)$ whenever the predicted tag is class k.

Consistency: Suppose $\mathcal{C}$ preserves the decision of the classifier i.e. for all inputs X, we have that $\hat{Y}_f(X) = \hat{Y}_{\mathcal{C}(f)}(X)$. Then, f_{cal} is also consistent with f since it is consistent with $\mathcal{C}_k^\star(f)$'s. Thus proposed method doesn't affect the classifier decision and it is guaranteed to preserve the accuracy of f.

Maximum ECE (Max-ECE): As discussed in Section 5, class-wise calibration improves the calibration error suffered by the worst class. We quantify this via the class-conditional distributions $\mathcal{D}_k = P(Y, X | \hat{Y} = k)$ and exploring the maximum ECE over all predicted classes defined as

$$\text{Max-ECE}(f) = \max_{1 \le k \le K} \text{ECE}(f, \mathcal{D}_k) \tag{4.1}$$

4.1 Class-wise Temperature Scaling (CTS)

Temperature scaling (TS) is a common and successful calibration technique, and is a special case of Platt scaling. TS simply tunes the logits with a single scalar α. Thus, given a range $[\alpha_-, \alpha_+]$, TS searches over a function space $\mathcal{F}$ defined as

$$\mathcal{F} = \{f_\alpha \quad \text{where} \quad \alpha \in [\alpha_-, \alpha_+]\} \quad \text{and} \quad f_\alpha(X) = \text{sftmx}(\alpha f^{\text{lgt}}(X)),$$

that minimizes the calibration error over a validation set $\mathcal{S} = (Y_i, X_i)_{i=1}^n$. Applying Algorithm 1 on TS leads to the Class-wise Temperature Scaling algorithm which applies TS to each class. Given

hyper-parameters $\alpha_-, \alpha_+, \Gamma$, CTS aims to finds parameters $\boldsymbol{\alpha} = (\alpha_i)_{i=0}^K$ over the constraint set

$$\mathcal{A} = \{\boldsymbol{\alpha} \mid |\alpha_j - \alpha_0| \leq \Gamma \quad \text{and} \quad \alpha_- \leq \alpha_0 \leq \alpha_+ \quad \text{for all } 1 \leq j \leq K\},$$

where $\Gamma > 0$ is the multi-task regularization and $\alpha_- \leq \alpha_+$ are the bounds on the shared variable α_0. The k'th class gets the temperature scaling α_k. We minimize the following calibration loss over the validation dataset $\mathcal{S}$ and constraint set $\mathcal{A}$

$$\boldsymbol{\alpha}^\star = \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{L}_{\text{calib}}(\boldsymbol{\alpha}, \mathcal{S}) \quad \text{where} \quad \mathcal{L}_{\text{calib}}(\boldsymbol{\alpha}, \mathcal{S}) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_{\text{calib}}(Y_i, f_{\alpha_{\hat{Y}_i}}(X_i)). \tag{4.2}$$

Here $\alpha_{\hat{Y}}$ means that calibration is done with respect to the predicted label $\hat{Y}$ where the calibrated function is $f_\alpha = \text{sftmx}(\alpha f^{\text{lgt}})$. The following theorem provides a finite sample generalization guarantee for the CTS optimization (4.2) as a function of problem variables.

Theorem 4.1 f^{lgt} *is a bounded function over $\mathcal{X}$ so that $\|f^{lgt}(\boldsymbol{x})\|_{\ell_2} \leq L_f$. Let $\mathcal{L}_{calib}(\cdot, \cdot) : \mathbb{R}^K \times \mathbb{R}^K \to [0,1]$ be a per-sample calibration loss which is L_c Lipschitz in its second variable. Suppose (for simplicity) that $\alpha_+ - \alpha_- \geq 2\Gamma$. Assume we have n i.i.d. validation samples $\mathcal{S} = (X_i, Y_i)_{i=1}^n \overset{i.i.d.}{\sim} \mathcal{D}$ and we solve the CTS calibration problem (4.2). With probability at least $1 - \delta$, the minimizer of empirical risk $\boldsymbol{\alpha}^\star$ achieves small calibration error on test data (i.e. on the distribution $\mathcal{D}$) as follows*

$$\mathcal{L}_{calib}(\boldsymbol{\alpha}^\star, \mathcal{D}) \leq \min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{L}_{calib}(\boldsymbol{\alpha}, \mathcal{D}) + 8\sqrt{\frac{\log(2/\delta)}{n}} + \begin{cases} \frac{4}{\sqrt{n}}\sqrt{\log(1 + 4(\alpha_+ - \alpha_-)\sqrt{n}L_f L_c)} & \text{if } \Gamma \leq \frac{1}{8\sqrt{n}L_f L_c} \\ \frac{4}{\sqrt{n}}\sqrt{\log(\frac{\alpha_+ - \alpha_-}{2\Gamma}) + (K+1)\log(8\Gamma\sqrt{n}L_f L_c)} & \text{else} \end{cases}$$

This result shows the finite sample convergence of CTS for bounded calibration losses and Lipschitz functions. When Γ is small ($\Gamma \leq 8/\sqrt{n}L_f L_c$), the required sample complexity grows as $n \gtrsim \log(\alpha_+ - \alpha_-) + \log L_f + \log L_c$. This quantity is only logarithmic in problem variables and is independent of the number of classes. This is the regime where shared parameter α_0 is dominant over the class-specific residuals $(\alpha_i - \alpha_0)_{i=1}^K$.

When Γ is large, sample complexity grows as $n \gtrsim \log(\alpha_+ - \alpha_-) + K(\log \Gamma + \log L_f + \log L_c)$. This is the regime where classes have more flexibility and sample size is linear in K. In this regime, except K, the dependence on the problem parameters are are again logarithmic. Intuitively the linear dependence on K is not avoidable since there are K variables in the problem. Note that in both scenarios, calibration error increases gracefully in constraints Γ and $\alpha_+ - \alpha_-$.

It is also informative to understand what happens when each class is optimized individually i.e. when each α_k can freely minimize the class-conditional calibration error. This corresponds to setting the regularization $\Gamma = \infty$. The following corollary provides bounds for this setup by solving the calibration problem where $\mathcal{L}_{\text{calib}}$ is same as (4.2)

$$\boldsymbol{\alpha}^\star = \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}'} \mathcal{L}_{\text{calib}}(\boldsymbol{\alpha}, \mathcal{S}) \quad \text{where} \quad \mathcal{A}' = \{\boldsymbol{\alpha} \mid \alpha_- \leq \alpha_j \leq \alpha_+ \quad \text{for all } 1 \leq j \leq K\}. \tag{4.3}$$

This is identical to individually minimizing calibration error over the sub-datasets $\mathcal{S}_k = \{(X_i, Y_i) \in \mathcal{S} \mid \hat{Y}_i = k\}$ by fitting the corresponding α_k's.

Corollary 4.2 *Consider the setup of Theorem 4.1 and solve the calibration problem (4.3) over the validation set $\mathcal{S}$. Define class-conditional distributions $\mathcal{D}_k = P(Y, X | \hat{Y} = k)$ and the minimum class*

probability $p_{\min} = \min_{1 \le k \le K} \mathbb{P}(\hat{Y} = k)$. *With probability at least* $1 - K(\delta + e^{-p_{\min} n/8})$, *the following generalization guarantees hold for all* $1 \le k \le K$

$$\mathcal{L}_{calib}(\alpha_k^\star, \mathcal{D}_k) \le \min_{\alpha_- \le \alpha \le \alpha_+} \mathcal{L}_{calib}(\alpha, \mathcal{D}_k) + 6\sqrt{\frac{\log(1 + 4(\alpha_+ - \alpha_-)\sqrt{n}L_f L_c)}{p_{\min} n}} + 12\sqrt{\frac{\log(2/\delta)}{p_{\min} n}} \quad (4.4)$$

This result establishes test calibration error performance for class-wise distributions $\mathcal{D}_k$ and associated parameters α_k and follows from a standard union bound argument. Assuming $p_{\min} \propto 1/K$, observe that the calibration error grows as $\sqrt{K/n}$ in a similar fashion to the large Γ regime of Theorem 4.1. This is intuitive as each class is calibrated separately i.e. effective sample size for calibration is n/K. Recalling (4.1) and setting $\mathcal{L}_{\text{calib}}$ to ECE, this corollary also bounds Max-ECE.

