QUANTIFYING AND MITIGATING THE IMPACT OF LABEL ERRORS ON MODEL DISPARITY METRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Errors in labels obtained via human annotation adversely affect a model's performance. Existing approaches propose ways to mitigate the effect of label error on a model's downstream accuracy, yet little is known about its impact on a model's disparity metrics[1]. Here we study the effect of label error on a model's disparity metrics. We empirically characterize how varying levels of label error, in both training and test data, affect these disparity metrics. We find that group calibration and other metrics are sensitive to train-time and test-time label error—particularly for minority groups. For the same level of label error, the percentage change in calibration error for the minority group is on average 1.5 times larger than the change for the majority group. This disparate effect persists even for models trained with noise-aware algorithms. To mitigate the impact of training-time label error, we present an approach to estimate the *influence* of a training input's label on a model's group disparity metric. We empirically assess the proposed approach on a variety of datasets and find a 10-40% improvement, compared to alternative approaches, in identifying training inputs that improve a model's disparity metric. We complement the approach with an automatic relabel-and-finetune scheme that produces updated models with, provably, improved group calibration error.

1 INTRODUCTION

Label error (noise) — mistakes associated with the label assigned to a data point — is a pervasive problem in machine learning (Northcutt et al., 2021b). For example, 30 percent of a random 1000 samples from the Google Emotions dataset (Demszky et al., 2020) had label errors (Chen, 2022). Similarly, an analysis of the MS COCO dataset found that up to 37 percent (273,834 errors) of all annotations are erroneous (Murdoch, 2022). Yet, little is known about the effect of label error on a model's group-based disparity metrics like equal odds (Hardt et al., 2016), group calibration (Pleiss et al., 2017), and false positive rate (Barocas et al., 2019).

It is now common practice to conduct 'fairness' audits (see: (Buolamwini and Gebru, 2018; Raji and Buolamwini, 2019; Bakalar et al., 2021)) of a model's predictions to identify data subgroups where the model underperforms. Label error in the test data used to conduct a fairness audit renders the results unreliable. Similarly, label error in the training data, especially if the error is systematically more prevalent in certain groups, can lead to models that associate erroneous labels to such groups. The reliability of a fairness audit rests on the assumption that labels are *accurate*; yet, the sensitivity of a model's disparity metrics to label error is still poorly understood. Towards such end, we ask:

what is the effect of label error on a model's disparity metric?

We address the high-level question in a two-pronged manner via the following questions:

1. **Research Question 1**: What is the sensitivity of a model's disparity metric to label errors in training and test data? Does the effect of label error vary based on group size?
2. **Research Question 2**: How can a practitioner identify training points whose labels have the most *influence* on a model's group disparity metric?

[1]Group-based disparity metrics like subgroup calibration, false positive rate, false negative rate, equalized odds, and equal opportunity are more often known, colloquially, as *fairness metrics* in the literature. We use the term group-based disparity metrics in this work.

CONTRIBUTIONS & SUMMARY OF FINDINGS

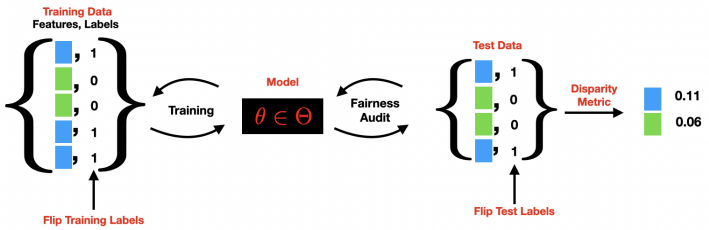

Figure 1: **A schematic of the test and train-time empirical sensitivity tests**. Here we show the model training and fairness audit pipeline. Our proposed sensitivity tests capture the effect of label error, in both stages, on the disparity metric. In the Test-time sensitivity test, we flip the label of a portion of the test data and then compare the corresponding disparity metric (group calibration for example) for the flipped dataset to the metrics for a standard model where the test labels were not flipped. In the Train-time sensitivity test, we flip the labels of a portion of the training set, and then measure the change in disparity metric to a standard model.

In addressing these questions, we make two broad contributions:

Empirical Sensitivity Tests. We assess the sensitivity of model disparity metrics to label errors with a label flipping experiment. First, we iteratively flip the labels of samples in the test set, for a fixed model, and then measure the corresponding change in the model disparity metric compared to an unflipped test set. Second, we fix the test set for the fairness audit but flip the labels of a proportion of the training samples. We then measure the change in the model disparity metrics for a model trained on the data with flipped labels. We perform these tests across a datasets and model combinations.

Training Point Influence on Disparity Metric. We propose an approach, based on a modification to the influence of a training example on a test example's loss, to identify training points whose labels have undue effects on any disparity metric of interest on the test set. We empirically assess the proposed approach on a variety of datasets and find a 10-40% improvement, compared to alternative approaches that focus solely on model's loss, in identifying training inputs that improve a model's disparity metric.

2 SETUP & BACKGROUND

In this section, we discuss notation, and set the stage for our contributions by discussing the disparity metrics that we focus on. We also provide an overview of the datasets and models used in the experimental portions of the paper.

Overview of Notation. We consider prediction problems, i.e, settings where the task is to learn a mapping, $\theta : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^d$ is the feature space, $\mathcal{Y} \in \{0, 1\}$ is the output space, and $\mathcal{A}$ is a group identifier that partitions the population into disjoint sets e.g. race, gender. We can represent the tuple (x_i, a_i, y_i) as z_i. Consequently, the n training points can be written as: $\{z_i\}_{i=1}^n$. Throughout this work, we will only consider learning via empirical risk minimization (ERM), which corresponds to: $\hat{\theta} := \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_i^n \ell(z_i, \theta)$. Similar to Koh and Liang (2017), we will assume that the ERM objective is twice-differentiable and strictly convex in the parameters. We focus on binary classification tasks, however, our analysis can be easily generalized.

Disparity Metrics. We define a group disparity metric to be a function, $\mathcal{GD}$, that gives a performance score given a model's probabilistic predictions (θ outputs the probability of belonging to the positive class) and 'ground-truth' labels. We consider the following metrics:

1. **Calibration**: defined as $\mathbb{P}\left(\hat{y} = y | \hat{p} = p\right), \forall p \in [0, 1]$. In this work, we measure calibration with two different metrics: 1) Expected Calibration Error (ECE) (Naeini et al., 2015; Pleiss et al., 2017), and 2) the Brier Score (Rufibach, 2010) (BS). We refer readers to the Appendix for a detailed overview of these metrics.

Dataset	Classes	n	d	Group	Test acc.	Source
CivilComments	2	$1,820,000$	768	Sex	97.4%	Koh and Liang (2017)
ACSIncome	2	$195,665$	10	Sex, Race	74.5%	Ding et al. (2021)
ACSEmployment	2	$378,817$	16	Sex, Race	73.2%	Ding et al. (2021)
ACSPublic Coverage	2	$138,554$	19	Sex, Race	72.9%	Ding et al. (2021)
Credit Dataset	2	$405,032$	6	Sex	67.4%	De Montjoye et al. (2015)

Table 1: Dataset characteristics and the test accuracies that logistic regression achieves (with regularization λ selected by cross-validation). n is the training set size and d is the number of features.

2. **(*Generalized*) False Positive Rate (FPR)**: is $\mathcal{GD}_{\text{fpr}}(\theta) = \mathbb{E}[\theta(x_i) \mid y_i = 0]$ (see Guo et al. (2017)),
3. **(*Generalized*) False Negative Rate (FNR)**: is $\mathcal{GD}_{\text{fnr}}(\theta) = \mathbb{E}[(1 - \theta(x_i)) \mid y_i = 1]$,
4. **Error Rate (ER)**: is the $\mathcal{GD}_{\text{er}}(\theta) = 1 - \text{acc}(\theta)$.

We consider these metrics separately for each group as opposed to relative differences. For each dataset, we consider the protected data subgroup with the largest size as the majority group, and the group the smallest size is the minority group.

Datasets. We consider datasets across different modalities: 4 tabular, and a text dataset. A description of these datasets along with test accuracy is provided in Table 2. Each dataset contains annotations with a group label for both training and test data, so we are able to manipulate these labels for our empirical sensitivity tests. For the purposes of this work, we assume that the provided labels are the ground-truth—a strong assumption that nevertheless does not impact the interpretation of our findings.

Model. We consider three kinds of model classes in this work: 1) a logistic regression model, 2) a Gradient-boosted Tree (GBT) classifier for the tabular datasets, and 3) a ResNet-18 model. These three classes of models allow us to also test whether the sensitivity of a model's disparity metric depends on the amount of overparametrization and the model class (i.e. Logistic regression vs Neural Networks). To train CNNs on tabular data, we first apply a simple 2-layer MLP on the tabular dataset. We then use hidden representations from the MLP (reshaped and normalized) as inputs to deepnet, which follows the strategy discussed in Borisov et al. (2021). We refer readers to the Appendix for details of model training and hyper-parameters.

3 Empirical Assessment of Label Sensitivity

In this section, we perform empirical sensitivity tests to quantify the impact of label error on test group disparity metrics. We report here results on a logistic regression model for subgroup calibration error (ECE) since these results demonstrate the critical empirical insights we observed from the tests. In addition, we provide a high-level summary of results on other model classes.

We conduct tests on data from two different stages of the ML pipeline: 1) Test-time (test dataset) and 2) Training-time (training data). We use as our primary experimental tool: label flipping, i.e., we flip the labels of a percentage of the samples, uniformly at random in either the test or training set, and then measure the concomitant change in the model disparity metric. We assume that each dataset's labels are the ground truth and that flipping the label results in label error for the samples whose labels have been overturned. Recent literature has termed this setting synthetic noise, i.e., the label flipping simulates noise that might not be representative of real-world noise in labels (Arpit et al., 2017; Zhang et al., 2021; Jiang et al., 2020).

