RETRAINING WITH PREDICTED HARD LABELS PROVABLY INCREASES MODEL ACCURACY

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of a model trained with *noisy labels* is often improved by simply *retraining* the model with its own predicted *hard* labels (i.e., 1/0 labels). Yet, a detailed theoretical characterization of this phenomenon is lacking. In this paper, we theoretically analyze retraining in a linearly separable setting with randomly corrupted labels given to us and prove that retraining can improve the population accuracy obtained by initially training with the given (noisy) labels. To the best of our knowledge, this is the first such theoretical result. Retraining finds application in improving training with local label differential privacy (DP) which involves training with noisy labels. We empirically show that retraining selectively on the samples for which the predicted label matches the given label significantly improves label DP training at *no extra privacy cost*; we call this *consensus-based retraining*. As an example, when training ResNet-18 on CIFAR-100 with $\epsilon = 3$ label DP, we obtain 6.4% improvement in accuracy with consensus-based retraining.

1 INTRODUCTION

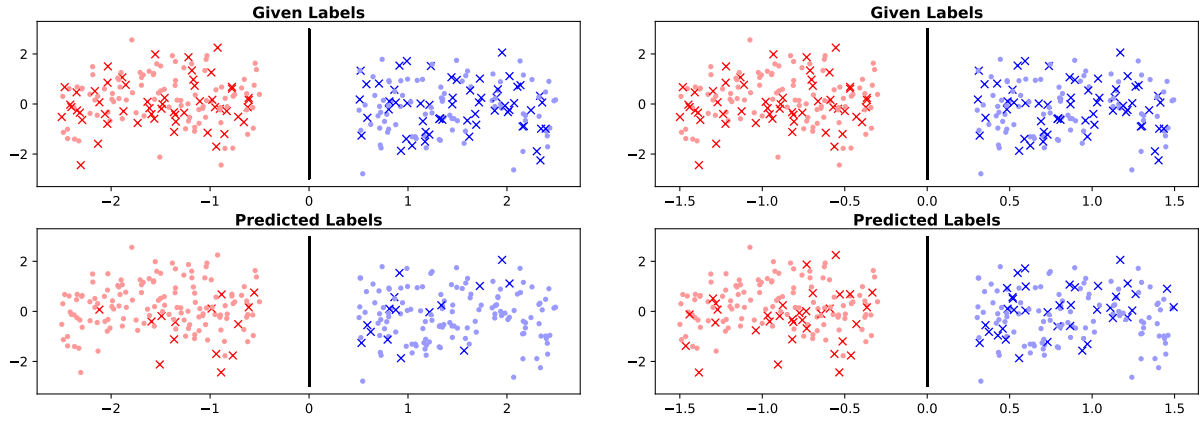
We study the simple idea of **retraining** an *already trained* model with its own predicted **hard** labels (i.e., 1/0 labels and *not* the raw probabilities) when the given labels with which the model is initially trained are **noisy**. This is a simple yet effective way to boost a model's performance in the presence of noisy labels. More formally, suppose we train a discriminative model $\mathcal{M}$ (for a classification problem) on a dataset of n samples and *noisy* label pairs $\{(\boldsymbol{x}_j, \widetilde{y}_j)\}_{j=1}^n$. Let $\widehat{\boldsymbol{\theta}}_0$ be the final learned weight/checkpoint of $\mathcal{M}$ and let $\widehat{y}_j = \mathcal{M}(\widehat{\boldsymbol{\theta}}_0, \boldsymbol{x}_j)$ be the current checkpoint's predicted *hard* label for sample $\boldsymbol{x}_j$. Now, we propose to *retrain* $\mathcal{M}$ with the $\widehat{y}_j$'s in one of the following two ways:

- (i) **Full retraining:** Retrain $\mathcal{M}$ with $\{(\boldsymbol{x}_j, \widehat{y}_j)\}_{j=1}^n$, i.e., retrain $\mathcal{M}$ with the *predicted* labels of *all* the samples.
- (ii) **Consensus-based retraining:** Define $\mathcal{S}_{\text{cons}} := \{j \in \{1, \ldots, n\} \mid \widetilde{y}_j = \widehat{y}_j\}$ to be the set of samples for which the predicted label matches the given noisy label; we call this the *consensus set*. Retrain $\mathcal{M}$ with $\{(\boldsymbol{x}_j, \widehat{y}_j)\}_{j \in \mathcal{S}_{\text{cons}}}$, i.e., retrain $\mathcal{M}$ with the *predicted* labels of *only the consensus set*.

Intuitively, retraining with predicted hard labels can be beneficial when the underlying classes are "**well-separated**". In such a case, the model can potentially correctly predict the labels of many samples in the training set far away from the decision boundary which were originally incorrectly labeled and presented to it. As a result, the model's accuracy (w.r.t. the actual labels) on the training data can be *significantly higher* than the accuracy of the noisy labels presented to it. Hence, retraining with predicted labels can potentially improve the model's performance. This intuition is illustrated in Figure 1 where we consider a *separable* binary classification problem with noisy labels. The exact details are in Appendix A but importantly, Figures 1a and 1b correspond to versions of this problem with "large" and "small" separation, respectively. Please see the figure caption for detailed discussion but in summary, Figure 1 shows us that *the success of retraining depends on the degree of separation between the classes*.

The motivation for *consensus-based retraining* is that matching the predicted and given labels can potentially yield a smaller but *much more accurate* subset compared to the entire set; such a filtering effect can further improve the model's performance. As we show in Section 5 (see Tables 3 and 7), this intuition bears out in practice.

There are plenty of ideas revolving around training a model with its own predictions, the two most common ones being *self-training* (Scudder, 1965; Yarowsky, 1995; Lee et al., 2013) and *self-distillation* (Furlanello et al., 2018; Mobahi et al., 2020); we discuss these and important differences from retraining in Section 2. However, from a *theoretical perspective*, we are not aware of any work proving that retraining a model with its predicted *hard* labels can be beneficial



(a) **Large** separation: predicted labels pretty accurate.

(b) **Small** separation: predicted labels not as accurate.

Figure 1: **Retraining Intuition.** Samples to the right (respectively, left) of the separator (black vertical line in the middle) and colored blue (respectively, red) have *actual* label $+1$ (respectively, -1). For both classes, the incorrectly labeled samples are marked by crosses ($\times$), whereas the correctly labeled samples are marked by dots ($\circ$) of the appropriate color. The amount of label noise and the number of training samples are the *same* in 1a and 1b. The top and bottom plots show the joint scatter plot of the training samples with the (noisy) labels given to us and the labels predicted by the model after training with the given labels, respectively. Notice that in 1a, *the model correctly predicts the labels of several samples that were given to it with the wrong label – especially, those that are far away from the separator*. This is not quite the case in 1b. This difference gets reflected in the *performance on the test set* after retraining. Specifically, in 1a, *retraining increases the test accuracy to* 97.67% *from* 89%. However, retraining yields no improvement in 1b. **So the success of retraining depends on the degree of separation between the classes.**

in the presence of label noise in any setting. In Section 4, **we derive the first theoretical result** (to our knowledge) showing that full retraining with hard labels *improves model accuracy*.