Finally, recall that Algorithm 1 and CTS mitigates the challenge of small sample size per class via multi-task regularization. An alternative approach for addressing small class sizes would be grouping classes with similar statistics (e.g. by noise level, size, etc) together to obtain larger class bundles with homogenous statistics. Once each bundle contains enough samples, we can safely fit individual calibration models (e.g. temperature scales).

Connection to vector scaling: Vector scaling (VS) is a generalization of the temperature scaling and allows for a more refined access to the softmax layer by using $2K$ parameters for calibration (compared to a single parameter in TS). However this may lead to overfitting in the calibration process [10]. Specifically, VS calibrates over a larger class of functions given by

$$f_{\boldsymbol{a},\boldsymbol{b}} = \text{sftmx}(\boldsymbol{a} \odot f^{\text{lgt}}(X) + \boldsymbol{b}). \quad (4.5)$$

Here $\odot$ is the entrywise product and $f_{\boldsymbol{a},\boldsymbol{b}}$ is parameterized by the K dimensional scaling vector $\boldsymbol{a}$ and bias vector $\boldsymbol{b}$.

Observe that, ignoring the bias $\boldsymbol{b}$, VS is related to CTS, as the entries of $\boldsymbol{a}$ provide knobs on how much we wish to emphasize a particular class. Experimentally, we find that VS is indeed a viable approach for addressing class imbalances. On the other hand, unlike CTS, VS may lead to inconsistency in that the classifier decision may change after calibration. Since each logit is calibrated with its own parameter, the relative order of the calibrated logits may change resulting in different accuracy. We will show that CTS achieves comparable ECE and max-ECE values without compromising the accuracy of the black-box classifier.

5   Numerical Results

5.1   Simulation Setup

Datasets: We use the CIFAR-100 dataset and perform image classification to demonstrate our proposed CTS method (Alg. 1). We also experiment with CIFAR-10 (see Table 2 and Appendix B), and find similar results to CIFAR-100. The CIFAR datasets contains 50k training samples and 10k testing samples. Whenever validation is needed, the original training set is split into 45k training samples and 5k validation samples. We only modify the 45k training samples. The validation set is unmodified, with equal class sizes of 50 samples each. All experiments use standard data augmentation techniques of image shifting and horizontal flipping. Experiments are repeated five times with different random seeds. To evaluate the impact of heterogeneous data, we construct two variants of the CIFAR-100 data:

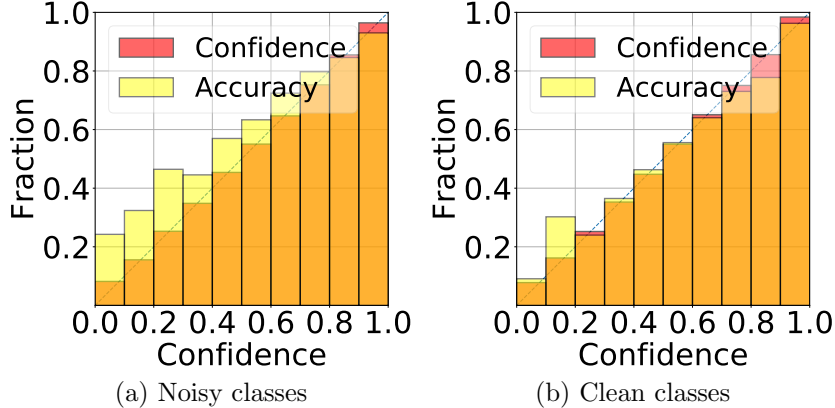(a) Noisy classes   (b) Clean classes

Figure 3: Reliability diagrams for the noisy and clean subsets of the dataset for noisy CIFAR-100 experiment provided in Section 5.3 (classes 0-49 are 30% noisy).

- *Noise-imbalanced dataset construction (Sec. 5.3):* In the training dataset, we add label noise to classes 0 to 49 with the noise rate ρ varying from 0 to 1 (i.e. with ρ probability, a label is randomly assigned to one of the 100 labels). Classes 50 to 99 remained unchanged.
- *Size-imbalanced dataset construction (Sec. 5.4):* We under-sample classes 0 to 49 in the training set, with the sampling rate $\rho \in [0.05, 1]$. Instead of the usual 450 training samples, under-sampled classes have only 450ρ training samples (e.g. $\rho = 0.05$ results in the smaller classes containing only 23 samples, producing a highly unbalanced dataset). The final training set includes the 50 under-sampled classes and the other 50 classes.

Comparison algorithms: We compare the performance of two class-wise approaches (class-wise temperature scaling and vector-scaling) versus two standard approaches that globally apply the same calibration to all samples (temperature scaling and no calibration), as described below. The ECE and max-ECE are computed from the test dataset.

- *Uncalibrated:* The model is not modified after training or calibrated.
- *Temperature Scaling (TS):* TS applies the same calibration parameter α to all classes (see Sec. 4).
- *Class-wise Temperature Scaling (CTS):* CTS uses a per-class calibration parameter α_k (see Sec. 4, Alg. 1). We do not apply multitask regularization (i.e. $\Gamma = \infty$) and instead solve (4.3) with $\alpha_- = 0$, $\alpha_+ = \infty$. In implementation, the predicted labels $\hat{Y}$ are obtained from the arg max of the softmax probabilities. Then $(\alpha_k)_{k=1}^{100}$ on each sub-dataset $(\mathcal{S}_k)_{k=1}^{100}$, where $\hat{Y} = k$, are obtained.
- *Vector Scaling (VS):* For CIFAR-100 data, VS uses 200 parameters to adjust the network's logits as in (4.5).

Metrics: We evaluate the following metrics:

- *ECE:* For VS and TS, ECE is computed as described in (Def. 2.1). For CTS, ECE is computed by merging the calibrated bins: The average confidence and accuracy are calculated from the samples of different predicted labels within bins of the same confidence intervals.
- *max-ECE:* As defined in (4.1), for CTS, max-ECE is computed as the maximum ECE across the 100 predicted labels. For TS and VS, max-ECE is computed by binning the already calibrated samples by their predicted labels and selecting the maximum ECE.

Neural network model and training: To perform image classification, we utilize the WideResNet-
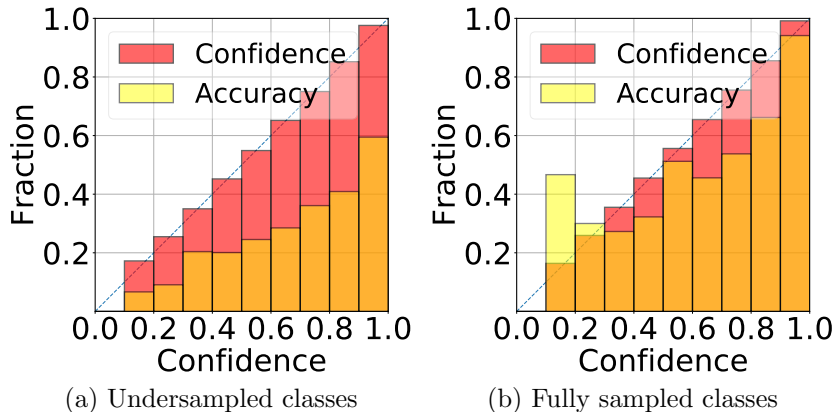
10

(a) Undersampled classes   (b) Fully sampled classes

Figure 4: Reliability diagrams for the undersampled and clean subsets of the dataset for under-sampled CIFAR-100 experiment provided in Section 5.4 (classes 0-49 are 10% undersampled).

28-10 network model. We optimize the differentiable NLL loss for calibration optimization (e.g. fitting TS, CTS, VS) as a proxy for ECE and max-ECE. Note that this is a common practice [10]. 200 training epochs of the WideResNet-28-10 with SGD optimizer were used to fit the data, with cross-entropy as the loss function using Keras and TensorFlow [5, 1]. The initial learning rate is initially set to 0.1 and decreases to 0.02, 0.004, 0.0008 after 60, 120, 160 epochs respectively.

5.2   Model Confidence of Heterogenous Datasets before Calibration

We first study the role of heterogeneity on model confidence at the end of training but before application of any calibration.