3.1 Sensitivity to Test-time Label Error

Overview & Experimental Setup. The goal of the test-time empirical test is to measure the impact of label error on the group calibration error of a fixed model. Consider the setting where a model has been trained, and a fairness assessment is to be conducted on the model. What impact does label error, in the test set used to conduct the audit, have on the calibration error on the test data? The test-time empirical tests answer this question. Given a fixed model, we iteratively flip a percentage of

the labels, uniformly at random, ranging from zero to 30 percent in the test data. We then estimate the model's calibration using the modified dataset. Critically, we keep the model fixed while performing these tests across each dataset.

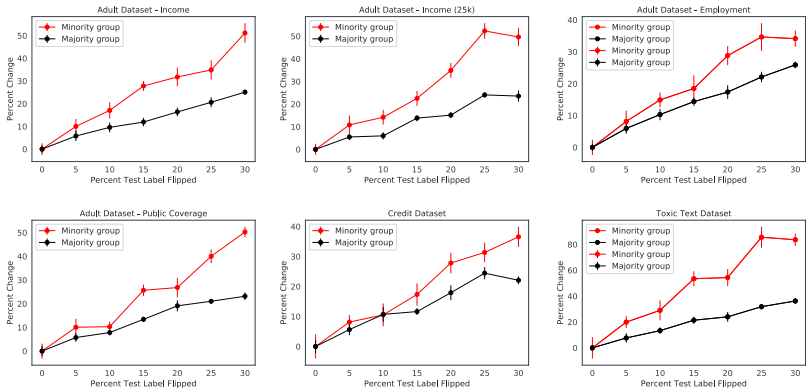

Figure 2: **Test-time Label Flipping Results across 6 datasets**. For each dataset, we plot the percent change in calibration error versus the corresponding percentage change in label error. Here, we plot the minority (smallest) group as well as the majority (largest) group. These two groups represent two ends of the spectrum for the impact of label error. We observe that across all datasets, the minority group incurs higher percentage change in group calibration compared to the majority group.

Results. In Figure 2, we report results of the label flipping experiments across 6 tasks. On the horizontal axis, we have the percentage of labels flipped in the test dataset, while on the vertical axis, we have the percentage change in the model's calibration. For each dataset, we compute model calibration for two demographic groups in the dataset, the majority and the minority—in size–groups. We do this since these two groups constitute the two ends of the spectrum in the dataset. As shown, we observe a more distinctive effect for the minority group across all datasets. This is to be expected since flipping even a small number samples in the minority group can have a dramatic effect on test and training accuracy within this group. For both groups, we observe a super-linear effect on the calibration error. For example, for the Income prediction task on the Adult dataset, a 10 percent label error induces a 20 percent change in the model's test calibration error. These results suggest that test-time label error has more pronounced effects for minority groups. Further, these results imply that - for calibration error - one can expect a change at least as large as the percentage of label error present in the test dataset. Similarly, we observe for other disparity metrics across all model classes that increases in percentage of labels flipped disproportionately affects the minority group (See Appendix E).

3.2 Sensitivity to Training Label Error

Overview & Experimental Setup. The goal of the training-time empirical tests is to measure the impact of label error on a trained model. More specifically, given a training set in which a fraction of the samples' labels have been flipped, what effect does the label error have on the calibration error compared to a model trained on data without label error? We simulate this setting by creating multiple copies of each of the datasets where a percentage of the training labels have been flipped uniformly at random. We then assess the model calibration of these different model using the same fixed test dataset. Under similar experimental training conditions for these models, we are then able to quantify the effect of training label error on a model's test calibration error. We conduct this analysis across all dataset-model task pairs.

Results & Implications. We show the results of the training-time experiments in Figure 3. Similar to the test-time experiments, we find minority groups are more sensitive to label error than larger groups. Specifically, we find that even a 5 percent label error can induce significant changes in the disparity metrics, of a model trained on such data, for these groups.

A conjecture for the higher sensitivity to extreme training-time error is that a model trained on significant label error might have a more difficult time learning patterns in the minority class where

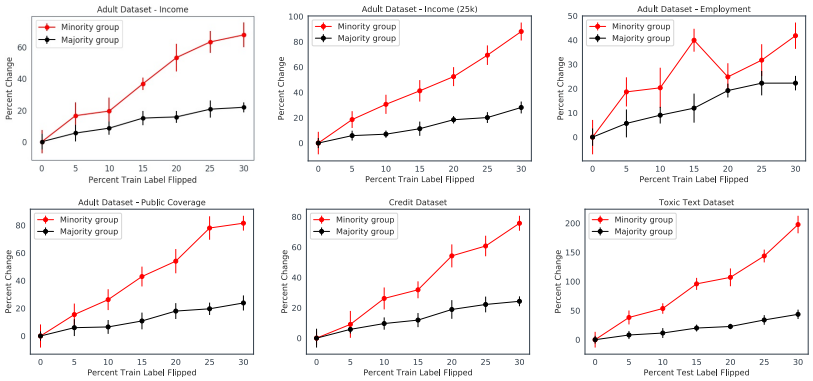

Figure 3: **Training-time Label Flipping Results across 6 datasets**. For each dataset, we plot the percent change in calibration error versus the corresponding percentage change in label error for the training set. Here, we plot the minority (smallest) group as well as the majority (largest) groups by size. Similar to the test-time setting, we observe that across all datasets, the minority group incurs higher percentage change in group calibration compared to the majority group. However, we observe a larger magnitude change for the minority groups.

there are not enough samples to begin with. Consequently, the generalization performance of this model worsens for inputs that belong to the minority group. Alternatively, in the majority group, the proportion of corrupted labels due to label error is smaller. This might mean that uniform flipping does not affect the proportion of true labels compared to the minority group. Even though the majority group exhibits label error, there still exists enough samples with true labels such that a model can learn the underlying signal for the majority class.

A second important finding is that overparameterization seems to confer more resilience to training label error. We find that for the same levels of training label error, an overparametrized model is less sensitive to such change compared to a model with a smaller number of parameters. Recent work suggests that models that learn functions that are more aligned with the underlying target function of the data generation process are more resilient to training label error (Li et al., 2021). It might be that compared to linear and tree-based models, an overparametrized deep net is more capable of learning an aligned function.

3.3 Noise-Aware Robust Learning has Disparate Impact

Overview & Experimental Setup. We now assess whether training models with noise-aware algorithmic interventions (e.g. robust loss functions (Ma et al., 2020; Ghosh et al., 2017)) results in models whose disparity metrics have reduced sensitivity to label error in the training set. We test this hypothesis on a modified Cifar-10 dataset following the setting of Hall et al. (2022). Specifically, the Cifar-10 dataset is modified to a binary classification setting along with group labels by inverting a subset of each class's examples. Given a specified parameter $\epsilon \in [0, 1/2]$, a $\frac{1}{2} - \epsilon$ of the negative class is inverted, while a $\frac{1}{2} + \epsilon$ of the positive class is inverted leading to 2ϵ fraction of one group of samples and $1 - 2\epsilon$ of the other group. In all experiments we set $\epsilon = 0.15$ for a 30 percent minority group membership. We replicate the label flipping experiment on the task with a Resnet-18 model. We test the MEIDTM (Cheng et al., 2022), DivideMix (Li et al., 2020), and a robust loss approach (Ghosh et al., 2017).

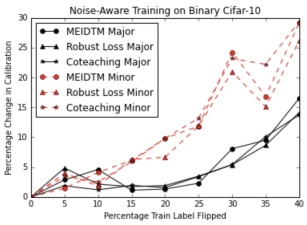

Figure 4: **Effect of Noise-aware algorithms on group calibration**.

Results. At a high level, for the majority group, we find that group calibration remains resilient to low rates of label error (below 25 percent). At high rates (>30 percent label error), we start to see increased sensitivity. However, for the minority group (30 percent of the dataset), we observe group

calibration remains sensitive to label error even at low levels. This finding suggests that noise-aware methods show are more effective for larger groups in the data.

4 Influence of Training Label on Test Disparity Metric

We now present an approach for estimating the 'influence' of perturbing a training point's label on a disparity metric of interest. We consider: 1) up-weighting a training point, and 2) perturbing the training label.

Upweighting a training point. Let $\hat{\theta}_{-z_i}$ be the ERM solution when a model is trained on all data points, $\{z_j\}_{j=1}^n$, except z_i. The influence, $\mathcal{I}_{\text{up,params}}$, of datapoint, z_i, on the model parameters is then defined: $\hat{\theta}_{-z_i} - \hat{\theta}$. This measure indicates how much the parameters change when the model is 'refit' on all training data points except z_i. Koh and Liang (2017) give a closed-form estimate of this quantity as:

$$\mathcal{I}_{\text{up,params}} \stackrel{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1}\nabla_\theta \ell(z_i, \hat{\theta}), \tag{1}$$

where H is the hessian, i.e., $H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \nabla_\theta^2 \ell(z_i, \theta)$.

The loss on a test example, $\ell(z_t, \hat{\theta})$, is a function of the model parameters, so using the chain-rule, we can estimate the influence, $\mathcal{I}_{\text{up,loss}}(z_i, z_t)$, of a training point, z_i, on $\ell(z_t, \hat{\theta})$ as:

$$\mathcal{I}_{\text{up,loss}}(z_i, z_t) \stackrel{\text{def}}{=} \left.\frac{d\ell(z_t, \hat{\theta}_{\epsilon,\, z_i})}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta \ell(z_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1}\nabla_\theta \ell(z_i, \hat{\theta}). \tag{2}$$

Perturbing a training point's label. A second notion of influence that Koh and Liang (2017) study is how perturbing a training point leads to changes in the model parameters. Specifically, given a training input, z_i, that is a tuple (x_i, y_i), how would the perturbation, $z_i \to z_{i,\delta}$, which is defined as $(x_i, y_i) \to (x_i, y_i + \delta)$, change the model's predictions? Koh and Liang (2017) give a closed-form estimate of this quantity as:

$$\mathcal{I}_{\text{pert,loss,y}}(z_j, z_t) \approx -\nabla_\theta \ell(z_t, \hat{\theta}_{z_{j,\delta}, -z_j})^\top H_{\hat{\theta}}^{-1}\nabla_y \nabla_\theta \ell(z_j, \hat{\theta}). \tag{3}$$

Adapting influence functions to group disparity metrics. We now propose modifications that allow us to compute the influence of a training point on a test group disparity metric (See Appendix D for longer discussion). Let S_t be a set of test examples. We can then denote $\mathcal{GD}(S_t, \hat{\theta})$ as the group disparity metric of interest, e.g., the estimated ECE for the set S_t given parameter setting $\hat{\theta}$.