The primary reason for our interest in retraining is that it turned out to be a simple yet effective way to improve training with local[1] *label differential privacy (DP)* whose goal is to safeguard the privacy of labels in a supervised ML problem by injecting label noise (see Section 3 for a formal definition). Label DP is used in scenarios where only the labels are considered sensitive, e.g., advertising, recommendation systems, etc. (Ghazi et al., 2021). Importantly, *retraining can be applied on top of any label DP training algorithm at no extra privacy cost*. Our main *algorithmic contribution* is empirically demonstrating *the efficacy of consensus-based retraining in improving label DP training* (Section 5). Three things are worth clarifying here. First, as a meta-idea, retraining is not particularly new; however, its application – especially with consensus-based filtering – *as a light-weight way to improve label DP training at no extra privacy cost* is new to our knowledge. Second, we are *not* advocating consensus-based retraining as a SOTA general-purpose algorithm for learning with noisy labels. Third, we do not view full retraining to be an algorithmic contribution; we consider it for theoretical analysis and as a baseline for consensus-based retraining.

Our **main contributions** can be summarized as follows:

- In Section 4, we consider a linearly separable binary classification problem wherein the data (feature) dimension is d, and we are given randomly flipped labels with the label flipping probability being $p < \frac{1}{2}$ independently for each sample. *Our main result is proving that full retraining with the predicted hard labels improves the population accuracy obtained by initially training with the given labels*, provided that p is close enough to $\frac{1}{2}$ and the dataset size n satisfies $\frac{d \log n}{(1-2p)} \lesssim n \lesssim \frac{d^2}{(1-2p)^2}$ ("$\lesssim$" means "bounded asymptotically", ignoring constant factor multiples); see Remark 4.10 for details. In addition, our results show that retraining becomes more beneficial as the amount of label noise (i.e., p) increases or as the degree of separation between the classes increases. To

[1] In this paper, we focus only on local label DP. So throughout the paper, we will mostly omit the word "local" for conciseness.

the best of our knowledge, **these are the first theoretical results** quantifying the benefits of retraining with predicted hard labels in the presence of label noise. *The analysis of retraining is particularly challenging* because of the dependence of the predicted labels on the entire training set and the non-uniform/sample-dependent nature of label noise in the predicted labels; see the discussion after the statement of Theorem 4.8.

- In Section 5, we show the promise of **consensus-based retraining** (i.e., retraining on only those samples for which the predicted label matches the given noisy label) as a simple way to improve the performance of any label DP algorithm, at no extra privacy cost. As an example, when training ResNet-18 on CIFAR-100 with $\epsilon = 3$ label DP, we obtain 6.4% improvement in accuracy with consensus-based retraining (see Table 2). The corresponding improvement for a small BERT model trained on AG News Subset (a news classification dataset) with $\epsilon = 0.5$ label DP is 11.7% (see Table 6).

2 Related Work

Self-Training (ST). Retraining is similar in spirit to ST (Scudder, 1965; Yarowsky, 1995; Lee et al., 2013; Sohn et al., 2020) which is the process of progressively training a model with its own predicted hard labels in the *semi-supervised* setting. Our focus in this work is on the fully supervised setting. This is different from ST (in the semi-supervised setting) which typically selects samples based on the model's confidence and hence, we call our algorithmic idea of interest *retraining* to distinguish it from ST. In fact, we show that our consensus-based sample selection strategy leads to better performance than confidence-based sample selection in Appendix H. There is a vast body of work on ST and related ideas; see Amini et al. (2022) for a survey. On the theoretical side also, there are several papers showing and quantifying different kinds of benefits of ST and related ideas (Carmon et al., 2019; Raghunathan et al., 2020; Kumar et al., 2020; Chen et al., 2020; Oymak & Gulcu, 2020; Wei et al., 2020; Zhang et al., 2022). But *none of these works characterize the pros/cons of ST or any related algorithm in the presence of noisy labels*. In contrast, we show that retraining can provably improve accuracy in the presence of label noise in Section 4. *Empirically*, ST-based ideas have been proposed to improve learning with noisy labels (Reed et al., 2014; Tanaka et al., 2018; Han et al., 2019; Nguyen et al., 2019; Li et al., 2020; Goel et al., 2022); but these works do not have rigorous theory. In the context of theory on label noise and model's confidence, Zheng et al. (2020) show that if the model's predicted score for the observed label is small, then the observed label is likely not equal to the true label. Note that the results of Zheng et al. (2020) pertain to the correctness of the observed labels, whereas our results pertain to the correctness of the predicted labels.

Self-Distillation (SD). Retraining is also similar in principle to SD (Furlanello et al., 2018; Mobahi et al., 2020); the major difference is that *soft labels (i.e., predicted raw probabilities) are used in SD*, whereas we use hard labels in retraining. Specifically, in SD, a teacher model is first trained with provided hard labels and then its predicted *soft* labels are used to train a student model *with the same architecture as the teacher*. SD is usually employed with a temperature parameter (Hinton et al., 2015) to force the teacher and student models to be different; we do not have any such parameter in retraining as it uses hard labels. SD is known to ameliorate learning in the presence of noisy labels (Li et al., 2017) and this has been theoretically analyzed by Dong et al. (2019); Das & Sanghavi (2023). Dong et al. (2019) propose their own SD algorithm that uses *dynamically updated soft* labels and provide some complicated conditions of when their algorithm can learn the correct labels in the presence of noisy labels. In contrast, we analyze retraining with *fixed hard* labels. Das & Sanghavi (2023) analyze the standard SD algorithm in the presence of noisy labels with fixed *soft labels* but their analysis in the classification setting requires some strong assumptions such as access to the population, feature maps of all points in the same class having the same inner product, etc. We do not require such strong assumptions in this paper (in fact, we present sample complexity bounds). Moreover, Das & Sanghavi (2023) have extra ℓ_2-regularization in their objective function to force the teacher and student models to be different. We do not apply any extra regularization for retraining.

Label Differential Privacy (DP). Label DP (described in detail in Section 3) is a relaxation of full-data DP wherein the privacy of only the labels (and not the features) is safeguarded (Chaudhuri & Hsu, 2011; Beimel et al., 2013; Wang & Xu, 2019; Ghazi et al., 2021; Malek Esmaeili et al., 2021; Ghazi et al., 2022; Badanidiyuru et al., 2023). In this work, we are not trying to propose a SOTA label DP algorithm (with an ingenious noise-injection scheme); instead, we advocate retraining as a simple post-processing step that can be applied on top of any label DP algorithm (regardless of the noise-injection scheme) to improve its performance, at no extra privacy cost. Similar to our goal, Tang et al. (2022) apply techniques from unsupervised and semi-supervised learning to improve label DP training. In particular, one of their steps involves keeping the given noisy label of a sample only if it matches a pseudo-label generated by unsupervised learning. This is similar in spirit to our consensus-based retraining scheme but a crucial difference is

that we do not perform any unsupervised learning; we show that matching the given noisy label to the model's own predicted label is itself pretty effective. Further, unlike our work, Tang et al. (2022) do not have any rigorous theory.