Contrasting Noisy vs Clean: Our first experiment explores the noise heterogeneity in the training data. We add 30% label noise on the training data of the CIFAR-100 classes 0-49 and keep the classes 50-99 clean. Figure 3 provides separate reliability diagrams for the noisy and clean subsets of the overall CIFAR-100 dataset at the end of training and before any calibration. In consistency with theoretical intuition, this figure demonstrates that noisy classes tend to be under-confident and clean classes tend to be over-confident. The average accuracy over noisy classes 0-49 is 0.689 and average confidence is 0.627. In contrast, average accuracy over clean classes 50-99 is 0.768 and average confidence is 0.781.

Contrasting Under-sampled vs Full Dataset: Our next experiment explores the heterogeneity on the sample sizes within the classes. We under-sample classes 0-49 at 10% (i.e. 50 per class rather than 500) and classes 50-99 remains untouched. Figure 4 provides reliability diagrams for undersampled vs fully-sampled classes. This figure demonstrates that under-sampled classes tend to be more over-confident than fully-sampled classes. The average accuracy over under-sampled classes 0-49 is 0.396 and average confidence is 0.728. In contrast, average accuracy over fully sampled classes 50-99 is 0.841 and average confidence is 0.909.

These figures indicate that observations of Figure 1 on the effects of noise and undersampling on homogeneous datasets indeed translate to the heterogenous datasets. We emphasize that in Figures 3 and 11b, the subsets of the data that we contrast (clean vs noisy or under-sampled vs fully-sampled) are determined by the actual test labels (rather than the labels predicted by the classifier).
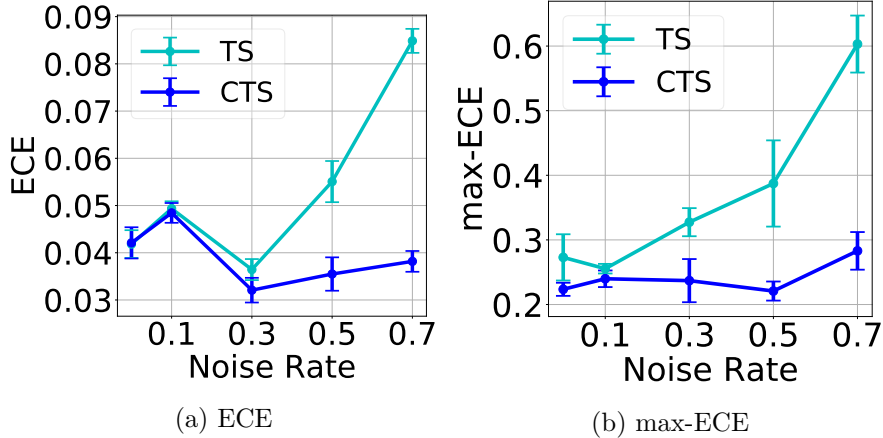
(a) ECE (b) max-ECE

Figure 5: Impact of training data noise on the TS and CTS methods. CTS has lower error in terms of both the ECE and max-ECE metrics.

Alg.	Acc. (%)	ECE (%)	max-ECE (%)
Uncal.	71.53 ± 0.13	3.78 ± 0.30	36.83 ± 4.30
VS	72.82 ± 0.22	3.26 ± 0.21	18.30 ± 1.31
TS	71.53 ± 0.13	3.64 ± 0.22	32.74 ± 2.17
CTS	71.53 ± 0.13	3.21 ± 0.26	23.70 ± 3.34

Table 1: Comparison of class-wise (VS, CTS) and non-class-wise (uncalibration, TS) calibration methods. There is 30% label noise on classes 0-49. CTS and VS have lower ECE and max-ECE than the non-class-wise method, while VS changes the prediction accuracy.

5.3 Noise-Imbalanced Training Data

CTS has lower max-ECE and ECE than TS even with noisy training data. We first evaluate the impact of noisy training data on the calibration error. In Fig. 5, we plot the ECE and max-ECE as we sweep across different noise rates (as described in Sec. 5.1). The CTS method shows a significant improvement (lower ECE and max-ECE) over simple TS, especially when there is more noise in the dataset. These results suggest that not only CTS can achieve better calibration on individual classes (as shown by the max-ECE plot), but it can also result in a better calibrated model from a global perspective (as shown from the global ECE plot).

The class-wise CTS and VS methods outperform non-class-wise methods. VS is another class-wise calibration method that may give better performance than non-class-wise methods, due to better fitting capability to each class (as long as overfitting does not occur). In this set of simulations, we compare the class-wise CTS and VS methods with non-class-wise TS and uncalibrated methods. We construct a training dataset with a 30% label corruption rate for half of the classes (as described in Sec. 5.1). We compare the calibration error of VS, TS and CTS is according to accuracy, ECE, and max-ECE. Table 1 shows the results. In terms of max-ECE, VS is the most preferable, while CTS also has good performance. In terms of ECE, CTS and VS outperform other methods due to their ability to handle heterogeneity via class-wise treatment. Table 2 shows the

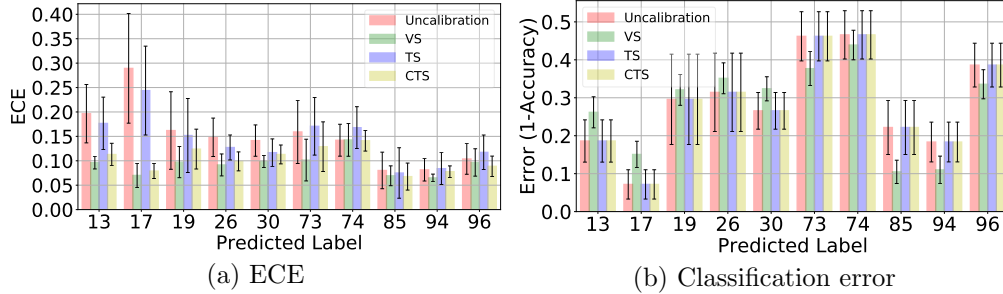|	(a) ECE	(b) Classification error

Figure 6: ECE and accuracy for five random classes (from each of 0-49 and 50-99) are visualized. (a) The class-wise algorithms (CTS, VS) have lower ECE than the traditional methods (uncalibrated, TS) in every class. (b) VS changes the classification accuracy, which may be undesirable.

Alg.	Acc. (%)	ECE (%)	max-ECE (%)
Uncal.	86.98 ± 0.45	6.50 ± 0.46	21.09 ± 0.49
VS	87.27 ± 0.39	1.31 ± 0.11	4.37 ± 0.56
TS	86.98 ± 0.45	3.69 ± 0.33	13.85 ± 0.64
CTS	86.98 ± 0.45	1.19 ± 0.25	5.46 ± 0.98

Table 2: Experiments on CIFAR-10 trained with ResNet-20. 5 classes have 30% label noise and 5 classes are clean. CIFAR-10 experiments show a more substantial benefit of calibration compared to CIFAR-100.

results on CIFAR-10 dataset with similar experimental setup. The improvements due to CTS and TS are noticeably more striking compared to CIFAR-100. We believe this is because CIFAR-10 is a relatively less complex dataset that is easier to (over)fit compared to CIFAR-100. Thus, calibration has a stronger impact.

VS changes the predictions (in a good way!). Aside from the similar calibration performance of VS and CTS, VS *improves the prediction accuracy* by 1.29%, which is a surprising observation. However, modifying the predictions of the original classifier may be undesirable in certain applications, such as those with fairness concerns. For instance, as shown in Fig. 11b, VS uniformly degrades the prediction accuracy over noisy classes (classes 0-49) and uniformly improves the average accuracy over clean classes (classes 50-99). Note that noisy classes are already suffering from lower accuracy due to the noise, and VS ends up amplifying this while improving the overall accuracy. In contrast, by construction the CTS prediction is guaranteed to be consistent with the original classifier as discussed in Sec. 4. We leave further investigation of VS and other calibration techniques that modify the classifier decision as an avenue for future investigation. Fig. 11a breaks down the results from Table 1, and shows that ECE is lower for VS and CTS in every class when compared to TS and no calibration.