Influence of upweighting a training point on a test group disparity metric. A group disparity metric on the test set is a function of the model parameters; consequently, we can apply the chain rule to $\mathcal{I}_{\text{up,params}}$ (from Equation 1) to estimate the influence, $\mathcal{I}_{\text{up,disparity}}$, of up-weighting a training point on the disparity metric as follows:

$$\begin{aligned}\mathcal{I}_{\text{up,disparity}}(z_i, S_t) &\stackrel{\text{def}}{=} \left.\frac{d\mathcal{GD}(S_t, \hat{\theta}_{\epsilon,\, z_i})}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top \left.\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon}\right|_{\epsilon=0}, \\ &= -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1}\nabla_\theta \ell(z_i, \hat{\theta}).\end{aligned} \tag{4}$$

We now have a closed-form expression for a training point's influence on a test group disparity metric.

Influence of perturbing a training point's label on a test group disparity metric. We now consider the influence of a training label perturbation on a group disparity metric of interest. To do this, we simply consider the group disparity metric function as the quantity of interest instead of the test loss. Consequently, the closed-form expression for the influence of a modification to the training label on disparity for a given test set is:

$$\mathcal{I}_{\text{pert,disparity,y}}(z_j, S_t) \approx -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_y \nabla_\theta \ell(z_j, \hat{\theta}). \tag{5}$$

With Equations 4cand 5, we have the key quantities of interest that allows us to rank training points, in terms of influence, on the test group disparity metric.

5 Identifying and Correcting Training Label Error

In this section, we empirically assess the modified influence expressions for calibration across these datasets for prioritizing mislabelled samples. We find a 10-40 percent improvement, compared to alternative approaches that estimate influence on test loss (Kong et al., 2021) and other uncertainty estimation based approaches. In addition, we propose an approach to automatically correct the labels identified by our proposed approach.

5.1 Identifying Label Error

Overview & Experimental Question. We are interested in surfacing training points whose change in label will induce a concomitant change in a test disparity metric like group calibration. Specifically, we ask: When the training points are ranked by influence on test calibration, are the most highly influential training points most likely to have the wrong labels? We conduct our experiments to directly measure a method's ability to answer this question.

Experimental Setup. For each dataset, we randomly flip the labels of 10 percent of the training samples. We then train our standard logistic regression classifier on this modified dataset. In this task, we have direct access to the ground-truth and knowledge of the exact samples whose labels were flipped. This allows us to directly compare the performance of our proposed methods to each of the baselines on this task. We then rank training points using a number of baseline approaches as well as the modified influence approaches. For the top 50 examples, we consider what fraction of these examples had flipped labels in the training set.

Approaches & Baselines. We consider the following methods: 1) **IF-Calib**: The closed-form approximation to the influence of a training point on the test calibration; 2) **IF-Calib-Label**: The closed-form approximation to the influence of a training point's label on the test calibration; 3) **Loss**: A baseline method which is the training loss evaluated at each data point in the training set. The intuition is that, presumably, more difficult training samples will have higher training loss; 4) **Uncertainty Estimation Based Approaches**: We also consider methods two uncertainty estimation algorithms: 1) an approach that estimate a sample's uncertainty based on a K-fold validation score, and 2) a logit-based approach (Wu and Klabjan, 2021). 5) **Two Noise-aware baselines**: which estimate a confidence/transition matrix on training samples to determine the probability of error of a sample's label for a given class (Northcutt et al., 2021a; Cheng et al., 2022).

Results: Prioritizing Samples. In Figure 5, we show the performance of the two approximations that we consider in this work as well as two baselines. We plot the fraction of inputs, out of the top ranked 50 ranked training points, whose labels were flipped in the training set. The higher this proportion, then the more effective an approach is in identifying the samples that likely have wrong labels. In practice, the goal is to surface these training samples and have a domain expert inspect them. If a larger proportion of the items to be inspected are mislabeled, then a higher proportion of training set mistakes, i.e. label error, can be fixed. Across the different datasets, we find a 10-40 percent improvement, compared to baseline approaches, in identifying critical training data points whose labels need to be reexamined.

We find the loss baseline to be ineffective for ranking in our experiments. A possible reason is that modern machine learning models can typically be trained to 'memorize' the training data; resulting in settings where a model has low loss even on outliers or mislabeled examples. In such a case, ranking by training loss for a sample is an ineffective ranking strategy. We find that the noise-aware approaches perform similarly to the IF-Norm baseline. We defer the results of the uncertainty-based baselines and the noise-aware methods to Appendix (Section F). We find that these baselines also underperform our proposed approaches. For example, the 5-fold cross validation setting underperforms the standard influence functions approach by 50 percent.

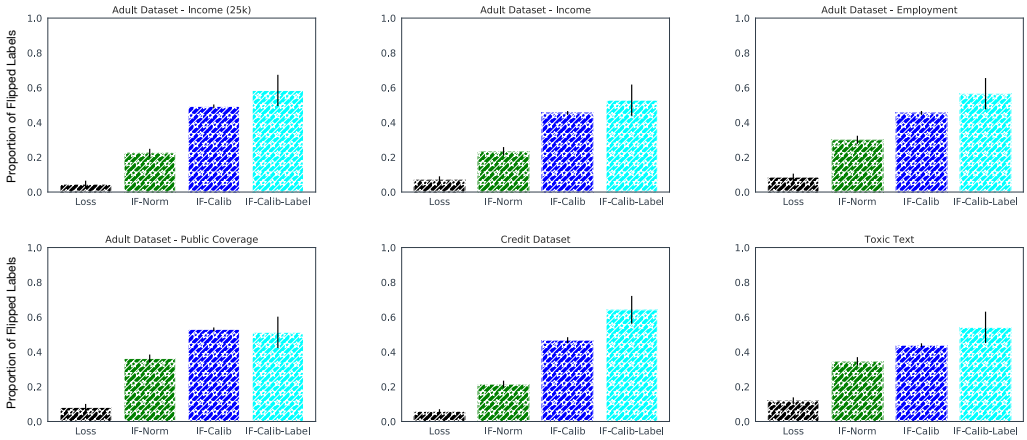

Figure 5: **Empirical Results for Training Point Ranking Across 6 datasets**. For the top 50 most influential examples, we show the proportion of samples whose labels were flipped in the training data. Confidence intervals is computed for N=5 different models.

5.2 Correcting Label Error

We take label error identification one step further to automatically relabelling inputs that have identified as critical. We restrict our focus to binary classification where the label set is $\{0, 1\}$, and the corresponding relabelling function is simply $1 - y_i$, where y_i is the predicted label.

Setup & Experiment: We consider the logistic regression model across all tasks for a setting with 20 percent training label error. We consider ECE (calibration) as the disparity function of interest. We then rank the top 20 percent of training points by label-disparity influence, our proposed approach. For these points, we apply the relabelling function, and then fine-tune the model for an additional epoch with the modified labels.

Results: First, we observe an average improvement of 8.7 percent improvement in group calibration across all groups, with larger (12.5 percent) improvement coming from the smallest group. As expected, we also observe an approximately 7 percent decrease in the average loss for the overall training set. These results point to increasing promise of automatic relabeling.

Theoretical Justification. We now present a theorem that suggests that the influence priorization and relabeling scheme described above provably leads to better calibrated models.

Theorem 1. *Given a κ-strongly convex loss function $\ell(.,.)$, with $\kappa > 0$, a training dataset, $\mathcal{D}$, where A indexes the data groups, and a model, $\hat{\theta} : x_i \to y_i$, optimized via $\ell(.,.)$ that maps inputs to labels. Let $\mathcal{Q}$ be a set of test examples all belonging to group $A = a$, where $\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta})$ is the expected calibration error of $\hat{\theta}$ on the set $\mathcal{Q}$. In addition, let $\mathcal{D}_{A=a}$ be the set of problematic training examples, belonging to group a, prioritized based on influence, i.e., $\mathcal{I}_{\mathrm{pert,calib},y^i}(x_a^i, \mathcal{Q}) > 0$. We term a model trained on a different training set ($\mathcal{D}_+$) where the problematic examples have been relabeled to be $\hat{\theta}_R$. Analogously, the expected calibration error of this new model on the set $\mathcal{Q}$ is $\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}_R)$. We have that:*

$$\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}_R) \leq \mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}).$$

We defer the proof to the Appendix. Theorem 1 suggests that when a model is trained on a relabeled dataset, following the influence prioritization scheme, the expected group calibration of the retrained model should be lower than that of a model trained on a dataset that has not been relabeled.

6 Related Work

We discuss directly related work here, and defer a longer discussion to Section A of the Appendix.

Impact of Label Error/Noise on Model Accuracy. Learning under label error falls under the category more commonly known as *learning under noise* (Frénay and Verleysen, 2013; Natarajan et al., 2013; Bootkrajang and Kabán, 2012). *Noise* in learning can come from different either input features or the labels. In this work, we focus on label error—categorization mistakes associated with the label in both the test and training data. Previous work focused primarily on the effect of label error in the training data; however, we advance this line of work to investigate the effect of label error in the test data used to conduct a fairness audit on the reliability of the audit. Model resilience to training label error has been studied for both synthetic (Arpit et al., 2017; Zhang et al., 2021; Rolnick et al., 2017) and real-world noise settings (Jiang et al., 2020). A major line of inquiry is the development of algorithmic approaches to learn accurate models given a training set with noisy labels. These approaches include model regularization (Srivastava et al., 2014; Zhang et al., 2017), bootstrap (Reed et al., 2014), knowledge distillation (Jiang et al., 2020), instance weighting (Ren et al., 2018; Jiang and Nachum, 2020), robust loss functions (Ma et al., 2020; Ghosh et al., 2017), or trusted data (Hendrycks et al., 2018), joint training (Wei et al., 2020), mixture models in semi-supervised learning (Li et al., 2020), and methods to learn a transition matrix that captures noise dependencies (Cheng et al., 2022). In contrast to this line of work, we primarily seek to identify the problematic instances that need to be relabelled, often by a human labeler, and not automatically learn a model that is robust to label error.