3 Preliminaries

Notation. For two functions $f(n)$ and $g(n)$, we write $f(n) \lesssim g(n)$ if there exists n_0 and a constant $C > 0$ such that for all $n \geq n_0$, $f(n) \leq Cg(n)$. We also write $f(n) \ll g(n)$ to indicate that f is asymptotically dominated by g, namely, for every fixed real number $C > 0$, there exists $n(C)$ such that $g(n) \geq Cf(n)$ for all $n > n(C)$. For any positive integer $m \geq 1$, we denote the set $\{1, \ldots, m\}$ by $[m]$. Let $\boldsymbol{e}_i$ denote the i^{th} canonical vector, namely, the vector of all zeros except a one in the i^{th} coordinate. We denote the ℓ_2 norm of a vector $\boldsymbol{v}$ by $\|\boldsymbol{v}\|_{\ell_2}$, and the operator norm of a matrix $\boldsymbol{M}$ by $\|\boldsymbol{M}\|$. The unit d-dimensional sphere (i.e., the set of d-dimensional vectors with unit norm) is denoted by $\mathbb{S}^{d-1}$. For a random variable X, its sub-gaussian norm, denoted by $\|X\|_{\psi_2}$, is defined as $\|X\|_{\psi_2} \coloneqq \sup_{q \geq 1} q^{-1/2} \big(\mathbb{E}\,|X|^q \big)^{1/q}$.[2] In addition, for random vector $\boldsymbol{X} \in \mathbb{R}^d$, its sub-gaussian norm is defined as $\|\boldsymbol{X}\|_{\psi_2} = \sup_{\boldsymbol{z} \in \mathbb{S}^{d-1}} \|\langle \boldsymbol{X}, \boldsymbol{z} \rangle\|_{\psi_2}$. We denote the CDF and complementary CDF (CCDF) of a standard normal variable (i.e., distributed as $\mathsf{N}(0,1)$) by $\Phi(.)$ and $\Phi^{\mathrm{c}}(.)$, respectively.

Definition 3.1 (Label Differential Privacy (DP)) *A randomized algorithm $\mathcal{A}$ taking as input a dataset and with range $\mathcal{R}$ is said to be* ϵ-labelDP *if for any two datasets* D *and* D' *differing in the label of a single example and for any $\mathcal{S} \subseteq \mathcal{R}$, it holds that $\mathbb{P}\big(\mathcal{A}(\mathrm{D}) \in \mathcal{S}\big) \leq e^{\epsilon} \mathbb{P}\big(\mathcal{A}(\mathrm{D}') \in \mathcal{S}\big)$.*

Local label DP training involves injecting noise into the labels and then training with these noisy labels. The simplest way of injecting label noise to ensure label DP is randomized response (RR) introduced by Warner (1965). Specifically, suppose we require ϵ-labelDP for a problem with C classes, then the distribution of the output $\widehat{y}$ of RR when the true label is y is as follows:

$$\mathbb{P}\big(\widehat{y} = z\big) = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+C-1} & \text{for } z = y, \\ \frac{1}{e^{\epsilon}+C-1} & \text{otherwise.} \end{cases}$$

Our label noise model in Section 4 (eq. (4.2)) is actually RR for 2 classes. Based on vanilla RR, more sophisticated ways to inject label noise for better performance under label DP have been proposed (Ghazi et al., 2021; Malek Esmaeili et al., 2021). Our empirical results in Section 5 are with RR and the method of Ghazi et al. (2021).

4 Full Retraining in the Presence of Label Noise: Theoretical Analysis

Here we will analyze full retraining (as introduced in Section 1) for a linear setting with noisy labels. Since full retraining is the only kind of retraining we consider here, we will omit the word "full" subsequently in this section.

Problem Setting. We consider binary classification under a discriminative mixtures of Gaussian data model with a positive margin. We will first describe the classical Gaussian mixture model and then propose a new model which is endowed with a positive margin.

In the classical Gaussian mixture model, each data point belongs to one of two classes $\{\pm 1\}$ with corresponding probabilities π_+, π_-, such that $\pi_+ + \pi_- = 1$. Denoting by $y_i \in \{-1, +1\}$ the label for data point i, the feature vectors $\boldsymbol{x}_i \in \mathbb{R}^d$, for $i \in [n]$, are generated independently as $\boldsymbol{x}_i \sim \mathsf{N}(y_i \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$. In other words the mean of feature vectors are $\pm\boldsymbol{\mu}$ depending on its class, and $\boldsymbol{\Sigma}$ is the covariance of the features. We let $\gamma \coloneqq \|\boldsymbol{\mu}\|_{\ell_2}$.

Gaussian mixture model with positive margin: In this model, each data point $(\boldsymbol{x}, y)$ is generated independently by first sampling the label $y \in \{-1, +1\}$ with the corresponding probabilities π_-, π_+, and then generating the feature vector $\boldsymbol{x} \in \mathbb{R}^d$ as:

$$\boldsymbol{x} = y(1+u)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2} \boldsymbol{z}\,, \text{ where} \tag{4.1}$$

- u is drawn independently for each sample from a common sub-gaussian distribution with positive support ($u > 0$).
- $\boldsymbol{z} \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ independent of u.

[2] We refer to (Vershynin, 2010, Lemma 5.5) for other equivalent definitions.

- $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is the covariance matrix in the space orthogonal to $\boldsymbol{\mu}$. We have $\boldsymbol{\Sigma}^{1/2}\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma}$ is of rank $d-1$.

We let $\boldsymbol{\Sigma}$ be an arbitrary positive semi-definite matrix to handle general covariances. Let $\lambda_{\min} > 0$ and $\lambda_{\max}$ denote the minimum non-zero eigenvalue and the maximum eigenvalue of $\boldsymbol{\Sigma}$, respectively. Also without loss of generality, we assume that u has unit sub-gaussian norm ($\|u\|_{\psi_2} = 1$).

Note that under this model, the projection of datapoints on the orthogonal space of $\boldsymbol{\mu}$ is distributed as normal vectors. On the direction $\boldsymbol{\mu}$, we have $\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle = y(1+u)\|\boldsymbol{\mu}\|_{\ell_2}^2$, and so the randomness of data along this direction is modeled in u. Since u has a positive support, $\operatorname{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle) = y$ and therefore $\boldsymbol{\mu}$ is a separator of the data (based on its labels). In addition, we have a margin of at least $\gamma = \|\boldsymbol{\mu}\|_{\ell_2}$; this is because $\frac{y\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle}{\|\boldsymbol{\mu}\|_{\ell_2}} = (1+u)\|\boldsymbol{\mu}\|_{\ell_2} \geq \gamma$. We will mostly refer to the margin γ as the "degree of separation".