5.4 Size Imbalanced Training Data

Smaller training dataset results in over-confidence. To verify the over-confidence on a small dataset, the complete standard training set (50k samples) of CIFAR-100 was undersampled without replacement, with sampling rates ranging from 5%, 10%, 60% to 100%. The reliability diagrams
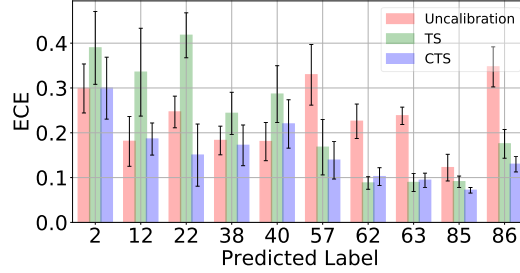
Figure 7: Smaller classes are sampled at 5%. Five random classes from each of 0-49 and 50-99 are visualized. TS favors the large classes (at the expanse of small ones). CTS generally has lower ECE for both small and large classes.

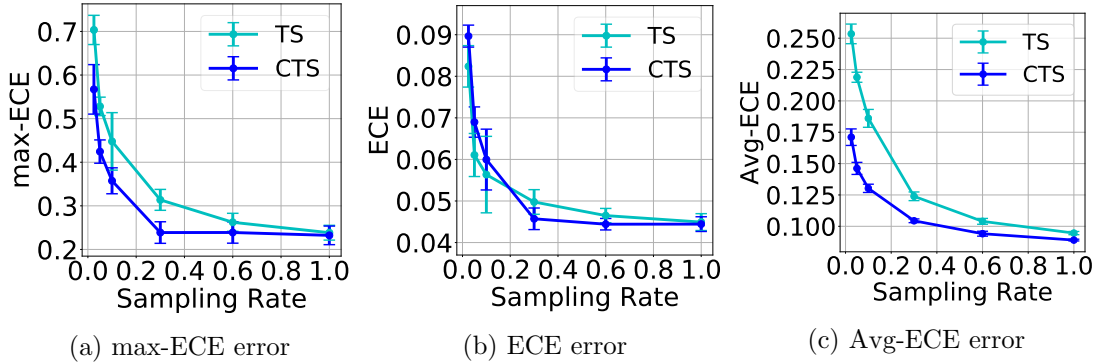(a) max-ECE error   (b) ECE error   (c) Avg-ECE error

Figure 8: Calibration error as a function of the training set sampling rate.

of the uncalibrated model are shown in Fig. 2 (finer resolution than Fig. 1). As the sampling rate decreases, the over-confidence of the model increases. For lower sampling rates, the accuracy is lower (exact numbers are provided in the captions), showing that the over-confidence of the models trained on fewer samples is due to lower accuracy.

CTS improves over the the uncalibrated classifier for imbalanced training data sizes. We next investigate the effectiveness of CTS on the size unbalanced training set. We construct the unbalanced training dataset as described in Sec. 5.1, where smaller classes are 5% as large as the non-down-sampled classes, containing only 27 training samples each. Fig. 7 shows the ECE errors associated with individual classes as labeled by the classifier, i.e. $\mathrm{ECE}_k = \mathrm{ECE}(f, \mathcal{D}_k)$ where $\mathcal{D}_k$ is the conditional distribution $P(Y, X | \hat{Y} = k)$. The results show that CTS provides uniform improvement over original uncalibrated classifier for all classes. In contrast, TS actually inflates the calibration errors of the under-represented smaller classes, while improving the performance over larger classes. This suggests that our proposed CTS provides a more fair treatment of the classes.

A closer look at the ECE metric. To further understand the impact of sample size on calibration error, we plot the ECE as a function of sampling rate in Fig. 8. Fig. 14a shows that CTS outperforms TS in terms of max-ECE metric for all sampling rates, highlighting the fairness benefit of CTS. However, perhaps surprisingly, we find that in terms of overall ECE (where all samples are aggregated), TS sometimes outperforms CTS (Fig. 14b). Upon digging deeper, we found that this is due to the way that individual class confidences output by TS combine in a favorable fashion when they are merged in a given confidence bin, as is done in the overall ECE metric. For example,

suppose there are two equal-sized classes, and fix a confidence bucket, e.g. $[0.4, 0.6]$.

- Suppose Class 1 has average accuracy of 0.52, TS confidence of 0.6, and CTS confidence of 0.54.
- Suppose Class 2 has average accuracy of 0.48, TS confidence of 0.4, and CTS confidence of 0.5.

In this case, TS will achieve $\text{ECE}_1 = \text{ECE}_2 = 0.08$ whereas CTS will achieve $\text{ECE}_1 = \text{ECE}_2 = 0.02$, so CTS seems better. However, CTS is overconfident in both classes whereas TS is perfectly calibrated when both classes are combined, resulting in aggregated $\text{ECE}_{TS} = 0$ and $\text{ECE}_{CTS} = 0.02$.

This indicates that the ECE metric is unable to capture how well each class is doing, and demonstrates the usefulness of worst-case metrics such as max-ECE. On the other hand, it also highlights that optimizing calibration over all samples (i.e. TS compared to CTS with $\Gamma = \infty$) can have favorable properties by aligning the over-/under-confidence directions. Motivated by this, we consider an alternative metric: namely, the average of the individual ECEs, as given by:

$$\text{Avg-ECE} = \frac{1}{K} \sum_{k=1}^{K} \text{ECE}(f, \mathcal{D}_k).$$

The comparison of the Avg-ECE's is shown in Fig. 14c. This figure shows a rather substantial performance difference, and thus highlights how CTS does a much better job than TS when focused on the performance across individual classes.

6   Conclusions

In this paper, we investigated the impacts of noise, sample size, and heterogeneity of the training data on the confidence of the machine learning prediction outputs. We provided empirical evidence and theory showing that noisiness in training data leads to under-confidence and fewer samples leads to over-confidence. To address these class-wise imbalance issues, we proposed class-wise calibration techniques, which, when applied to the temperature-scaling method, leads to our proposed class-wise temperature scaling algorithm. We provided validation sample complexity bounds for our proposed scheme, and our numerical simulations demonstrate the efficacy of these class-wise techniques. Future work includes examining the effectiveness of class-wise calibration beyond temperature scaling and on more complex and wider range of datasets, as well as developing class-wise algorithms for jointly optimizing accuracy and calibration during training.

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[3] Mikhail Belkin, Daniel Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. 06 2018.

[4] Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? 06 2018.

[5] François Chollet et al. Keras, 2015.

[6] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342, 2018.

[7] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[8] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[9] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[10] 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-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[11] Ursula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Calibration for the (computationally-identifiable) masses. *arXiv preprint arXiv:1711.08513*, 2017.

[12] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.

[13] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *arXiv preprint arXiv:1901.09960*, 2019.

[14] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, 2011.

[15] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017.

[16] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

[17] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. *arXiv preprint arXiv:1909.10155*, 2019.

[18] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2810–2819, 2018.

[19] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[20] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

[21] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv preprint arXiv:1903.11680*, 2019.

[22] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[23] Allan H Murphy. A new vector partition of the probability score. *Journal of applied Meteorology*, 12(4):595–600, 1973.

[24] Allan H Murphy and Robert L Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 26(1):41–47, 1977.

[25] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

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

[27] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017.

[28] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980, 2019.

[29] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019.

[30] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations*, 2016.

A   CIFAR-10 Experiments

The remaining experiments and figures are on CIFAR-10 dataset and closely follow the setup of CIFAR100. In our experiments, whenever validation is needed, the original training set is split into 45k training samples and 5k validation samples. We only modify the training data. The validation set is always clean (i.e., not noisy), and validation class sizes are equal with 500 samples each. All experiments use the standard data augmentation. Experiments are repeated five times with different random seeds. To evaluate the impact of heterogeneous data, we construct two variants of CIFAR-10:

- **Noise-imbalanced dataset construction:** In the training dataset, we add label noise to classes 0 to 4 with noise rate ρ varying from 0 to 1. Noise rate ρ means that with ρ probability a label is randomly assigned to one of the 10 labels. Classes 5 to 9 remained unchanged. This results in a partially noisy training set.

- **Size-imbalanced dataset construction:** We under-sample the classes 0 to 4 in the training set, with the sampling rate ρ varying from 0.01 to 1. Instead of the usual 4500 training samples, undersampled classes have only 4500ρ training samples. For instance, with $\rho = 0.01$, the smaller classes contain only 45 samples resulting in a highly unbalanced dataset. The overall training set is obtained by combining the 5 downsampled classes and the other 5 classes. The validation set remain unchanged (same size for every class).