Impact of Label Error on Model *'Fairness'*. This work contributes to the burgeoning area that studies the impact of label error on a model's 'fairness' (termed 'group-based disparity' in this paper) metrics. Fogliato et al. (2020) studied a setting in which the labels used for model training are a noisy proxy for the true label of interest, e.g., predicting rearrest as a proxy for rearrest. Wang et al. (2021) considers an ERM problem subject to group disparity constraints with group-dependent label noise, and provides theoretical results along with a scheme to obtain classifiers that are robust to noise. Different from their setting, we consider unconstrained ERM (no fairness constraints during learning). Similarly, Konstantinov and Lampert (2021) study the effect of adversarial data corruptions on fair learning in a PAC model. Jiang and Nachum (2020) propose a re-weighting scheme that is able to correct for label noise.

Influence Functions & Their Uses. Influence functions originate from robust statistics where it is used as a tool to identify outliers Cook and Weisberg (1982); Cook (1986); Hampel (1974). Koh and Liang (2017) introduced influence functions for modern machine learning models, and used them for various model debugging tasks. Most similar to our work, Kong et al. (2021) propose RDIA, a relabelling scheme based on the influence function that is able to provably correct for label error in the training data. RDIA identifies training samples that have a high influence on the test loss for a validation set; however, we focus on identifying training samples whose change in label has a high influence on a group-disparity metric on a test/audit set. We compare to their approach in Section 5. In recent work, De-Arteaga et al. (2021) study expert consistency in data labeling and use influence functions to estimate the impact of labelers on a model's predictions. Along similar direction, Brunet et al. (2019) adapt the influence function approach to measure how removing a small part of a training corpus, in a word embedding task, affects test bias as measured by the word embedding association test Caliskan et al. (2017). Feldman and Zhang (2020) use influence functions to estimate how likely a training point is to have been memorized by a model. More generally, influence functions are gaining widespread use as a tool for debugging model predictions (Barshan et al., 2020; Han et al., 2020; Yeh et al., 2018; Pruthi et al., 2020). Different from these uses of influence functions, here we isolate the effect of a training point's label on a model's disparity metric on a audit data.

7 Conclusion

In this paper, we sought to address two key questions: *1) What is the impact of label error on a model's group disparity metric, especially for smaller groups in the data;* and *2) How can a practitioner identify training samples whose labels would also lead to a significant change in the test disparity metric of interest?* We find that disparity metrics are, indeed, sensitive to test and training time label error particularly for minority groups in the data. In addition, we present an approach for estimating the 'influence' of perturbing a training point's label on a disparity metric of interest, and find a 10-40% improvement, compared to alternative approaches, in identifying training inputs that improve a model's disparity metric. We present an approach to estimate the effect of a training input's label on a model's group disparity metric. Lastly, perform a simple automatic relabel-and-finetune scheme that produces updated models with, provably, improved group calibration error.

REFERENCES

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Chloé Bakalar, Renata Barreto, Stevie Bergman, Miranda Bogen, Bobbie Chern, Sam Corbett-Davies, Melissa Hall, Isabel Kloumann, Michelle Lam, Joaquin Quiñonero Candela, et al. Fairness on the ground: Applying algorithmic fairness approaches to production systems. *arXiv preprint arXiv:2103.06172*, 2021.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.

Jakramate Bootkrajang and Ata Kabán. Label-noise robust logistic regression and its applications. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 143–158. Springer, 2012.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *arXiv preprint arXiv:2110.01889*, 2021.

Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard Zemel. Understanding the origins of bias in word embeddings. In *International Conference on Machine Learning*, pages 803–811. PMLR, 2019.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Edwin Chen. 30% of google's emotions dataset is mislabeled, 2022. URL https://www.surgehq.ai/blog/30-percent-of-googles-reddit-emotions-dataset-is-mislabeled.

De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Masashi Sugiyama. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16630–16639, 2022.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

R Dennis Cook. Assessment of local influence. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(2):133–155, 1986.

R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

Maria De-Arteaga, Artur Dubrawski, and Alexandra Chouldechova. Leveraging expert consistency to improve algorithmic decision support. *arXiv preprint arXiv:2101.09648*, 2021.

Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221): 536–539, 2015.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*, 2020.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *arXiv preprint arXiv:2008.03703*, 2020.

Riccardo Fogliato, Alexandra Chouldechova, and Max G'Sell. Fairness evaluation in presence of biased noisy labels. In *International Conference on Artificial Intelligence and Statistics*, pages 2325–2336. PMLR, 2020.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Melissa Hall, Laurens van der Maaten, Laura Gustafson, and Aaron Adcock. A systematic study of bias amplification. *arXiv preprint arXiv:2201.11706*, 2022.

Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. *arXiv preprint arXiv:2005.06676*, 2020.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 702–712. PMLR, 2020.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, pages 4804–4815. PMLR, 2020.

Erik Jones, Shiori Sagawa, Pang Wei Koh, Ananya Kumar, and Percy Liang. Selective classification can magnify disparities across groups. *arXiv preprint arXiv:2010.14134*, 2020.

Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations*, 2021.

Nikola Konstantinov and Christoph H Lampert. Fairness-aware pac learning from corrupted data. *arXiv preprint arXiv:2102.06004*, 2021.

Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

Jingling Li, Mozhi Zhang, Keyulu Xu, John Dickerson, and Jimmy Ba. How does a neural network's architecture impact its robustness to noisy labels? *Advances in Neural Information Processing Systems*, 34:9788–9803, 2021.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

Lydia T Liu, Max Simchowitz, and Moritz Hardt. The implicit fairness criterion of unconstrained learning. In *International Conference on Machine Learning*, pages 4051–4060. PMLR, 2019.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

Edwin Murdoch. How i found nearly 300,000 errors in ms coco, 2022. URL https://medium.com/@jamie_34747/how-i-found-nearly-300-000-errors-in-ms-coco-79d382edf22b.

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

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26:1196–1204, 2013.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021b.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *arXiv preprint arXiv:1709.02012*, 2017.

Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. Interpretable data-based explanations for fairness debugging. *arXiv preprint arXiv:2112.09745*, 2021.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracing gradient descent. *arXiv preprint arXiv:2002.08484*, 2020.

Inioluwa Deborah Raji and Joy Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 429–435, 2019.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.

Kaspar Rufibach. Use of brier score to assess binary predictions. *Journal of clinical epidemiology*, 63(8):938–939, 2010.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Cuong Tran, Ferdinando Fioretto, Jung-Eun Kim, and Rakshit Naidu. Pruning has a disparate impact on model accuracy. *arXiv preprint arXiv:2205.13574*, 2022.

Jialu Wang, Yang Liu, and Caleb Levy. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 526–536, 2021.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

Huiyu Wu and Diego Klabjan. Logit-based uncertainty measure in classification. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 948–956. IEEE, 2021.

Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, pages 11492–11501. PMLR, 2021.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Appendix

Table of Contents

A Additional Related Work **14**

B Additional Discussion of Influence Functions **15**

C Appendix: Datasets, Models, & Experimental Details **18**

C.1 Datasets 18

C.2 Models 19

C.3 Additional Discussion on ECE and Brier Score 19

D Influence function Implementation **21**

E Additional Empirical Results **21**

F Noisy Example Identification: BenchMarking Uncertainty Estimation & Noise-Aware Schemes **24**

G BenchMarking Noise Aware Approaches **25**

H Theoretical Insights for Automatic Relabeling **26**

I Longer Conclusion **27**

A Additional Related Work

We discuss here additional literature.

Low Performance Subgroup Identification We use influence functions to identify training points that have the most effect on the model's disparity metric; however, there are other approaches for surfacing training points that need to be prioritized. For example, Kim et al. (2019) propose an algorithm to identify groups in the data where a model has high test errors, and to 'boost' the model performance for these groups. Similarly, Creager et al. (2021) propose a two-stage scheme to identify critical subgroups in the data when demographic labels are not known ahead of time. Our approach differs from the aforementioned since we focus principally on the effect to the model's disparity metric instead of the test loss.

Label Flipping. We use labeling flipping as a primary tool in our empirical tests to measure the sensitivity of a model's disparity metrics to label error. Label flipping has also previously been used for alternative purposes Arpit et al. (2017); Zhang et al. (2021). In now seminal work, Zhang et al. (2021) used label flipping and shuffling to show that deep neural networks easily memorize data with random labels. More generally, label flipping can also constitute an 'adversarial attack' against an ML model, for which there are increasingly new methods to help defend against such attacks Rosenfeld et al. (2020).

Explainability Increasingly, 'explanations' derived from a trained model can point to the reason why an input has been misclassified. Ultimately, one holy grail use-case for explanations has been help identify and suggest potential fixes for model performance disparities. Towards this end, Pradhan et al. (2021) propose an approach that intervenes on the training data and measures the changes to downstream performance on the basis of these changes.

B Additional Discussion of Influence Functions

For completeness, we recap a non-rigorous derivation of the influence of a training point on the parameters of a model following Koh and Liang (2017). We first recall the notation and setup from the paper.

Overview of Notation We will consider prediction problems, i.e, settings where the task to to learn a mapping, h, from an input space $\mathcal{X} \in \mathbb{R}^d$ to an output space $\mathcal{Y} \in \mathbb{R}^k$. We follow the notation of Koh and Liang (2017) in this work, so we consider the training points to be: $z_1, \ldots, z_n$, where each z_i is a tuple $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Given a function family, Θ, the learning task to learn a particular parameter setting $\theta \in \Theta$. Throughout this work, we will only consider learning via empirical risk minimization (ERM), which corresponds to: $\hat{\theta} := \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_i^n \ell(z_i, \theta)$. Similar to Koh and Liang (2017), we will assume that the ERM solution is twice-differentiable and strictly convex in the parameters.