We are given access to a training set $\mathcal{T} = \{(\boldsymbol{x}_i, \widehat{y}_i)\}_{i\in[n]}$ where for each $i \in [n]$, $\widehat{y}_i$ is a noisy version of the true label y_i (which we do not observe). Specifically:

$$\widehat{y}_i = \begin{cases} y_i & \text{with probability } 1-p\,, \\ -y_i & \text{with probability } p\,, \end{cases} \tag{4.2}$$

for some $p < 1/2$, and independently for each $i \in [n]$.[3]

4.1 Vanilla Training

Given the training set $\mathcal{T} = \{(\boldsymbol{x}_i, \widehat{y}_i)\}_{i\in[n]}$, we consider the following linear classifier (Carmon et al., 2019):

$$\widehat{\boldsymbol{\theta}}_0 = \frac{1}{n}\sum_{i=1}^{n} \widehat{y}_i \boldsymbol{x}_i\,. \tag{4.3}$$

This classifier's predicted label for a sample $\boldsymbol{x} \in \mathbb{R}^d$ is $\operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle)$. Note that $\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle = (1/n)\sum_i \widehat{y}_i \langle \boldsymbol{x}_i, \boldsymbol{x}\rangle$ is an average of noisy labels in the training set with weights given by $\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle/n$. Therefore, it is similar to kernel methods with the inner product kernel.

Our next result bounds the probability of $\widehat{\boldsymbol{\theta}}_0$ correctly classifying a given fixed test point $\boldsymbol{x}$.

Theorem 4.1 (Vanilla Training) *Consider $\boldsymbol{x} \notin \mathcal{T}$ and let y be its true label. We have*

$$\mathbb{P}(\operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle) = y) \leq \alpha_0(\boldsymbol{x}) := 1 - \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{5(1+\sqrt{n}(1-2p))^2\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) \text{ and}$$

$$\mathbb{P}(\operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle) = y) \geq \widetilde{\alpha}_0(\boldsymbol{x}) := 1 - \frac{1}{2}\exp\left(-\frac{n(1-2p)^2\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{4\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) - e\exp\left(-cn(1-2p)^2\right),$$

for an absolute constant $c > 0$.

The proof of Theorem 4.1 is in Appendix B. Notice that the learned classifier $\widehat{\boldsymbol{\theta}}_0$ is more likely to be wrong on the samples that are less aligned with (or closer to orthogonal to) the ground truth separator, i.e., $\boldsymbol{\mu}$. We can view $\widehat{\boldsymbol{\theta}}_0$ as a *noisy* label provider (a.k.a. pseudo-labeler) where the degree of label noise is **non-uniform** or **sample-dependent** unlike the original noisy source used to learn $\widehat{\boldsymbol{\theta}}_0$. Specifically, for a sample $\boldsymbol{x}$ with true label y and predicted label $\widetilde{y} = \operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle)$, we have:

$$\widetilde{y} = \begin{cases} y \text{ with probability at least } \geq \widetilde{\alpha}_0(\boldsymbol{x})\,, \\ -y \text{ with probability at most } \leq 1 - \widetilde{\alpha}_0(\boldsymbol{x}), \text{ where } \widetilde{\alpha}_0(\boldsymbol{x}) \text{ is as defined in Theorem 4.1.} \end{cases}$$

We will next define the accuracy of a classifier $\widehat{\boldsymbol{\theta}}$ as

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}) := \mathbb{P}(\operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}\rangle) = y)\,, \tag{4.4}$$

[3] In the context of ϵ-labelDP for a binary classification problem with randomized response, $p = \frac{1}{e^\epsilon + 1}$.

where the probability is with respect to the randomness in the training set used to learn $\widehat{\boldsymbol{\theta}}$ as well as $(\boldsymbol{x}, y)$. Our next result bounds the accuracy of the classifier $\widehat{\boldsymbol{\theta}}_0$ obtained by vanilla training.

Theorem 4.2 (Vanilla Training: Population Accuracy) *We have*

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) \le 1 - \frac{1}{4\sqrt{2\pi}} \exp\left(-160(1+\sqrt{n}(1-2p))^2 \frac{\gamma^4}{\lambda_{\min}^2 d}\right)\left(1 - e^{-d/16}\right) \quad \textit{and} \tag{4.5}$$

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) \ge 1 - \frac{1}{2} \exp\left(-\frac{n(1-2p)^2\gamma^4}{8\lambda_{\max}^2 d}\right) - e^{-d/8} - e\exp\left(-cn(1-2p)^2\right), \tag{4.6}$$

for an absolute constant $c > 0$.

The proof of Theorem 4.2 is in Appendix C.

Remark 4.3 (Tightness of Accuracy Bounds) *When $n \lesssim \frac{d^2}{(1-2p)^2\gamma^4}$, $\gamma \lesssim d^{1/4}$ and $\frac{\lambda_{\max}}{\lambda_{\min}} \lesssim 1$, then the lower and upper bounds for $\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0)$ in Theorem 4.2 match (up to constant factors).*

Based on Theorem 4.2, we have the following corollary.

Corollary 4.4 (Vanilla Training: Sample Complexity) *For any $\delta > 2e^{-d/8} + 2e\exp\left(-cn(1-2p)^2\right)$, having sample size $n \ge 8\lambda_{\max}\frac{\log 1/\delta}{(1-2p)^2}\frac{d}{\gamma^4}$ ensures that $\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) > 1-\delta$. In particular, when the label flipping probability satisfies $p > 2e^{-d/8} + 2e\exp\left(-cn(1-2p)^2\right)$, then $n \ge 8\lambda_{\max}\frac{\log 1/\delta}{(1-2p)^2}\frac{d}{\gamma^4}$ ensures that $\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) > 1-p$, i.e., our learned classifier $\widehat{\boldsymbol{\theta}}_0$ has better accuracy than the source providing noisy labels (used to learn $\widehat{\boldsymbol{\theta}}_0$).*

Remark 4.5 (Effect of Degree of Separation) *As the parameter quantifying the degree of separation γ decreases, the accuracy bound in Theorem 4.2 also decreases and the sample complexity required to outperform the noisy label source in Corollary 4.4 increases. This is consistent with our intuition that a classification task should become harder as the degree of separation reduces; we also saw this in Figure 1.*

We conclude this section by deriving an information-theoretic lower bound on the sample complexity of *any classifier* to argue that $\widehat{\boldsymbol{\theta}}_0$ attains the optimal sample complexity with respect to d and p.