Neural network model: To perform image classification, we utilize the ResNet-20 network model. **Training:** 200 training epochs of the ResNet-20 with Adam optimizer were used to fit the data, with cross-entropy as the loss function using Keras and TensorFlow (Chollet et al. (2015); Abadi et al. (2016)). The initial learning rate is initially set to 10^{-3} and decreases to 10^{-4} after 80 epochs.

The outcomes of the experiments parallel the outcomes of CIFAR-100 experiments. However the calibration has a more dramatic impact on CIFAR-10 as discussed previously (e.g. contrasting Tables 1 and 2). Figure 9 is generated in a similar fashion to and is counterpart of Figure 1. CIFAR-10 experiments on noisy dataset are illustrated by Figures 10 and 11. These figures are created in a similar fashion to Figures 5 and 6. CIFAR-10 experiments on size-unbalanced dataset are illustrated by Figures 12 and 14. Figure 12 is generated in a similar fashion to Figure 2 for different level of under-samplings (size-imbalancedness level). Finally, Figure 14 is created in a similar way to Figure 8.

(a) Standard CIFAR-10 model trained with clean data

(b) CIFAR-10 model trained with noisy data (30% chance of label corruption)

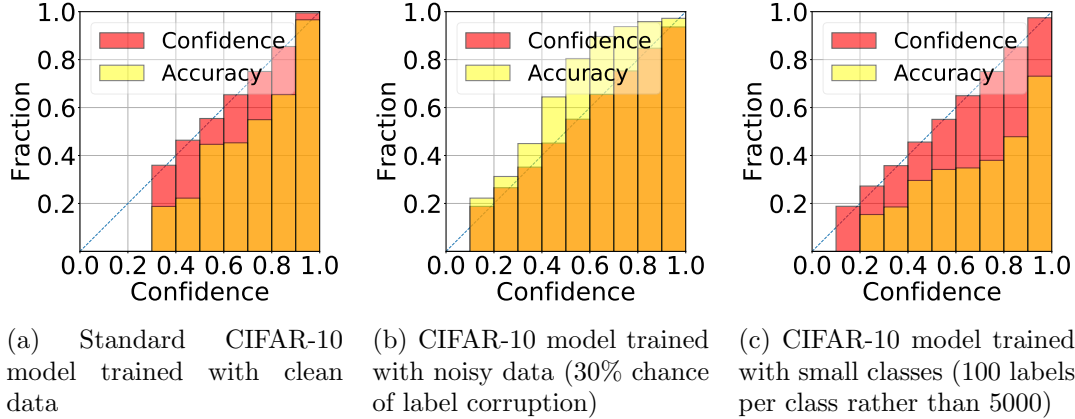(c) CIFAR-10 model trained with small classes (100 labels per class rather than 5000)

Figure 9: Reliability diagrams for CIFAR-10, generated by binning the test results by confidence values, and plotting the confidence (red), accuracy (yellow), and their overlap (orange). With perfect calibration, the accuracy should align exactly with the confidence along the diagonal. The model (a) trained with clean data is over-confident (higher confidence than accuracy); (b) trained with noisy data is under-confident (lower confidence than accuracy); and (c) trained with small data size is more over-confident than (a).
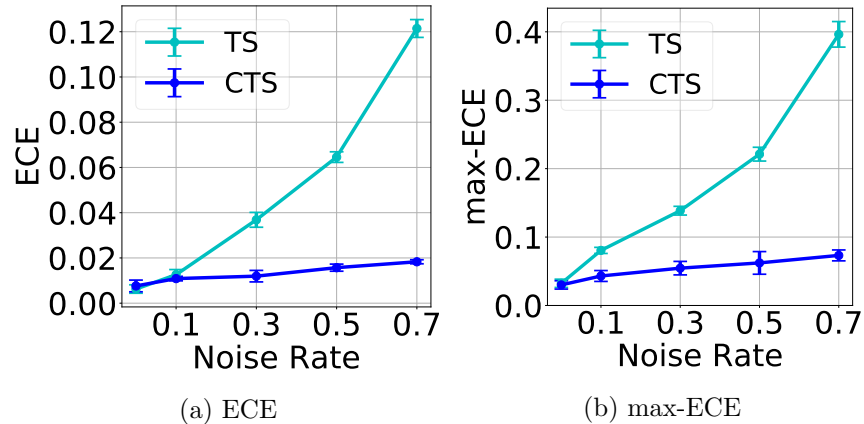

(a) ECE

(b) max-ECE

Figure 10: Noise imbalance: Impact of training data noise on the TS and CTS methods. CTS has lower error in terms of both the ECE and max-ECE metrics.
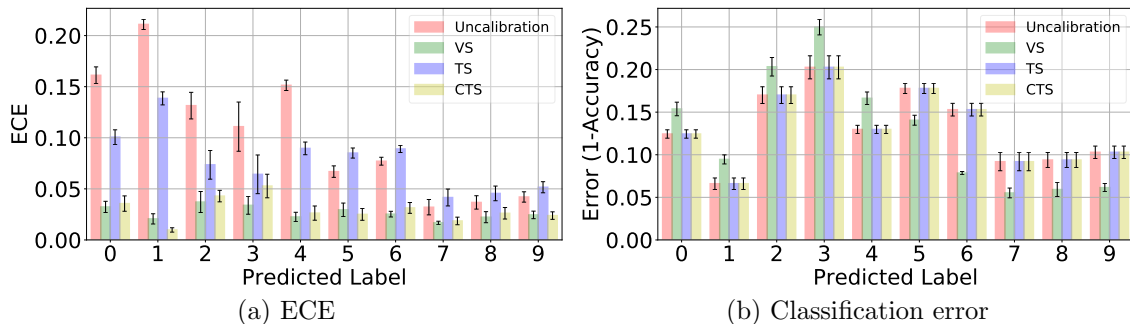

(a) ECE

(b) Classification error

Figure 11: Noise imbalance: Per-class error in terms of ECE and classification accuracy.(a) The class-wise algorithms (CTS, VS) have lower ECE than the traditional methods (uncalibrated, TS) in every class. (b) VS changes the classification accuracy, which may be undesirable.

19

(a) 2% sampling rate, 80.5% confidence, 53.8% accuracy

(b) 10% sampling rate, 90.5% confidence, 75.7% accuracy

(c) 40% sampling rate, 94.7% confidence, 86.9% accuracy

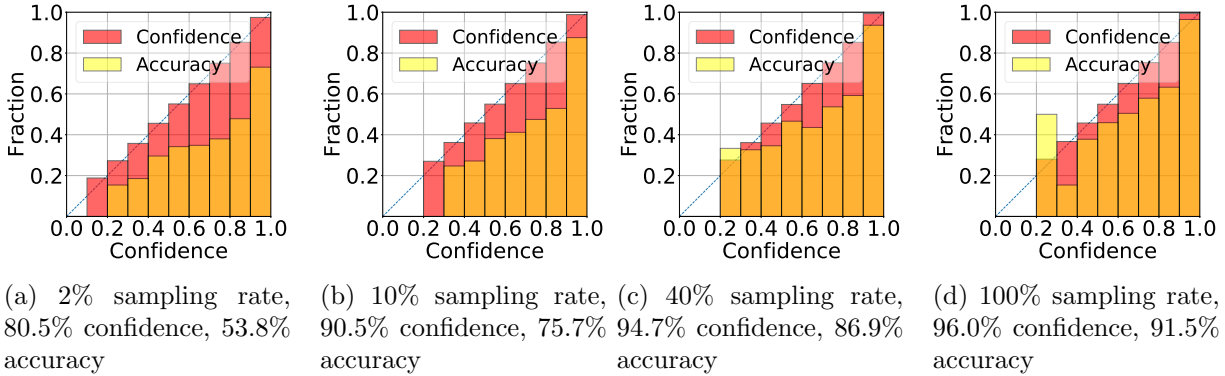(d) 100% sampling rate, 96.0% confidence, 91.5% accuracy