Upweighting a training point Let $\hat{\theta}_{-z_i}$ be the ERM solution when a model is trained on all data points except z_i. The influence, $\mathcal{I}_{\text{up,params}}$, of datapoint, z_i, is then defined as the change: $\hat{\theta}_{-z_i} - \hat{\theta}$.

We can define the empirical risk as:

$$R(\theta) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, \theta).$$

Since we assumed that the ERM solution is twice-differentiable and strictly convex in the parameters, we can specify the hessian as:

$$H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^2 \ell(z_i, \theta) = \nabla^2 R(\theta)$$

To start, let:

$$\hat{\theta}_{\epsilon,\, z_i} = \arg\min_{\theta \in \Theta} R(\theta) + \epsilon \ell(z_i, \theta).$$

Then we can define the parameter change to be: $\Delta_\epsilon = \hat{\theta}_{\epsilon,\, z_i} - \hat{\theta}$. Since $\hat{\theta}$ does not depend on ϵ, we have that:

$$\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon} = \frac{d\Delta_\epsilon}{d\epsilon}.$$

We know that $\hat{\theta}_{\epsilon,\, z_i}$ is a solution, i.e., minimizer, so we will form the first-order optimality conditions and form a Taylor expansion of that expression.

To do this we have that:

$$\nabla_\theta (R(\theta) + \epsilon \ell(z_i, \theta)) = 0,$$
$$\nabla R(\theta) + \epsilon \nabla \ell(z_i, \theta) = 0.$$

The Taylor expansion of the above expression is then:

$$[\nabla (R(\hat{\theta}) + \epsilon \nabla \ell(z_i, \theta)] + [\nabla^2 (R(\hat{\theta}) + \epsilon \nabla^2 \ell(z_i, \theta) \Delta_\epsilon] \approx 0.$$

Above, we only keep the first two terms of the Taylor Expansion.

We will now solve for Δ_ϵ in the above equation and then differentiate by ϵ to obtain the final expression.

Solving for Δ_ϵ, results in:

$$\Delta_\epsilon \approx -[\nabla^2 (R(\hat{\theta}) + \epsilon \nabla^2 \ell(z_i, \theta)]^{-1} [\nabla (R(\hat{\theta}) + \epsilon \nabla \ell(z_i, \theta)].$$

If we look at the expression for Δ_ϵ, we find that $\nabla(R(\hat{\theta})$ is zero, since we know that $\hat{\theta}$ minimizes the empirical risk R. In looking at the term $[\nabla^2(R(\hat{\theta}) + \epsilon\nabla^2\ell(z_i, \theta)]$, we find that it is equivalent to $[H_{\hat{\theta}} + \epsilon\nabla^2\ell(z_i, \theta)]$. We can further make the assumption that the contribution of the $\epsilon\nabla^2\ell(z_i, \theta)$ term is small, so we have $[H_{\hat{\theta}} + \epsilon\nabla^2\ell(z_i, \theta)] \approx H_{\hat{\theta}}$.

With the above assumptions and substitutions, we arrive at the new expression for Δ, which is now:

$$\Delta_\epsilon \approx -H_{\hat{\theta}}^{-1}\nabla\ell(z_i, \theta)\epsilon.$$

Let's differentiate the above expression by ϵ, and we have that:

$$\frac{d\Delta_\epsilon}{d\epsilon} = -H_{\hat{\theta}}^{-1}\nabla\ell(z_i, \theta).$$

Now recall that:

$$\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon} = \frac{d\Delta_\epsilon}{d\epsilon},$$

which means:

$$\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon} = \frac{d\Delta_\epsilon}{d\epsilon} = -H_{\hat{\theta}}^{-1}\nabla\ell(z_i, \theta).$$

As discussed in the main draft, $\mathcal{I}_{\text{up,params}}$ is defined to be the influence of estimate for training point z_i, so :

$$\mathcal{I}_{\text{up,params}} \stackrel{\text{def}}{=} \frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon} = -H_{\hat{\theta}}^{-1}\nabla\ell(z_i, \theta).$$

As we have seen, we obtain the influence of a training point on a the loss of a test simply by applying the chain rule to the loss-derivative quantity of interest. We will now extend this notion of influence.

Upweighting a training point Let $\hat{\theta}_{-z_i}$ be the ERM solution when a model is trained on all data points except z_i. The influence, $\mathcal{I}_{\text{up,params}}$, of datapoint, z_i, is then defined as the change: $\hat{\theta}_{-z_i} - \hat{\theta}$. This measure indicates how much the parameters change when the model is 'refit' on all training data points except z_i. One approach to estimate this quantity is to train two copies of the model: one with all data points, and another on all data points except z_i and then compare these two parameter settings. Such approach is time and compute prohibitive especially if one is interested in computing the influence of all training points—computing the influence of n training points will require refiting the model n times. For model classes that require significant compute to estimate, such a procedure is infeasible. To circumvent the retraining challenge, Koh and Liang (2017) give a closed-form approximation to the influence quantity as:

$$\mathcal{I}_{\text{up,params}} \stackrel{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_i, \hat{\theta}), \tag{6}$$

where H is the hessian of the loss and defined as: $H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_n^{i=1}\nabla_\theta^2\ell(z_i, \theta)$. Equation 1 gives a closed-form approximation of the influence of a training point z_i on the model parameters. Recall that this quantity tells us how the model parameters change when the training point is upweighted by a small amount (say ϵ). Upweighting a training point by $-\frac{1}{n}$ is equivalent to removing this training point from the dataset.

Given a closed-form approximation to $\mathcal{I}_{\text{up,params}}$, we can estimate the influence of a training point on functions of the parameters. For example, Koh and Liang (2017) show, using a chain-rule argument, that the influence, $\mathcal{I}_{\text{up,loss}}(z_i, z_t)$, of a training point, z_i, on the test loss for a test example has the following closed-form expression:

$$
\begin{aligned}
\mathcal{I}_{\text{up,loss}}(z_i, z_t) &\stackrel{\text{def}}{=} \left.\frac{d\ell(z_t, \hat{\theta}_{\epsilon,\, z_i})}{d\epsilon}\right|_{\epsilon=0} \\
&= -\nabla_\theta \ell(z_t, \hat{\theta})^\top \left.\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon}\right|_{\epsilon=0}, \\
&= -\nabla_\theta \ell(z_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_i, \hat{\theta}). \qquad (7)
\end{aligned}
$$

As we have seen, we obtain the influence of a training point on a the loss of a test simply by applying the chain rule to the loss-derivative quantity of interest. We will now extend this notion of influence.

Perturbing a training point A second notion of influence that Koh and Liang (2017) study is how perturbing a training point leads to changes in the model parameters. Specifically, given a training input, z_i, that is a tuple (x_i, y_i), then how would the perturbation, $z_i \to z_{i,\delta}$, which is defined as $(x_i, y_i) \to (x_i + \delta, y_i)$, change the model's predictions? The key issue here is how an infinitesimal change in the input example changes the model parameters and predictions. Let $\hat{\theta}_{\epsilon, z_{i,\delta}, -z_j}$ be the ERM solution to the following minimization problem:

$$
\arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, \theta) + \epsilon \ell(z_{j,\delta}, \theta) - \epsilon \ell(z_j, \theta).
$$

As shown, $\hat{\theta}_{\epsilon, z_{i,\delta}, -z_j}$ is the ERM solution obtained when an infinitesimal mass, ϵ, is shifted from z_j i.e. (x_j, y_j) to $z_{j,\delta}$, i.e. $(x_j + \delta, y_j)$. Similarly, Koh and Liang (2017) show that the closed-form approximation for such training point perturbation on the parameters is:

$$
\begin{aligned}
\mathcal{I}_{\text{up,params}}(z_{j,\delta}) - \mathcal{I}_{\text{up,params}}(z_j) &= \left.\frac{d\hat{\theta}_{\epsilon, z_{i,\delta}, -z_j}}{d\epsilon}\right|_{\epsilon=0}, \\
&\approx -H_{\hat{\theta}}^{-1} [\nabla_x \nabla_\theta \ell(z_j, \hat{\theta})] \delta. \qquad (8)
\end{aligned}
$$

With a closed-form expression for the influence of perturbing a training point, we can also obtain similar forms for functions of the parameters. Consequently, to obtain the influence of perturbing a training point on the model's prediction (i.e. a model with parameters $\hat{\theta}$), we differentiate with respect to δ, and apply the chain rule to obtain:

$$
\begin{aligned}
\mathcal{I}_{\text{pert,loss}}(z_j, z_t) &\stackrel{\text{def}}{=} \left.\nabla_\delta \ell(z_t, \hat{\theta}_{z_{j,\delta}, -z_j})\right|_{\delta=0} \\
&\approx -\nabla_\theta \ell(z_t, \hat{\theta}_{z_{j,\delta}, -z_j})^\top H_{\hat{\theta}}^{-1} \nabla_x \nabla_\theta \ell(z_j, \hat{\theta}). \qquad (9)
\end{aligned}
$$

We now have closed-form expressions for estimating the influence of a training point and a modification to this training point on the loss of a new test example respectively. Shortly, we will make simple modifications to these equations to obtain the influence of the training point on the calibration for a test point.

Perturbing a training point's label We are also interested in how a modification to the *training label* influences the parameters and the test loss. In this case, we are interested in how the perturbation, $(x_i, y_i) \to (x_i, y_i + \delta)$ changes the model's predictions. An analogous calculation to that of Equation 8 results in:

$$
\mathcal{I}_{\text{up,params,y}}(z_{j,\delta}) - \mathcal{I}_{\text{up,params,y}}(z_j) \approx -H_{\hat{\theta}}^{-1} [\nabla_y \nabla_\theta \ell(z_j, \hat{\theta})] \delta. \qquad (10)
$$

In Equation 10, the inner gradient is taken with respect to the label, y, as opposed to the the input, x, as was done in Equation 8. Consequently, the closed-form expression for the influence of a modification to the training label on the loss of a new test example is:

$$\mathcal{I}_{\text{pert,loss,y}}(z_j, z_t) \approx -\nabla_\theta \ell(z_t, \hat{\theta}_{z_{j,\delta},-z_j})^\top H_{\hat{\theta}}^{-1} \nabla_y \nabla_\theta \ell(z_j, \hat{\theta}). \tag{11}$$

We now have all the closed-form expressions we need for estimating influence with respect to a disparity metric of interest.