Theorem 4.6 (Information-Theoretic Lower Bound on Sample Complexity) *With a slight generalization of notation, let $\mathsf{acc}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu})$ denote the accuracy of the classifier $\widehat{\boldsymbol{\theta}}$, when the ground truth separator is $\boldsymbol{\mu}$. We also consider the case of $\boldsymbol{\Sigma} = \mathcal{P}_{\boldsymbol{\mu}}^{\perp}$, viz., the projection matrix onto the space orthogonal to $\boldsymbol{\mu}$. For any classifier $\widehat{\boldsymbol{\theta}}$ learned from $\mathcal{T} := \{(\boldsymbol{x}_j, \widetilde{y}_j)\}_{j\in[n]}$, in order to achieve $\inf_{\boldsymbol{\mu}\in\mathbb{S}^{d-1}} \mathsf{acc}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu}) \ge 1-\delta$, the condition $n = \Omega\left(\frac{(1-\delta)}{(1-2p)^2}d\right)$ is necessary in our problem setting.*

It is worth mentioning that there is a similar lower bound in Gentile & Helmbold (1998) for a different classification setting. In contrast, Theorem 4.6 is tailored to our setting and moreover, the proof technique is also different and interesting in its own right. Specifically, for the proof of Theorem 4.6, we follow a standard technique in proving minimax lower bounds which is to reduce the problem of interest to an appropriate multi-way hypothesis testing problem; this is accompanied by the application of the conditional version of Fano's inequality and some ideas from high-dimensional geometry. This proof is in Appendix D.

Remark 4.7 (Minimax Optimality of Sample Complexity) *Note that the dependence of the sample complexity on d and p in Corollary 4.4 matches that of the lower bound in Theorem 4.6. Thus, our sample complexity bound in Corollary 4.4 is optimal with respect to d and p.*

4.2 Retraining

We first label the training set using $\widehat{\boldsymbol{\theta}}_0$. Denote by $\widetilde{y}_i = \mathrm{sign}(\langle \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}_0\rangle)$ the predicted label for sample $\boldsymbol{x}_i$. We then retrain the model using these predicted labels. Our learned classifier here is similar to the one in Section 4.1, except that the

observed labels are replaced by the predicted labels. Specifically, our retraining classifier is the following:

$$\widehat{\boldsymbol{\theta}}_1 = \frac{1}{n}\sum_i \widetilde{y}_i \boldsymbol{x}_i\,. \tag{4.7}$$

Our retraining classifier's predicted label for a sample $\boldsymbol{x} \in \mathbb{R}^d$ is $\text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle)$. Our next result bounds the probability of $\widehat{\boldsymbol{\theta}}_1$ correctly classifying a given fixed test point $\boldsymbol{x}$.

Theorem 4.8 (Retraining) *Suppose that* $\frac{n}{d} > \frac{4\lambda_{\max}}{\gamma^2(1-2p)}$ *and* $nd > \frac{\gamma^4}{\lambda_{\max}^2}$. *Consider* $\boldsymbol{x} \notin \mathcal{T}$ *and let* y *be its true label. We have*

$$\mathbb{P}(\text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle) = y) \geq \alpha_1(\boldsymbol{x})\,,$$

where

$$\alpha_1(\boldsymbol{x}) := 1 - \exp\left(-\frac{cn(1-2q')^2\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) - e\exp\left(-cn(1-2p)^2\right) - \frac{n}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/16}\right)e^{d/n}\,,$$

for an absolute constant $c > 0$, *with*

$$q' = \exp\left(-\frac{n(1-2p)\gamma^2}{40\lambda_{\max}}\right) \text{ and } p' = \left(1 + \frac{3\gamma^4}{8\lambda_{\max}^2 nd}\right)p\,.$$

The proof of Theorem 4.8 is in Appendix E.

Technical Challenges. The proof of Theorem 4.8 is especially challenging because as we discussed after Theorem 4.1, the classifier learned with vanilla training, i.e., $\widehat{\boldsymbol{\theta}}_0$, is a *non-uniform noisy label provider*. In addition, $\widehat{\boldsymbol{\theta}}_0$ depends on all the samples in the training set and hence each predicted label $\widetilde{y}_i$ is also dependent on the entire training set. *This dependence of the predicted labels on all the data points makes the analysis even more challenging because standard arguments with independence are not applicable here.* In contrast, the analysis of $\widehat{\boldsymbol{\theta}}_0$ was technically simpler as it was based on noisy labels $\widetilde{y}_i$'s that were independent across samples. The high-level proof idea for Theorem 4.8 is that for each sample $\boldsymbol{x}_\ell = y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell$ in the training set, we carefully curate a dummy label $\widetilde{y}'_\ell$ which does not depend on $\boldsymbol{z}_k$ for $k \neq \ell$, but depends on all $\{u_k\}_{k\in[n]}$; see (E.2) in Appendix E. We show that with high probability the labels predicted by the vanilla estimator $\widehat{\boldsymbol{\theta}}_0$ on the training samples, i.e. $\text{sign}(\langle \boldsymbol{x}_\ell, \widehat{\boldsymbol{\theta}}_0\rangle)$ are the same as the dummy labels $\widetilde{y}'_\ell$. Next, under this high probability event, we use the dummy labels instead, which are more amenable to analysis as they are independent of $\{\boldsymbol{z}_k\}_{k\neq\ell}$. At the beginning of Appendix E, we provide an outline of these steps in more detail.

It is worth noting that the lower bound derived on $\alpha_1(\boldsymbol{x})$ is increasing in $\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle$. So, just like the vanilla training classifier $\widehat{\boldsymbol{\theta}}_0$, the classifier $\widehat{\boldsymbol{\theta}}_1$ learned with retraining is more likely to be wrong on the samples that are less aligned with the ground truth separator $\boldsymbol{\mu}$. We will now provide a lower bound on the accuracy (defined in (4.4)) of the classifier $\widehat{\boldsymbol{\theta}}_1$.

Theorem 4.9 (Retraining: Population Accuracy) *Under the setting of Theorem 4.8 we have*

$$\begin{aligned}\text{acc}(\widehat{\boldsymbol{\theta}}_1) \geq 1 &- \exp\left(-\frac{cn(1-2q')^2\gamma^4}{\gamma^4 + 2\lambda_{\max}^2 d}\right) - e^{-d/8} \\ &- e\exp\left(-cn(1-2p)^2\right) - \frac{n}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/16}\right)e^{d/n}\,.\end{aligned} \tag{4.8}$$

The proof of Theorem 4.9 is in Appendix F. As expected, the lower bound (4.8) is increasing in γ, because as the degree of separation increases, the classification task becomes easier.

Accuracy of Retraining vs. Vanilla Training. We will now compare the accuracy of vanilla training model $\widehat{\boldsymbol{\theta}}_0$ with that of the retrained model $\widehat{\boldsymbol{\theta}}_1$, identifying regimes where retraining improves the model accuracy. Before proceeding further, it is useful to note that the dominant term in $1 - \text{acc}(\widehat{\boldsymbol{\theta}}_1)$ is the last term in (4.8) which has $(1-2p')$ inside the exponent and p' can become arbitrarily close to p as n or d grows. In contrast, the upper bound on $\text{acc}(\widehat{\boldsymbol{\theta}}_0)$ in (4.5) has $(1-2p)^2$ inside the exponent. *This is the main observation which indicates that retraining can improve model accuracy, and this improvement becomes increasingly significant as* p *approaches* $\frac{1}{2}$. We will now formalize this observation by providing sufficient conditions for this improvement to happen.