Figure 12: Sample imbalance: Reliability diagram and over-confidence for different training dataset sizes for CIFAR-10. A smaller training dataset size leads to further over-confidence. This is because the model accuracy degrades rapidly with fewer samples.
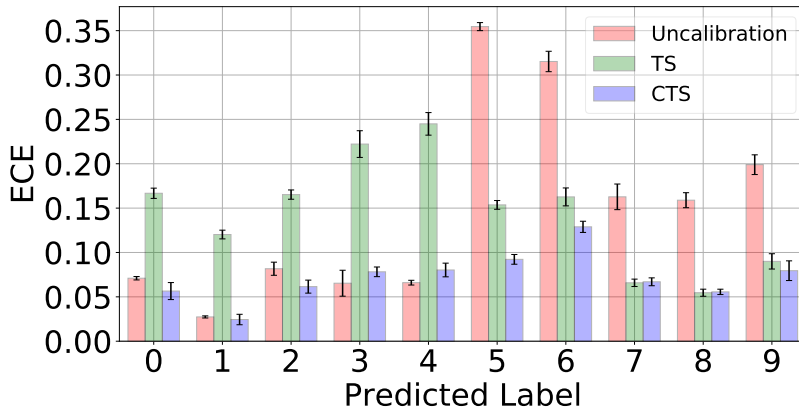

Figure 13: Sample imbalance: Per-class ECE when the first five classes are downsampled at 6%. TS heavily favors the large classes (at the expanse of small ones). CTS generally has lower ECE for both the small and large classes.
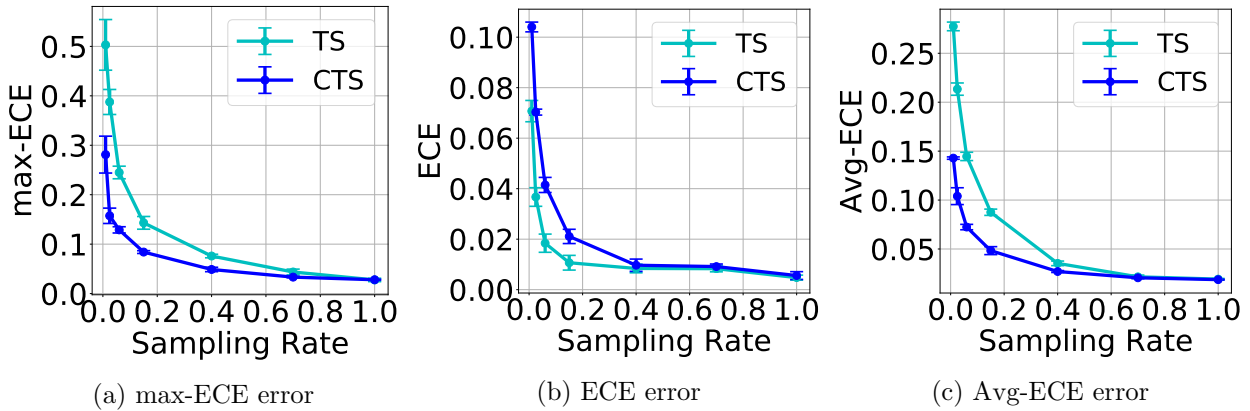

(a) max-ECE error

(b) ECE error

(c) Avg-ECE error

Figure 14: Sample imbalance: Calibration error as a function of the training set sampling rate.

B   Proofs of Main Theorems

B.1   Proof of Theorem 3.1

Below is a restatement of Theorem 3.1 where we provide separate statements for the sets $\mathcal{S}_1$ and $\mathcal{S}_2$.

Theorem B.1 *There exists a distribution $\mathcal{D}$ (with unit ℓ_2 norm inputs $\mathcal{X}$) as follows. Generate datasets $\mathcal{S}_1 = (X_i, Y_i)_{i=1}^n \overset{i.i.d.}{\sim} \mathcal{D}$ and $\mathcal{S}_2 = (X_i, Y_i)_{i=1}^{30n} \overset{i.i.d.}{\sim} \mathcal{D}$ and fix $R > 0$. Minimize the empirical NLL loss to find linear classifiers f_1, f_2 as follows.*

$$f_i = \arg\min_{f \in \{f_{\boldsymbol{a},b} \mid \|\boldsymbol{a}\|_{\ell_2} \leq R\}} NLL(f, \mathcal{S}_i).$$

Given precision $\varepsilon > 0$, we have the following accuracy and confidence behavior over $\mathcal{S}_1$ and $\mathcal{S}_2$.
* *Suppose $R \geq 3\log\max(6, \varepsilon^{-1})$. Then, with probability at least $0.95 - 2e^{-n/100}$, for all $(X, Y) \sim \mathcal{D}$ and $i \in \{1, 2\}$:*

$$\text{For all } X \in \mathcal{X}: \hat{P}_{f_1}(X) \geq 1 - \varepsilon \quad and \quad \mathbb{P}_{(Y,X) \sim \mathcal{D}}(\hat{Y}_{f_1}(X) = Y) \leq 1 - 1/20n.$$

* *Suppose $R \geq 6\log\max(50n, \varepsilon^{-1})$ and $\varepsilon < 1/2$. Then, with probability at least 0.9,*

$$\text{For all } X \in \mathcal{X}: \hat{P}_{f_2}(X) \geq 1 - \varepsilon \quad and \quad \mathbb{P}_{(Y,X) \sim \mathcal{D}}(\hat{Y}_{f_2}(X) = Y) = 1.$$

Proof The NLL (cross-entropy) loss on a dataset $\mathcal{S}$ is given by

$$\text{NLL}(f_{\boldsymbol{a},b}, \mathcal{S}) = -\frac{1}{n}\sum_{i=1}^n \log\left(\frac{e^{y_i(\boldsymbol{a}^T X_i + b)}}{1 + e^{y_i(\boldsymbol{a}^T X_i + b)}}\right) = \frac{1}{n}\sum_{i=1}^n \log(1 + e^{-y_i(\boldsymbol{a}^T X_i + b)})$$

Fix orthogonal unit ℓ_2 norm vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p$. Set $\boldsymbol{v}' = (\boldsymbol{u} + \boldsymbol{v})/\sqrt{2}$. Define the binary distribution $\mathcal{D}$ as follows.

$$\mathbb{P}(Y = 1 | X = \boldsymbol{v}) = \mathbb{P}(Y = 0 | X = \boldsymbol{v}') = \mathbb{P}(Y = 0 | X = -\boldsymbol{v}) = 1 \tag{B.1}$$

$$\mathbb{P}(X = \boldsymbol{v}) = 1/2, \quad \mathbb{P}(X = \boldsymbol{v}') = 1/N, \quad \mathbb{P}(X = -\boldsymbol{v}) = 1/2 - 1/N. \tag{B.2}$$

Let E_i be the event that $\boldsymbol{v}'$ appears as an input in dataset $\mathcal{S}_i$. Observe that

$$e^{-n/N} \geq 1 - \mathbb{P}(E_1) = (1 - 1/N)^n \geq 1 - n/N.$$

Thus, setting $N = 20n$, we find $\mathbb{P}(E_1) \leq 0.05$ and $\mathbb{P}(E_2) \geq 1 - e^{-50n/20n} = 1 - e^{-2.5} \geq 0.91$. Also let B be the event that at least $1/3$ of the training inputs are equal to $\boldsymbol{v}$ and at least $1/3$ are equal to $-\boldsymbol{v}$. Applying a standard Chernoff bound yields that $\mathbb{P}(B) \geq 1 - 2e^{-\frac{n}{100}}$.