Adapting influence functions to group disparity metrics. We now propose modifications that allow us to compute the influence of a training point on a test group disparity metric. Let S_t be a set of test examples. We can then denote $\mathcal{GD}(S_t, \hat{\theta})$ as the group disparity metric of interest, e.g., the estimated ECE for the set S_t given parameter setting $\hat{\theta}$.

Influence of upweighting a training point on a test group disparity metric. A group disparity metric on the test set is a function of the model parameters; consequently, we can apply the chain rule to $\mathcal{I}_{\text{up,params}}$ (from Equation 1) to estimate the influence, $\mathcal{I}_{\text{up,disparity}}$, of up-weighting a training point on the disparity metric as follows:

$$\begin{aligned}\mathcal{I}_{\text{up,disparity}}(z_i, S_t) &\stackrel{\text{def}}{=} \left.\frac{d\mathcal{GD}(S_t, \hat{\theta}_{\epsilon,\, z_i})}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top \left.\frac{d\hat{\theta}_{\epsilon,\, z_i}}{d\epsilon}\right|_{\epsilon=0}, \\ &= -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_i, \hat{\theta}). \end{aligned} \tag{12}$$

We now have a closed-form expression for a training point's influence on a test group disparity metric.

Influence of perturbing a training point's label on a test group disparity metric. We now consider the influence of a training label perturbation on a group disparity metric of interest. To do this, we simply consider the group disparity metric function as the quantity of interest instead of the test loss. Consequently, the closed-form expression for the influence of a modification to the training label on disparity for a given test set is:

$$\mathcal{I}_{\text{pert,disparity,y}}(z_j, S_t) \approx -\nabla_\theta \mathcal{GD}(S_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_y \nabla_\theta \ell(z_j, \hat{\theta}). \tag{13}$$

With Equations 12 and 13, we have the key quantities of interest that allows us to rank training points, in terms of influence, on the test group disparity metric. We can then use these quantities to prioritize samples that should be more carefully inspected.

C Appendix: Datasets, Models, & Experimental Details

We provide additional details about the datasets, models, and general experimental details. We follow the discussion in the background section of the paper and provide additional details where necessary. We plan to release opensource code that can be used to replicate all of or analyses.

C.1 Datasets

We start with a discussion of the datasets used in this work.

Text Toxicity Classification we use a publicly available dataset from JIGSAW called the *unintended bias in toxicity classification* dataset. The TC dataset contains a subset of comments from the 'Civil Comments' platform that have been annotated by human raters for level of toxic severity. For example, a comment can be tagged as 'benign', 'obscene', 'threat', 'insult', or 'sexually explicit' among several categories. A given text is then indicated as 'toxic' or 'not toxic' on the basis of these tags. The task here is a binary classification one, which is to categorize an input text as either toxic or not.

To obtain the toxicity labels, each comment was shown to up to 10 annotators. Annotators were asked to:"Rate the toxicity of this comment": Very Toxic, Toxic, Hard to Say, and Not Toxic. These ratings were then aggregated with the target value representing the fraction of annotations that annotations fell within the former two categories.

To collect the identity labels, annotators were asked to indicate all identities that were mentioned in the comment. An example question that was asked as part of this annotation effort was:"What

genders are mentioned in the comment?": Male, Female, Transgender, Other Gender, and No gender mentioned. We consider the Gender variable to be the sensitive attribute of interest.

For the dataset, we taken of the raw sentence text and convert them into embedding vectors using the XLM-R models obtaining a vector, for each sentence, that is 768 dimensional. We then further reduce the input dimension of this vector from 768 to 50 via a random projection. We make this reduction to make computing the hessian of the loss function of the logistic regression model trained on this data easy to compute. The 50-dimensional embedding is then used as input for our analyses.

Adult Census Dataset we use a series of tabular datasets more broadly called the 'Adult Dataset'. Specifically, we consider a recently revamped version of the dataset introduced by Ding et al. (2021), which is derived from the broader US Census. We consider the following tasks among the compilation of datasets available in this database:

1. ACSIncome Prediction, where the task is to predict whether an individual has an income above 50000 USD;
2. ACSIncome Prediction (25k) where the task is to predict whether an individual has an income above 25000 USD;
3. ACSEmployment Prediction, where the task is to predict whether the adult individual is employed; and
4. ACSPublic Coverage: where the task is to predict whether a low-income individual has coverage from public health insurance.

The adult dataset comes annotated with race categories, which we take as the key demographic variable in our analyses. We restrict our analyses to data from year 2018 for the state of California across all datasets.

Credit Dataset : a dataset of financial transactions that consists of aggregate demographic and credit information for customers of a large commercial bank. For each customer, information includes their gender, marital status, educational level, employment status, income, age, and asset. Using this information, the bank estimates probability of default on loan for each customer (a scalar between 0 and 1). We consider the prediction of the default probability as our primary focus. The dataset comes with gender as a sensitive attribute for two categories: Male and Female. While this dataset is not publicly available—it is hosted by the first author's institution, it has been used as part of previous work studying financial well-being classification.

C.2 Models

.

Models For the logistic regression model. All but one of the datasets we consider are tabular, and mostly low-dimensional. We implement the logistic regression model in PyTorch. We tuned the batch size using the validation set in the range [16, 32, 64, 128] across all datasets, but did not observe substantial across differences, so we set default batch size to be: 128. We use the SGD plain optimizer with a default learning rate of 0.001, we train the all models for 20 epochs. For the Resnet-18 and GBT models, we keep the default parameter settings.

C.3 Additional Discussion on ECE and Brier Score

The principal metric that we consider in this work is group calibration. We also consider other group-based metrics such as true-positive and false negative rates. Let $\hat{y}_i = h(x_i)$ be the output of a trained model of interest for input x_i. The output can either be an output probability for a binary prediction or a class prediction in a multi-class setting. We denote the ground-truth confidence or probability of correctness as $\hat{p}_i$. We say h is calibrated if $\hat{p}_i$ represents a true probability. As an example, if given 100 predictions with confidence 0.95, then if h is calibrated, 95 of these predictions should be correct given ground-truth labels.

Several recent works have also studied model calibration and found that modern neural network models are uncalibrated. Guo et al. (2017) found that a simple temperature based Platt-scaling post

processing was the most effective technique. While the literature abound with disparity metrics, several of which provide conflicting and often counter-intuitive insights simultaneously, group calibration has emerged as a useful metric in practice Pleiss et al. (2017).

There are a variety of metrics for quantifying a model's level of calibration. In this work, we measure calibration with using the metric: Expected Calibration Error (ECE) Naeini et al. (2015). A calibration metric summarizes the difference between the empirical distribution of a model's prediction and the ground-truth probabilities for a perfectly calibrated classifier. Here, perfect calibration is defined as:

$$\mathbb{P}\left(\hat{y} = y | \hat{p} = p\right), \forall p \in [0, 1].$$

In practice, the quantity above is impossible to compute, so previous literature made empirical approximations. We follow binning the approach by Guo et al. (2017). To estimate the accuracy empirically, we group predictions into M interval bins (each of size 1/M) and calculate the accuracy of each bin. Across all experiments, we take $M = 10$—recent work Küppers et al. (2020) found this default setting effective across a range of datasets. We discuss this choice in more detail in the appendix. Let B_m be the set of indices of samples whose prediction confidence falls into the interval $I_m = [\frac{m-1}{M}, \frac{m}{M}]$. Here the accuracy of B_m is then:

$$\mathrm{acc(B_m)} = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y_i} = y_i),$$

where $\hat{y_i}$, and y_i are the predicted and true class labels for data sample i.

The ECE is the absolute difference in expectation between confidence and accuracy. We can define the average confidence within a bin B_m as:

$$\mathrm{conf(B_m)} = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p_i},$$

where $\hat{p_i}$ is the confidence for sample i. Consequently, a perfectly calibrated model will have accuracy and confidence equal for all bins.

A notion of miscalibration is then which is the difference in expectation between confidence and accuracy.

The ECE approximates the above notion of miscalibration, and is defined as:

$$\sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{conf(B_m)} - \mathrm{acc(B_m)}|.$$

We use ECE as the primary metric of miscalibration in this work. As we will see, we will compare ECE metrics across various data groups to help identify input groups where a model is miscalibrated.

Throughout all of our experiments, we use a logistic regression model. All but one of the datasets we consider are tabular, and mostly low-dimensional, so we did not observe substantial accuracy gains for more sophisticated models. In addition, as we will see, our proposed influence function approach requires Hessian-vector products, which are more easily tractable for lower-dimensional models; in the case of the logistic regression model, the hessian of the loss has a simple closed-form expression that is easy to work with and compute. Lastly, a point we will return to in the discussion section is that influence function approaches have been shown to be challenging to work with for modern deep learning models.

For the TC dataset, we obtain embedding vectors, 768 dimensional, for all samples using the XLM-R, a state of the art multi-lingual model Conneau et al. (2019). We then further reduce the input dimension of this vector from 768 to 50 via a random projection. We make this reduction to make computing the hessian of the loss function of the logistic regression model trained on this data easy to compute. The 50-dimensional embedding is then used as input for our analyses. Across all datasets, we take 70 percent of each dataset for training and model validation, while we keep 30 percent as a holdout set for computing the disparity metrics. For the 70 percent portion, we reserve 20 percent as validation set. We refer to the Appendix for additional details.

Approach	Adult	Credit	Toxic
IF-Disparity	0.47	0.51	0.45
IF-Disparity-label	0.71	0.69	0.82
IF-Norm	0.24	0.3	0.27
Loss	0.11	0.21	0.15

Table 2: Additional influence function metrics across datasets.