Remark 4.10 (When Does Retraining Improve Accuracy?) *We consider an asymptotic regime where $n, d \to \infty$. Also, suppose $\frac{\lambda_{\max}}{\lambda_{\min}} \lesssim 1$. We would like to characterize a regime where $1 - \mathsf{acc}(\widehat{\boldsymbol{\theta}}_1) \leq 1 - \mathsf{acc}(\widehat{\boldsymbol{\theta}}_0)$, i.e., under what conditions the misclassification error of $\widehat{\boldsymbol{\theta}}_1$ is smaller than that of $\widehat{\boldsymbol{\theta}}_0$. By comparing the bounds in Theorems 4.2 and 4.9, we obtain the following sufficient conditions. There exists an absolute constant c_0 such that if $p \in (\frac{1}{2} - c_0, \frac{1}{2})$ and*

$$\frac{d}{\gamma^2(1-2p)} \max\left\{1, \frac{\log n}{\gamma^2}\right\} \lesssim n \lesssim \frac{d^2}{\gamma^4(1-2p)^2}\,, \tag{4.9}$$

*then the lower bound on the accuracy of retraining (in eq. (4.8)) is **greater** than the upper bound on the accuracy of vanilla training (in eq. (4.5)).*

Note that as the degree of separation γ decreases, the range of n in (4.9) becomes larger but also shifts up, indicating we need more samples (keeping other parameters fixed) to obtain an improvement with retraining.

5 Improving Label DP Training with Retraining (RT)

Motivated by our theoretical results in Section 4 which show that retraining (abbreviated as RT henceforth) can improve accuracy in the presence of label noise, we propose to apply our proposals in Section 1, viz., full RT and more importantly, *consensus-based RT* to improve label DP training (because it involves noisy labels). Note that this **can be done on top of any label DP mechanism** and that too **at no additional privacy cost** (both the predicted labels and originally provided noisy labels are private). We empirically evaluate full and consensus-based RT on three classification datasets (available on TensorFlow) *trained with label DP*. These include two vision datasets, namely CIFAR-10 and CIFAR-100, and one language dataset, namely AG News Subset (Zhang et al., 2015). All the empirical results are averaged over three different runs. We only provide important experimental details here; the other details can be found in Appendix G.

CIFAR-10/100. We train a ResNet-18 model on CIFAR-10 and CIFAR-100 with label DP. Label DP training is done with the prior-based method of Ghazi et al. (2021) – specifically, Alg. 3 with two stages. Our training set consists of 45k examples and we assume access to a validation set with clean labels consisting of 5k examples which we use for deciding when to stop training, setting hyper-parameters, etc.[4] For CIFAR-10 and CIFAR-100 with three different values of ϵ, we list the test accuracies of the baseline (i.e., the method of Ghazi et al. (2021)), full RT and consensus-based RT in Tables 1 and 2, respectively. Notice that *consensus-based RT is the clear winner*. Also, for the three values of ϵ in Table 1 (CIFAR-10), the size of the consensus set (used in consensus-based RT) is $\sim 31\%$, 55% and 76%, respectively, of the entire training set. The corresponding numbers for Table 2 (CIFAR-100) are $\sim 11\%$, 34% and 56%, respectively. So for small ϵ (high label noise), *consensus-based RT comprehensively outperforms full RT and baseline with a small fraction of the training set*. Further, in Table 3, we list the accuracies of predicted labels and given labels over the entire (training) dataset and accuracies of predicted labels (which are the same as the given labels) over the consensus set for CIFAR-10 and CIFAR-100. To summarize, the accuracy of predicted labels over the consensus set is significantly more than the accuracy of predicted and given labels over the entire dataset. This gives us an idea of why consensus-based RT is much better than full RT and baseline, even though the consensus set is smaller than full dataset.

Table 1: **CIFAR-10.** Test set accuracies (mean ± standard deviation). *Consensus-based RT is better than full RT which is better than the baseline*.

ϵ	Baseline	Full RT	Consensus-based RT
1	57.78 ± 1.13	60.07 ± 0.63	**63.84** ± 0.56
2	79.06 ± 0.59	81.34 ± 0.40	**83.31** ± 0.28
3	85.18 ± 0.50	86.67 ± 0.28	**87.67** ± 0.28

One may wonder how more model parameters affect the results due to potential overfitting. So here we consider training ResNet-34 (which has more parameters than ResNet-18) on CIFAR-100 with label DP; the setup is exactly the same as our previous experiments with ResNet-18. Please see the results and discussion in Table 4. In summary,

[4] In practice, we do not need full access to the validation set. Instead, the validation set can be stored by a secure agent which returns us a private version of the validation accuracy and this will not be too far off from the true validation accuracy when the validation set is large enough.

Table 2: **CIFAR-100.** Test set accuracies (mean ± standard deviation). Overall, *consensus-based RT is **significantly** better than full RT which is somewhat better than the baseline*.

ϵ	Baseline	Full RT	Consensus-based RT
3	23.53 ± 1.01	24.42 ± 1.22	**29.98** ± 1.11
4	44.53 ± 0.81	46.99 ± 0.66	**51.30** ± 0.98
5	55.75 ± 0.36	56.98 ± 0.43	**59.47** ± 0.26

Table 3: **CIFAR-10 (Top) and CIFAR-100 (Bottom).** Accuracies of predicted labels and given labels over the entire (training) dataset and accuracies of predicted labels over the consensus set. Note that the *accuracy over the consensus set* $\gg$ *accuracy of over the entire dataset* (with both predicted and given labels). This gives us an idea of why consensus-based RT is much better than full RT and baseline, even though the consensus set is smaller than the full dataset ($\sim 31\%$, 55% and 76% of the full dataset for $\epsilon = 1, 2$ and 3 in the case of CIFAR-10, and $\sim 11\%$, 34% and 56% of the full dataset for $\epsilon = 3, 4$ and 5 in the case of CIFAR-100).

	ϵ	Acc. of **predicted** labels on **full** dataset	Acc. of **given** labels on **full** dataset	Acc. of **predicted** labels on **consensus** set
CIFAR-10	1	59.30 ± 0.74	32.61 ± 0.74	**76.17** ± 0.15
	2	81.62 ± 0.18	57.11 ± 0.05	**92.65** ± 0.22
	3	89.28 ± 0.35	76.73 ± 0.12	**95.94** ± 0.23

	ϵ	Acc. of **predicted** labels on **full** dataset	Acc. of **given** labels on **full** dataset	Acc. of **predicted** labels on **consensus** set
CIFAR-100	3	24.90 ± 0.92	22.35 ± 0.41	**76.09** ± 0.85
	4	50.85 ± 0.82	46.32 ± 0.34	**91.59** ± 1.24
	5	66.51 ± 0.02	68.09 ± 0.33	**94.83** ± 0.15

consensus-based RT is pretty effective even with ResNet-34 but the amount of improvement here is less compared to ResNet-18 due to overfitting. Additionally, one may wonder if retraining is useful when a label noise-robust technique is used during initial training, i.e., in the baseline (the method of Ghazi et al. (2021)). To that end, we once again train ResNet-34 on CIFAR-100 but with the popular noise-correcting technique of "forward correction" (Patrini et al., 2017) applied to the first stage of our baseline (it is not clear how to apply it to the second stage).[5] Please see the result and discussion for $\epsilon = 5$ in Table 5. In summary, **retraining can offer gains even after noise-robust training**.