Before proceeding further, we also note that for all $x \geq 0$, we have

$$e^{-x}/2 \leq \log(1 + e^{-x}) \leq e^{-x}$$

Analyzing $\mathcal{S}_1$ on the event $E_1 \cap B$: Suppose E_1 holds. Note that the training dataset only contains inputs $\boldsymbol{v}$ and $-\boldsymbol{v}$. Thus, it can be concluded that the optimal classifier has the form $R\boldsymbol{v} + b$

21

for some scalar b i.e. $\boldsymbol{a} = R'\boldsymbol{v}$ for $|R'| \le R$. Let $0 \le \gamma \le 1$ denote the fraction of $+\boldsymbol{v}$ inputs within the training data. The empirical (training) NLL is given by

$$\text{NLL}(f_{\boldsymbol{a},b}, \mathcal{S}_1) = \gamma \log(1 + \frac{1}{\mathrm{e}^{R'+b}}) + (1 - \gamma) \log(1 + \frac{1}{\mathrm{e}^{R'-b}})$$

Minimizing NLL over R' reveals $R' = R$ and the loss is given by

$$\text{NLL}(f_{\boldsymbol{a},b}, \mathcal{S}_1) = \gamma \log(1 + \frac{1}{\mathrm{e}^{R+b}}) + (1 - \gamma) \log(1 + \frac{1}{\mathrm{e}^{R-b}})$$

We next bound the optimal b choice. Under event B, $\gamma, 1 - \gamma \ge 1/3$. Using $\boldsymbol{a} = R\boldsymbol{v}, b = 0$ as an upper bound, we have that

$$\mathrm{e}^{-R} \ge \gamma \log(1 + \frac{1}{\mathrm{e}^{R+b}}) + (1 - \gamma) \log(1 + \frac{1}{\mathrm{e}^{R-b}}) \tag{B.3}$$

$$\ge \frac{1}{3} \log(1 + \frac{1}{\mathrm{e}^{R-|b|}}) \ge \min(\frac{1}{6} \frac{1}{\mathrm{e}^{R-|b|}}, \frac{\log(2)}{3}) \tag{B.4}$$

which implies $|b| \le \log 6$. Now observe that optimal classifier (on training), which obeys $\boldsymbol{a} = R\boldsymbol{v}, |b| \le \log 6$, outputs the wrong decision on $\boldsymbol{v}'$ since

$$\hat{Y}_{f_1}(\boldsymbol{v}') = \text{sign}(\boldsymbol{a}^T \boldsymbol{v}' + b) = \text{sign}(\frac{R}{\sqrt{2}} + |b|) = 1$$

as $R \ge \sqrt{2} \log 6$. This implies $\mathbb{P}(\hat{Y}_{f_1}(X) = Y) \le 1 - 1/20n$. However, confidence on $\boldsymbol{v}'$ (as well as on $\pm \boldsymbol{v}$) is lower bounded as follows

$$\hat{P}_{f_1}(X) \ge \frac{1}{1 + \mathrm{e}^{-(R/\sqrt{2} - \log 6)}} \ge 1 - \mathrm{e}^{-(R/\sqrt{2} - \log 6)} \ge 1 - \varepsilon$$

whenever $R \ge \sqrt{2}(\log 6 + \log(1/\varepsilon))$ which is implied by $R \ge 3 \log \max(6, 1/\varepsilon)$.

Analyzing $\mathcal{S}_2$ on the event E_2: We claim that the classifier achieves small loss on all examples $\boldsymbol{v}, -\boldsymbol{v}, \boldsymbol{v}'$ which will help show the result. First we pick a baseline classifier $\boldsymbol{a} = R\frac{\boldsymbol{v} - 2\sqrt{2}\boldsymbol{u}}{3}$ and $b = 0$. This guarantees that for all $(Y, X) \sim \mathcal{D}$

$$Y X^T \boldsymbol{a} \ge R/3.$$

Thus empirical NLL over $\mathcal{S}_2$ is at most $-\log(\frac{\mathrm{e}^{R/3}}{1 + \mathrm{e}^{R/3}}) = \log(1 + \frac{1}{\mathrm{e}^{R/3}}) \le \mathrm{e}^{-R/3}$. The overall loss will bound the individual losses i.e. at the optimal classifier $(\boldsymbol{a}, b)$ (on training data) for any training example $(X, Y) \in \mathcal{S}_2$ we have

$$\mathrm{e}^{-R/3} \ge \text{NLL}(f_{\boldsymbol{a},b}, \mathcal{S}_2) \ge \frac{-1}{50n}(Y \log f(X) + (1 - Y) \log(1 - f(X)))$$

Since $R \ge 6 \log(50n)$, we find $50n \mathrm{e}^{-R/3} \le \mathrm{e}^{-R/6}$. Without losing generality, let us assume $Y = 1$. This implies

$$\mathrm{e}^{-R/6} \ge -\log f(X) \implies f(X) \ge \mathrm{e}^{-\mathrm{e}^{-R/6}} \implies f(X) \ge 1 - \mathrm{e}^{-R/6}.$$

To achieve $1 - \varepsilon$ probability, we need $\mathrm{e}^{-R/6} \le \varepsilon$ which holds whenever $R \ge 6 \log(1/\varepsilon)$. For $\varepsilon < 1/2$, this also implies the classification is correct i.e. $Y = \hat{Y}$ since $f(X) > 1/2$. The identical argument holds when $Y = 0$. ∎

B.2 Proof of Lemma 3.3

Proof Classifier outputs the probability $f(X) = \frac{e^{\boldsymbol{a}^T X + b}}{1 + e^{\boldsymbol{a}^T X + b}}$. Note that $X = x\boldsymbol{v}$ for $x \in \{-1, 1\}$ hence without losing generality, we can assume $\boldsymbol{a} = a\boldsymbol{v}$ since any direction orthogonal to $\boldsymbol{v}$ has zero inner product with input. Then, classifier simplifies to a single dimension as follows

$$f(X) = \frac{e^{ax+b}}{1 + e^{ax+b}}$$

We need to find a_*, b_* that maximizes the negative NLL loss

$$-\mathcal{L}(\boldsymbol{a}, b) = \mathbb{E}[Y \log(f(X)) + (1 - Y) \log(1 - f(X))]$$

This expectation leads to the scalar optimization

$$-2\mathcal{L}(\boldsymbol{a}, b) = (1 - p_+) \log\left(\frac{e^{a+b}}{1 + e^{a+b}}\right) + p_+ \log\left(\frac{1}{1 + e^{a+b}}\right) + (1 - p_-) \log\left(\frac{e^{a-b}}{1 + e^{a-b}}\right) + p_- \log\left(\frac{1}{1 + e^{a-b}}\right).$$

Note that we can re-parameterize the loss by considering it as a function of $\alpha = a + b$ and $\beta = a - b$. Together it gives

$$-2\mathcal{L}(\alpha, \beta) = (1 - p_+) \log\left(\frac{e^{\alpha}}{1 + e^{\alpha}}\right) + p_+ \log\left(\frac{1}{1 + e^{\alpha}}\right) + (1 - p_-) \log\left(\frac{e^{\beta}}{1 + e^{\beta}}\right) + p_- \log\left(\frac{1}{1 + e^{\beta}}\right).$$

Right hand side is maximized when partial derivatives with respect to α and β are zero i.e.

$$\frac{-2\partial \mathcal{L}(\alpha, \beta)}{\partial \alpha} = \left[(1 - p_+)\frac{1}{1 + e^{\alpha}} - p_+ \frac{1}{1 + e^{-\alpha}}\right)\right]$$

$$\frac{-2\partial \mathcal{L}(\alpha, \beta)}{\partial \beta} = \left[(1 - p_-)\frac{1}{1 + e^{\beta}} - p_- \frac{1}{1 + e^{-\beta}}\right].$$

Note that partial derivative w.r.t. α depends only on p_+ and partial derivative w.r.t. β depends only on p_- which greatly simplifies our life. Proceeding, we find that $\alpha_* = a_* + b_*$ satisfies the likelihood ratio

$$\frac{1 - p_+}{1 + e^{\alpha_*}} - \frac{p_+}{1 + e^{-\alpha_*}} = 0 \implies \frac{1 + e^{\alpha_*}}{1 + e^{-\alpha_*}} = \frac{1 - p_+}{p_+} \tag{B.5}$$

Note that this implies that the classifier output is

$$f(\boldsymbol{v}) = \frac{e^{a_* + b_*}}{1 + e^{a_* + b_*}} = \frac{e^{\alpha_*}}{1 + e^{\alpha_*}} = 1 - p_+.$$

Similarly, following $\beta_* = a_* - b_*$, we find $\frac{1 + e^{\beta_*}}{1 + e^{-\beta_*}} = \frac{1 - p_-}{p_-}$ and $f(-\boldsymbol{v}) = p_-$. On the other hand, this classifier always predicts 1 for $X = \boldsymbol{v}$ and 0 for $X = -\boldsymbol{v}$ (as $1 - p_+ > 1/2$ and $p_- < 1/2$). As a result, since the test data is distributed with $\mathcal{D}_{\text{noisy}}(p_{\text{test}}, p_{\text{test}})$, the test accuracy will be $1 - p_{\text{test}}$ for both classes. ∎