Compute. All of the experiments in the paper we performed on a cluster with a single Tesla T4 GPU.

D INFLUENCE FUNCTION IMPLEMENTATION

We perform our influence function experiments only on the logistic regression model class. We make this choice for scalability reasons—computing inverse hessian-vector products for the logistic regression model is straightforward and does not incur approximation challenges.

We present additional results here:

E ADDITIONAL EMPIRICAL RESULTS

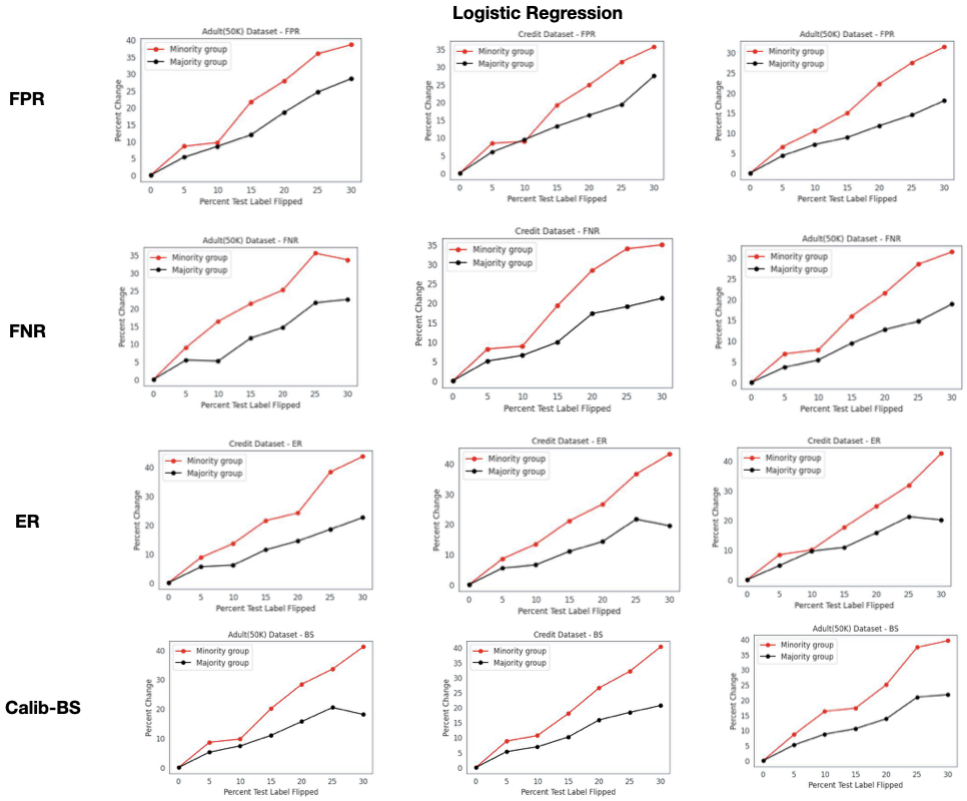

Figure 6: **Label Flipping Results for Logistic Regression Model Class**. For each dataset, we plot the percent change in disparity metric versus the corresponding percentage change in label error for the test & training set. Here, we plot the minority (smallest) group as well as the majority (largest) groups by size.

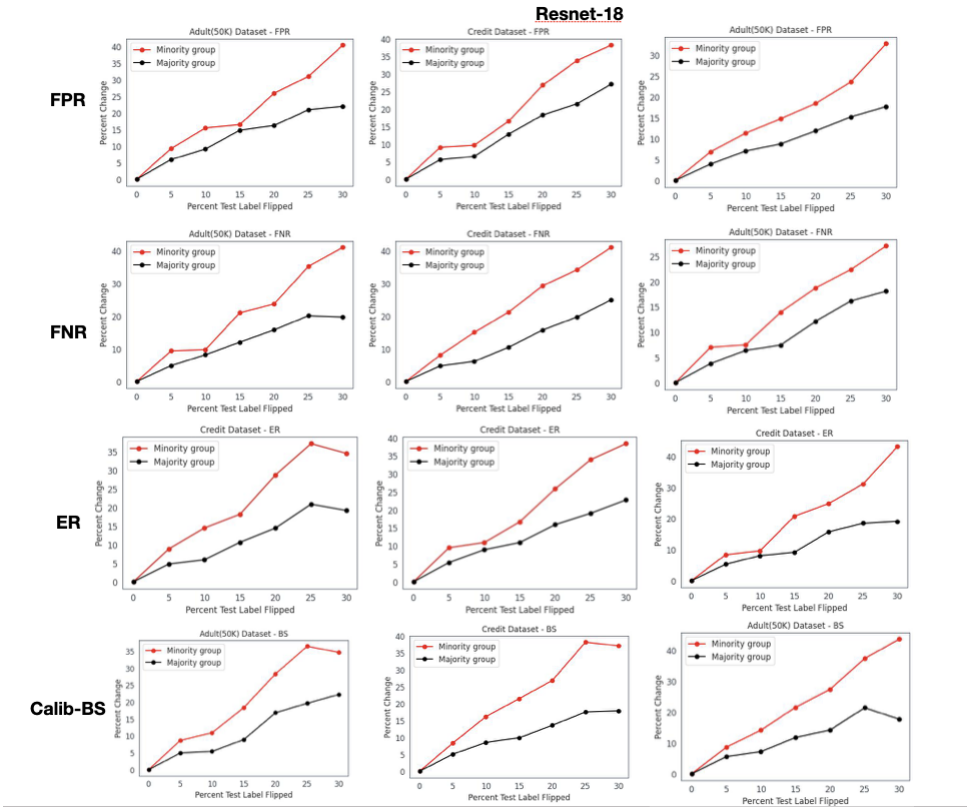

Figure 7: **Label Flipping Results for Resnet-18 Model Class**. For each dataset, we plot the percent change in disparity metric versus the corresponding percentage change in label error for the test & training set. Here, we plot the minority (smallest) group as well as the majority (largest) groups by size.

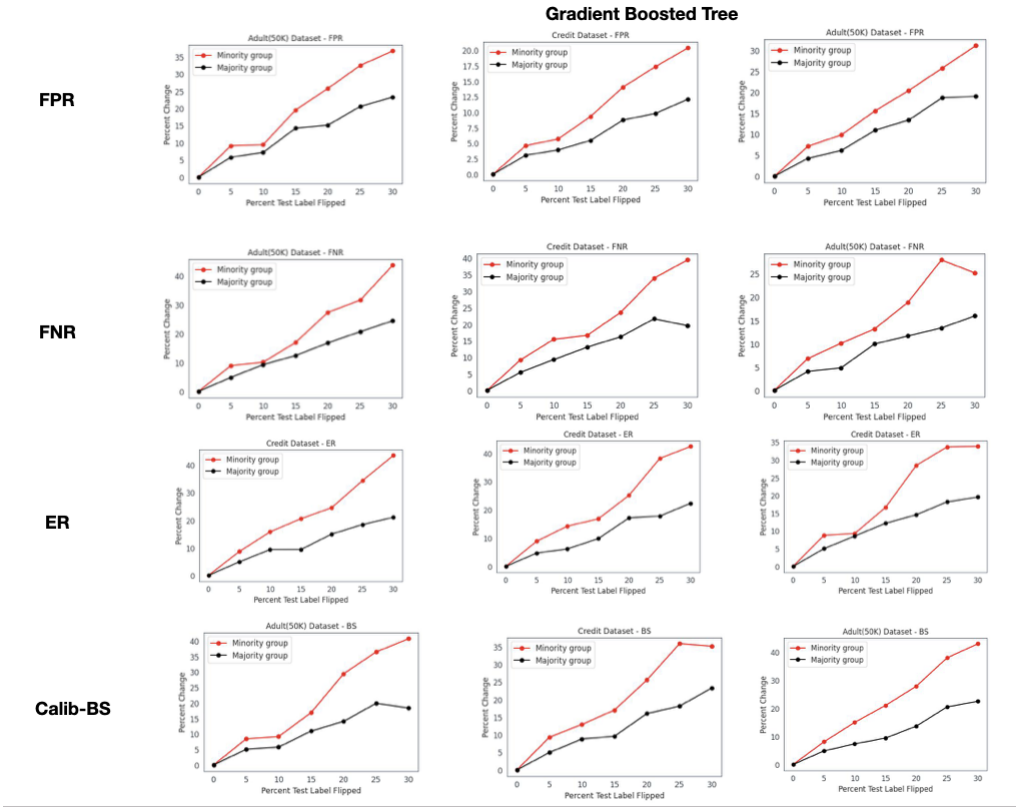

Figure 8: **Label Flipping Results for Gradient Boosted Regression Model Class**. For each dataset, we plot the percent change in disparity metric versus the corresponding percentage change in label error for the test & training set. Here, we plot the minority (smallest) group as well as the majority (largest) groups by size.

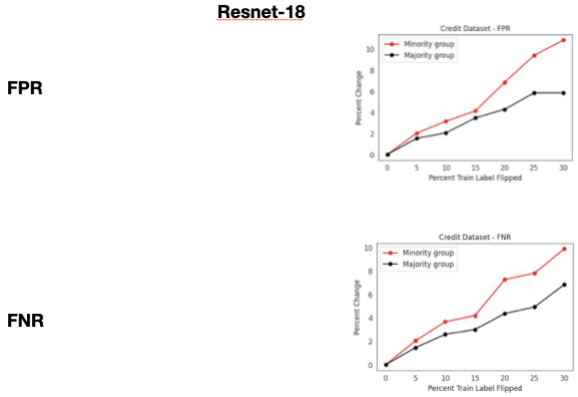

Figure 9: **Training Label Flipping Results for Resnet-18 Model Class**.

F NOISY EXAMPLE IDENTIFICATION: BENCHMARKING UNCERTAINTY ESTIMATION & NOISE-AWARE SCHEMES

In this section, we benchmark two kinds of uncertainty estimation algorithms: 1) that estimates the uncertainty in a sample's label based on a k-fold (5, and 1) cross validation score and 2) a logit based uncertainty estimation algorithm due to Wu and Klabjan (2021). We test these three variants under the sample setup as described in the main draft in Figure 4.