Table 4: **CIFAR-100 w/ ResNet-34.** Test set accuracies (mean ± standard deviation). Just like Table 2 (ResNet-18), consensus-based RT is the clear winner here. However, note that the performance and amount of improvement with ResNet-34 is worse than ResNet-18 due to more overfitting because of more parameters; this is expected.

ϵ	Baseline	Full RT	Consensus-based RT
3	17.53 ± 0.05	18.20 ± 0.29	**21.63** ± 0.31
4	37.53 ± 1.58	39.33 ± 1.37	**43.87** ± 1.62
5	51.13 ± 0.69	52.33 ± 0.38	**55.43** ± 0.42

Table 5: **Noise-robust "forward correction" (Patrini et al., 2017) applied to baseline (CIFAR-100 w/ ResNet-34).** Test set accuracies (mean ± standard deviation). We see that *retraining can yield improvement even after noise-robust training*, although the amount of improvement is less compared to Table 4 (no noise-correction).

ϵ	Baseline	Full RT	Consensus-based RT
5	53.83 ± 0.83	54.43 ± 0.48	**56.60** ± 0.43

[5] Forward correction worked the best among other popular noise-robust techniques such as backward correction (Patrini et al., 2017) and noise-robust loss functions (Wang et al., 2019; Zhang & Sabuncu, 2018).

AG News Subset (https://www.tensorflow.org/datasets/catalog/ag_news_subset). This is a news article classification dataset consisting of 4 categories – world, sports, business or sci/tech. We reserve 10% of the given training set for validation and we use the rest for training with label DP. Just like the CIFAR experiments, we assume that the validation set comes with clean labels. We use the small BERT model available in TensorFlow and the BERT English uncased preprocessor; links to both of these are in Appendix G. We pool the output of the BERT encoder, add a dropout layer with probability = 0.2, followed by a softmax layer. We fine-tune the full model. Here, label DP training is done with randomized response. We list the test accuracies of the baseline (i.e., randomized response), full RT and consensus-based RT in Table 6 for three different values of ϵ. Even here, *consensus-based RT is the clear winner*. For the three values of ϵ in Table 6, the size of the consensus set (used in consensus-based RT) is $\sim 28\%$, 32% and 38%, respectively, of the entire training set. So here, *consensus-based RT appreciably outperforms full RT and baseline with less than two-fifths of the entire training set*. Finally, in Table 7, we list the accuracies of predicted labels and given labels over the entire dataset and accuracies of predicted labels (= given labels) over the consensus set. Even here, the accuracy of predicted labels over the consensus set is significantly more than the accuracy of predicted and given labels over the entire dataset. This explains why consensus-based RT performs the best, even though the consensus set is much smaller than the full dataset.

Table 6: **AG News Subset.** Test set accuracies (mean ± standard deviation). *Consensus-based RT is better than full RT which is better than the baseline*.

ϵ	Baseline	Full RT	Consensus-based RT
0.3	54.54 ± 0.97	60.03 ± 2.90	**65.91** ± 1.93
0.5	69.21 ± 0.31	75.63 ± 1.08	**80.95** ± 1.47
0.8	79.10 ± 1.43	82.19 ± 1.54	**84.26** ± 1.03

Table 7: **AG News Subset.** Accuracies over the entire dataset and over the consensus set. Conclusions are the same as Table 3.

ϵ	Acc. of **predicted** labels on **full** dataset	Acc. of **given** labels on **full** dataset	Acc. of **predicted** labels on **consensus** set
0.3	53.20 ± 2.82	32.52 ± 2.05	**61.81** ± 2.66
0.5	66.78 ± 1.31	35.5 ± 0.14	**76.48** ± 0.93
0.8	79.98 ± 0.80	42.53 ± 0.13	**89.59** ± 0.43

So in summary, **consensus-based RT significantly improves model accuracy**.

Additional Empirical Results in the Appendix. In Appendix H, we show that consensus-based RT also outperforms retraining on samples for which the model is the most confident; this is similar to self-training's method of sample selection in the semi-supervised setting. In Appendix I, we show that RT is beneficial even without a clean validation set. Furthermore, going beyond label DP, we show that consensus-based RT is also beneficial in the presence of human annotation errors which can be thought of as "real" label noise in Appendix J.

6 Conclusion

In this work, we provided the first theoretical result showing retraining with hard labels can provably increase model accuracy in the presence of label noise. We also showed the efficacy of consensus-based retraining (i.e., retraining on only those samples for which the predicted label matches the given noisy label) in improving label DP training at no extra privacy cost. We will conclude by discussing some future directions of work. Our theoretical results in this work focus on the full retraining scheme. Given that consensus-based retraining worked very well empirically, we would like to analyze it theoretically as a future work. Another potential extension of our work is to analyze retraining under non-uniform label noise models. We also hope to test our ideas on larger scale models and datasets in the future.

References

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1968.

Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Emilie Devijver, and Yury Maximov. Self-training: A survey. *arXiv preprint arXiv:2202.12040*, 2022.

Ashwinkumar Badanidiyuru, Badih Ghazi, Pritish Kamath, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Avinash V Varadarajan, and Chiyuan Zhang. Optimal unbiased randomizers for regression with label differential privacy. *arXiv preprint arXiv:2312.05659*, 2023.

Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31(1-58):26, 1997.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pp. 363–378. Springer, 2013.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 155–186. JMLR Workshop and Conference Proceedings, 2011.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020.

Rudrajit Das and Sujay Sanghavi. Understanding self-distillation in the presence of label noise. In *International Conference on Machine Learning*, pp. 7102–7140. PMLR, 2023.

Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. Distillation ≈ early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network. *arXiv preprint arXiv:1910.01255*, 2019.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Claudio Gentile and David P Helmbold. Improved lower bounds for learning from noisy examples: An information-theoretic approach. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 104–115, 1998.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in neural information processing systems*, 34:27131–27145, 2021.

Badih Ghazi, Pritish Kamath, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Avinash V Varadarajan, and Chiyuan Zhang. Regression with label differential privacy. *arXiv preprint arXiv:2212.06074*, 2022.

Malay Ghosh. Exponential tail bounds for chisquared random variables. *Journal of Statistical Theory and Practice*, 15 (2):35, 2021.

Arushi Goel, Yunlong Jiao, and Jordan Massiah. Pars: Pseudo-label aware robust sample selection for learning with noisy labels. *arXiv preprint arXiv:2201.10836*, 2022.

Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5138–5147, 2019.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International conference on machine learning*, pp. 5468–5479. PMLR, 2020.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

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

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1910–1918, 2017.

Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34:6934–6945, 2021.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. *arXiv preprint arXiv:1910.01842*, 2019.

Samet Oymak and Talha Cihad Gulcu. Statistical and algorithmic insights for semi-supervised learning with self-training. *arXiv preprint arXiv:2006.11006*, 2020.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1944–1952, 2017.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

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.