B.3 Proof of Theorem 4.1

Proof Let us start with a change of variable from $\boldsymbol{\alpha}$ to $\boldsymbol{\beta}$ where $\beta_0 = \alpha_0$ and $\beta_i = \alpha_i - \alpha_0$. The goal is centering the α_i terms for simplifying subsequent discussion. Let $\mathcal{A}_{\boldsymbol{\beta}}$ be the constraint set $\mathcal{A}$ adapted to the variable $\boldsymbol{\beta}$ (which is a function of $\boldsymbol{\alpha}$). The proof will utilize a standard covering and union bound arguments. Fix covering resolution $\varepsilon > 0$ and let S_0 be an ε ℓ_2-covering of the set $[\alpha_-, \alpha_+]$ and S_i be an ε covering of the set $[-\Gamma, \Gamma]$. Clearly $|S_0| \le 1 \vee \frac{\alpha_+ - \alpha_-}{\varepsilon}$ and $|S_i| \le 1 \vee \frac{2\Gamma}{\varepsilon}$ where $\vee$ returns the maximum of two scalars. Let $\mathcal{A}'_{\boldsymbol{\beta}}$ be a cover of $\mathcal{A}_{\boldsymbol{\beta}}$ defined as

$$\mathcal{A}'_{\boldsymbol{\beta}} = \{\boldsymbol{\beta} \mid \beta_i \in S_i \quad \text{and} \quad \beta_0 \in S_0 \quad \forall i \ge 1\}.$$

First let us bound the loss between $\boldsymbol{\beta} \in \mathcal{A}_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}' \in \mathcal{A}'_{\boldsymbol{\beta}}$ such that $\boldsymbol{\beta}'$ is an ε-neighbor of $\boldsymbol{\beta}$ in the sense that $|\beta'_j - \beta_j| \le \varepsilon$ for $j \ge 0$. Define $\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}, \hat{Y}) = \mathbb{E}[\mathcal{L}_{\mathrm{calib}}(Y, f_{\alpha_{\hat{Y}}}(X)) | \hat{Y}]$. Note that $\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}, \hat{Y})$ is $L_f L_c$ Lipschitz function of $\alpha_{\hat{Y}}$ via initial assumptions and 1-Lipschitzness of softmax [9]. Consequently

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}, \hat{Y}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{D}, \hat{Y})| \le 2 L_f L_c \varepsilon. \tag{B.6}$$

Taking expectation over $\hat{Y}$, this also implies

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{D})| \le \mathbb{E}_{\hat{Y}} |\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}, \hat{Y}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{D}, \hat{Y})| \le 2 L_f L_c \varepsilon. \tag{B.7}$$

We similarly have $|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{S}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{S})| \le 2 L_f L_c \varepsilon$. What remains is showing the closeness of finite sample and population over the cover which is a standard argument. Observe that each sample is independent and $\mathcal{L}_{\mathrm{calib}}(Y, f_{\alpha_{\hat{Y}}}(X))$ is a random variable bounded by 1. For each $\boldsymbol{\beta}' \in \mathcal{A}'_{\boldsymbol{\beta}}$, Bernstein bound guarantees that with probability at least $1 - 2\mathrm{e}^{-\frac{t^2}{2+t/\sqrt{n}}}$

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}', \mathcal{S})| \le \frac{t}{\sqrt{n}}. \tag{B.8}$$

Observe that cardinality of the cover obeys $|\mathcal{A}'| = |\mathcal{A}'_{\boldsymbol{\beta}}| \le (1 \vee \frac{2\Gamma}{\varepsilon})^K (1 \vee \frac{\alpha_+ - \alpha_-}{\varepsilon})$. Pick $n \ge t^2 = 3\log|\mathcal{A}'| + 3\log(2/\delta)$. Union bounding over the cover, this yields that, for all $\boldsymbol{\beta}' \in \mathcal{A}'_{\boldsymbol{\beta}}$, the bound above holds with probability at least

$$1 - 2|\mathcal{A}'|\mathrm{e}^{-\frac{t^2}{2+t/\sqrt{n}}} \ge 1 - 2|\mathcal{A}'|\mathrm{e}^{-\frac{t^2}{3}} \ge 1 - \delta.$$

Overall, using triangle inequality and combining (B.6) and (B.8), with $1 - \delta$ probability, we obtain that, for all $\boldsymbol{\alpha} \in \mathcal{A}$,

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{S})| \le 4 L_f L_c \varepsilon + \frac{\sqrt{3}\sqrt{K \log(1 \vee \frac{2\Gamma}{\varepsilon}) + \log(1 \vee \frac{\alpha_+ - \alpha_-}{\varepsilon})}}{\sqrt{n}} + \sqrt{\frac{3\log(2/\delta)}{n}}. \tag{B.9}$$

To further simplify, set $\varepsilon = \frac{1}{4\sqrt{n} L_f L_c}$. If $2\Gamma \ge \varepsilon$, (B.9) yields

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{S})| \le 2\sqrt{\frac{\log(\frac{\alpha_+ - \alpha_-}{2\Gamma}) + (K+1)\log(8\Gamma\sqrt{n} L_f L_c)}{n}} + 4\sqrt{\frac{\log(2/\delta)}{n}}. \tag{B.10}$$

If $\varepsilon > 2\Gamma$, using $|\mathcal{A}'| \leq 1 \vee \frac{\alpha_+ - \alpha_-}{\varepsilon}$, the bound (B.9) simplifies to

$$|\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{S})| \leq 2\sqrt{\frac{\log(1 + 4(\alpha_+ - \alpha_-)\sqrt{n}L_f L_c)}{n}} + 4\sqrt{\frac{\log(2/\delta)}{n}}. \tag{B.11}$$

Since these hold for all $\boldsymbol{\alpha} \in \mathcal{A}$, using triangle inequality on minimizer of empirical loss $\boldsymbol{\alpha}^\star$ and the minimizer of population loss, we always have

$$\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}^\star, \mathcal{D}) \leq \min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) + 2 \sup_{\boldsymbol{\alpha} \in \mathcal{A}} |\mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{D}) - \mathcal{L}_{\mathrm{calib}}(\boldsymbol{\alpha}, \mathcal{S})|. \tag{B.12}$$

This concludes the proof after accounting for both cases ((B.10) and (B.11) bounds) in the statement. ∎

B.4   Proof of Corollary 4.2

Proof   The first step is ensuring that each sample set $\mathcal{S}_k$ with predicted labels $\hat{Y} = k$ has at least $p_{\min}n/2$ samples. Note that in expectation the cardinality obeys $\mathbb{E}|\mathcal{S}_k| \geq p_{\min}n$. Hence, applying a standard Chernoff bound followed by a union bound over K classes gives an overall probability of success of at least

$$\mathbb{P}\Big(\bigcup_{k=1}^{K} \{|\mathcal{S}_k| \geq p_{\min}n/2\}\Big) \geq 1 - K \exp\Big(-\frac{p_{\min}n}{8}\Big). \tag{B.13}$$

The rest of the proof is by applying the result of Theorem 4.1 on individual subsets $\mathcal{S}_k$ of the validation set. Each α_k solves the minimization

$$\alpha_k^\star = \arg \min_{\alpha_- \leq \alpha_k \leq \alpha_+} \mathcal{L}_{\mathrm{calib}}(\alpha_k, \mathcal{S}_k).$$

For fixed k (and $\mathcal{S}_k$), we can apply Theorem 4.1 with $K \leftarrow 1$, $\Gamma \leftarrow 0$, $\mathcal{D} \leftarrow \mathcal{D}_k$ to find that, with probability at least $1 - \delta$

$$\mathcal{L}_{\mathrm{calib}}(\alpha_k^\star, \mathcal{D}_k) \leq \min_{\alpha_- \leq \alpha \leq \alpha_+} \mathcal{L}_{\mathrm{calib}}(\alpha, \mathcal{D}_k) + 4\sqrt{\frac{\log(1 + 4(\alpha_+ - \alpha_-)\sqrt{n}L_f L_c)}{n_k}} + 8\sqrt{\frac{\log(2/\delta)}{n_k}},$$

where $n_k = |\mathcal{S}_k|$. Conditioning on $n_k \geq p_{\min}n/2$ via (B.13) and union bounding over all $1 \leq k \leq K$, we conclude with the advertised bound of (4.4). ∎