Method	Adult (Income 25k)	Adult (Income)	Adult (Employment)	Adult PC	Credit Dataset
Logit	0.15 (0.05)	0.17 (0.09)	0.24 (0.09)	0.27 (0.01)	0.21(0.04)
CV ($k = 1$)	0.21 (0.01)	0.23 (0.05)	0.25 (0.07)	0.31 (0.11)	0.19 (0.1)
CV ($k = 5$)	0.09 (0.06)	0.11 (0.09)	0.17 (0.1)	0.23 (0.14)	0.085 (0.02)
Confident Learning	0.17 (0.1)	0.18 (0.12)	0.2 (0.039)	0.25 (0.11)	0.15 (0.05)
MEIDTM	0.165 (0.03)	0.17 (0.025)	0.19 (0.11)	0.22 (0.29)	0.16(0.03)

G BENCHMARKING NOISE AWARE APPROACHES

Overview & Experimental Setup. We now assess whether training models with noise-aware algorithmic interventions (e.g. robust loss functions (Ma et al., 2020; Ghosh et al., 2017)) results in models whose disparity metrics have reduced sensitivity to label error in the training set. We test this hypothesis on a modified Cifar-10 dataset following the setting of Hall et al. (2022). Specifically, the Cifar-10 dataset is modified to a binary classification setting along with group labels by inverting a subset of each class's examples. Given a specified parameter $\epsilon \in [0, 1/2]$, a $\frac{1}{2} - \epsilon$ of the negative class is inverted, while a $\frac{1}{2} + \epsilon$ of the positive class is inverted leading to 2ϵ fraction of one group of samples and $1 - 2\epsilon$ of the other group. In all experiments we set $\epsilon = 0.15$ for a 30 percent minority group membership. We replicate the label flipping experiment on the task with a Resnet-18 model. We test the MEIDTM (Cheng et al., 2022), DivideMix (Li et al., 2020), and a robust loss approach (Ghosh et al., 2017). We train the Resnet-18 with default parameters from Hall et al. (2022).

Results. At a high level, for the majority group, we find that model accuracy and downstream disparity metrics remain resilient to low rates of label error (below 25 percent). At high rates (>30 percent label error), we start to see declines in these performance metrics. However, for the minority group (30 percent of the dataset), we observe that the disparity metrics show consistent decline as label error is injected in these groups. This finding suggests that noise-aware methods show disparate performance in their ability to confer robustness to label error depending on data group sizes. A similar observation has also been made for other algorithmic interventions like Pruning (Tran et al., 2022; Hooker et al., 2019), differential privacy (Bagdasaryan et al., 2019), selective Classification (Jones et al., 2020) and adversarial training (Xu et al., 2021).

H THEORETICAL INSIGHTS FOR AUTOMATIC RELABELING

In this Section, we will prove Theorem 1. To do this, we will appeal to two Lemmas from recent work. First, we restate the theorem here:

Theorem 2. *Given a κ-strongly convex loss function $\ell(.,.)$, with $\kappa > 0$, a training dataset, $\mathcal{D}$, where A indexes the data groups, and a model, $\hat{\theta} : x_i \rightarrow y_i$, optimized via $\ell(.,.)$ that maps inputs to labels. Let $\mathcal{Q}$ be a set of test examples all belonging to group $A = a$, where $\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta})$ is the expected calibration error of $\hat{\theta}$ on the set $\mathcal{Q}$. In addition, let $\mathcal{D}_{A=a}$ be the set of problematic training examples, belonging to group a, prioritized based on influence, i.e., $\mathcal{I}_{\mathrm{pert,calib},y^i}(x_a^i, \mathcal{Q}) > 0$. We term a model trained on a different training set ($\mathcal{D}_+$) where the problematic examples have been relabeled to be $\hat{\theta}_R$. Analogously, the expected calibration error of this new model on the set $\mathcal{Q}$ is $\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}_R)$. We have that:*

$$\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}_R) \leq \mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}).$$

First, we will state Lemma that bounds the calibration of a classifier in terms of its excess risk. Liu et al. (2019) study unconstrained ERM and the associated calibration of the resulting model. They give a bound on the calibration of a model obtained via ERM in terms of its excess risk.

Lemma 3 (Theorem 2.3 from Liu et al. (2019))**.** *Given a κ-strongly convex loss function $\ell(.,.)$, with $\kappa > 0$, let ℓ^* be the population risk of the Bayes classifier, then $\mathrm{ECal}(\hat{\theta})$, the expected calibration error of a classifier, $\hat{\theta}$, can be bounded as:*

$$\mathrm{ECal}(\hat{\theta}) \leq 2\sqrt{\frac{\ell(\hat{\theta}) - \ell^*}{\kappa}}.$$

We refer readers to Liu et al. (2019) for the proof. In the Lemma, we are able to bound the group expected calibration error for an ERM solution relative to that of the best classifier in the model class. The strategy we will take to prove Theorem 1 will be to relate the bound from the Lemma 3 for two different classifiers. We will compare the bound for a classifier trained on a dataset without relabelled training samples to one trained on a data that has been relabeled. In particular, the relabeling will be according to the influence of a change in a training point's label on the expected loss on a particular test set.

We now define a few quantities. We let: $\mathcal{Q}$ represent the collection of examples in the test set. For now, we will restrict $\mathcal{Q}$ to consist of examples for a single group $a \in A$. Here $A \in [1, \ldots, |A|]$ indexes the groups in the training set, $\mathcal{D}$. We will let a model trained on the original training set (no relabelling) to be: $\hat{\theta}$.

We recall that $\mathcal{I}_{\mathrm{pert,calib},y^i}(x_a^i, \mathcal{Q})$, defined from the main text, estimates the influence of changing the label of the training sample x_a^i by an infinitesimal amount. Consequently, based on the relabelling scheme that we have defined in Section 5.2, we can define a subset of the training set: $\mathcal{D}_{A=a}$, which would be examples for which $\mathcal{I}_{\mathrm{pert,calib},y^i}(x_a^i, \mathcal{Q}) > 0$. These are the problematic examples that need to be relabeled. The set of examples that corresponds to the relabeled version is $\mathcal{D}_+$. Since we are primarily focused on a binary classification task, the relabeled version is simply a bit flip of the original labels.

Lemma 4 (Theorem 1 from Kong et al. (2021))**.** *Given two binary classifiers: $\hat{\theta}$, with loss $\ell(\hat{\theta}, \mathcal{Q})$, and $\hat{\theta}_R$ with loss $\ell(\hat{\theta}_R, \mathcal{Q})$ on the test set $\mathcal{Q}$ respectively, then:*

$$\ell(\hat{\theta}_R, \mathcal{Q}) \leq \ell(\hat{\theta}, \mathcal{Q}) \leq 0.$$

Lemma 4 indicates that the loss of the model trained on the related dataset is lower or equal to the loss of the model trained on the dataset where a problematic data points, according to influence, have been relabeled.

To conclude the prove of the Theorem, we will combine insights from these two previously stated lemmas. First, we have that the bound on the expected calibration error of $\hat{\theta}$ according to Lemma 3 is:

$$\mathrm{ECal}(\hat{\theta}) \leq 2\sqrt{\frac{\ell(\hat{\theta}) - \ell^*}{\kappa}}.$$

Similarly, the bound on the expected calibration error of $\hat{\theta}_R$ according to Lemma 3 is:

$$\mathrm{ECal}(\hat{\theta}_R) \leq 2\sqrt{\frac{\ell(\hat{\theta}_R) - \ell^*}{\kappa}}.$$

From Lemma 4, we know that $\ell(\hat{\theta}_R, \mathcal{Q}) \leq \ell(\hat{\theta}, \mathcal{Q}) \leq 0$, consequently, we can conclude that: $\mathrm{ECal}_{\mathcal{Q}}(\hat{\theta}_R) \leq \mathrm{ECal}_{\mathcal{Q}}(\hat{\theta})$.

I Longer Conclusion

In this work, we study the effect of label error, both in the training and test set, on a model's disparity metric. Fairness audits are typically used to help surface disparate performance across groups in the data. However, if either the test data used to perform a fairness assessment or the training data used to fit the model contains label error, then it is unclear whether the conclusions of such a fairness assessment are reliable. Given the widespread presence of label error in the ML labeling pipeline, understanding the downstream effects of label error on a model's disparity metric is important. In this paper, we sought to address two key questions: *1) What is the impact of label error on a model's group disparity metric, especially for smaller groups in the data;* and *2) How can a practitioner identify training samples whose labels would also lead to a significant change in the test disparity metric of interest?*

To address the first question, we conducted empirical sensitivity tests. First, we flip the labels in the test set, to simulate label error, then measure the corresponding change to the disparity metric. Second, we fix a test set, but instead iteratively simulate varying levels of label error in the training set. We then measure the change in disparity metric, e.g., group calibration error for models trained on these 'contaminated' datasets compared to a model derived from data without label error.

We find that disparity metrics are, indeed, sensitive to test and training time label error. In addition, for the same level of label error, the percentage change in group calibration error for the minority group is on average 1.5 times larger than the change for the majority group. A possible explanation for the training-time results is that label error affects the ability of the model to easily learn patterns for the minority group even at lower fractions of label error in the training data.

Second, we present an approach for estimating the 'influence' of perturbing a training point's label on a disparity metric of interest. We empirically assess the proposed approach on a variety of datasets and find a 10-40% improvement, compared to alternative approaches, in identifying training inputs that improve a model's disparity metric.

Our findings come with certain limitations. In this work, we focused on the influence of label error on disparity metrics. However, other components of the ML pipeline can also impact downstream model performance. The proposed empirical tests simulate the impact of label error; however, it might be the case that real-world label error is less pernicious to model learning dynamics than the synthetic flipping results suggest. Ultimately, we see our work as helping to provide insight and as an additional tool for practitioners seeking to address the challenge of label error particularly in relation to a disparity metric of interest.