Jonathan Scarlett and Volkan Cevher. An introductory guide to fano's inequality with applications in statistical estimation. *arXiv preprint arXiv:1901.00555*, 2019.

Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5552–5560, 2018.

Xinyu Tang, Milad Nasr, Saeed Mahloujifar, Virat Shejwalkar, Liwei Song, Amir Houmansadr, and Prateek Mittal. Machine learning with differentially private labels: Mechanisms and frameworks. *Proceedings on Privacy Enhancing Technologies*, 2022.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. In *International Conference on Machine Learning*, pp. 6628–6637. PMLR, 2019.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 322–330, 2019.

Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. How does unlabeled data improve generalization in self-training? a one-hidden-layer theoretical analysis. *arXiv preprint arXiv:2201.08514*, 2022.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pp. 11447–11457. PMLR, 2020.

Appendix

CONTENTS

A Problem Setting of Figure 1

The setting is exactly the same as the problem setting in Section 4 with $\boldsymbol{\mu} = \gamma \boldsymbol{e}_1$, $\boldsymbol{\Sigma} = \boldsymbol{I} - \boldsymbol{e}_1 \boldsymbol{e}_1^\top$, $d = 50$ and $p = 0.4$. In the direction of $\boldsymbol{\mu}$, we have $\langle \boldsymbol{x}, \boldsymbol{\mu} \rangle = y(1+u) \|\boldsymbol{\mu}\|_{\ell_2}^2 = y(1+u)\gamma^2$. We consider balanced classes (i.e., $\pi_+ = \pi_- = \frac{1}{2}$) and choose $u \sim \mathsf{Unif}[0, 4]$, the uniform distribution over the interval $[0, 4]$. In Figure 1a, $\gamma^2 = 0.5$ (large separation) and in Figure 1b, $\gamma^2 = 0.3$ (small separation). The number of training samples in each case is 300 and the retraining is done on the same training set on which the model is initially trained. The learned classifiers from vanilla training and retraining are the same as in Section 4 (i.e., eq. (4.3) and eq. (4.7), respectively). Finally, the test accuracy of both vanilla training and retraining in Figure 1b is 68%.

B Proof of Theorem 4.1

We first prove the upper bound $\alpha_0(\boldsymbol{x})$ and then prove the lowr bound $\widetilde{\alpha}_0(\boldsymbol{x})$.

B.1 Upper bound $\alpha_0(\mathbf{x})$

Let $\xi_i = y_i \widehat{y}_i$. So $\xi_i = 1$ with probability $1 - p$ and $\xi_i = -1$ with probability p. Under our data model, for each sample we can write $\boldsymbol{x}_i = y_i(1+u_i)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_i$ with $\boldsymbol{z}_i \sim \mathsf{N}(0, \boldsymbol{I}_d)$. Hence,

$$\widehat{\boldsymbol{\theta}}_0 = \sum_{i \in [n]} \widehat{y}_i \boldsymbol{x}_i = \Big(\sum_{i \in [n]} \xi_i (1+u_i) \Big) \boldsymbol{\mu} + \sum_{i \in [n]} \widehat{y}_i \boldsymbol{\Sigma}^{1/2} \boldsymbol{z}_i \,.$$

We also note that

$$\mathbb{P}(\mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0 \rangle) \neq y) = \mathbb{P}(y \langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0 \rangle \leq 0) = \mathbb{P}\left(\Big(\sum_{i \in [n]} \xi_i (1+u_i) \Big) \langle y\boldsymbol{x}, \boldsymbol{\mu} \rangle + \Big\langle \boldsymbol{x}, \sum_{i \in [n]} y \widehat{y}_i \boldsymbol{\Sigma}^{1/2} \boldsymbol{z}_i \Big\rangle \leq 0 \right)$$

Define $\tilde{\boldsymbol{z}} := \sum_{i \in [n]} y \widehat{y}_i \boldsymbol{\Sigma}^{1/2} \boldsymbol{z}_i$. Conditioning on $\{\widehat{y}_i\}_{i \in [n]}$ we have $\tilde{\boldsymbol{z}} \sim \mathsf{N}(0, n\boldsymbol{\Sigma})$, and so $\langle \boldsymbol{x}, \tilde{\boldsymbol{z}} \rangle \sim \mathsf{N}(0, n \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_{\ell_2}^2)$.

We write

$$\mathbb{P}\left(\Big(\sum_{i \in [n]} \xi_i (1+u_i) \Big) \langle y\boldsymbol{x}, \boldsymbol{\mu} \rangle + \langle \boldsymbol{x}, \tilde{\boldsymbol{z}} \rangle \leq 0 \;\middle|\; \{\xi_i, y_i, u_i\}_{i \in [n]} \right) = \Phi\left(- \frac{\big(\sum_{i \in [n]} \xi_i (1+u_i)\big) \langle y\boldsymbol{x}, \boldsymbol{\mu} \rangle}{\sqrt{n} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_{\ell_2}} \right),$$

where $\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-s^2/2} \mathrm{d}s$ denotes the cdf of standard normal distribution.

Combining the previous two equations, we arrive at

$$\begin{aligned} \mathbb{P}(\mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0 \rangle) \neq y) &= \mathbb{E}\left[\Phi\left(- \frac{\big(\sum_{i \in [n]} \xi_i (1+u_i)\big) \langle y\boldsymbol{x}, \boldsymbol{\mu} \rangle}{\sqrt{n} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_{\ell_2}} \right) \right] \\ &\geq \mathbb{E}\left[\Phi\left(- \frac{\big|\sum_{i \in [n]} \xi_i (1+u_i)\big| \, |\langle \boldsymbol{x}, \boldsymbol{\mu} \rangle|}{\sqrt{n} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_{\ell_2}} \right) \right] \\ &\geq \Phi\left(- \frac{\mathbb{E}\big[\big|\sum_{i \in [n]} \xi_i (1+u_i)\big|\big] \, |\langle \boldsymbol{x}, \boldsymbol{\mu} \rangle|}{\sqrt{n} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_{\ell_2}} \right) \end{aligned} \tag{B.1}$$

where the expectation is with respect to $\{\xi_i, i \in [n]\}$ and the last step follows by Jensen's inequality and the fact that $\Phi(t)$ is a convex function for $t \leq 0$.

We next note that by Cauchy–Schwarz inequality,

$$\begin{aligned}\mathbb{E}\left[\left|\sum_i \xi_i(1+u_i)\right|\right] &\le \sqrt{\mathbb{E}[(\sum_i \xi_i(1+u_i))^2]} \\ &= \sqrt{\sum_{i,j}\mathbb{E}[\xi_i\xi_j(1+u_i)(1+u_j)]} \\ &= \sqrt{\sum_i \mathbb{E}[(1+u_i)^2] + \sum_{i\neq j}\mathbb{E}[\xi_i]\,\mathbb{E}[\xi_j]\,\mathbb{E}[1+u_i]\,\mathbb{E}[1+u_j]} \\ &= \sqrt{5n+4n(n-1)(1-2p)^2} \le \sqrt{5n}+\sqrt{5}n(1-2p)\,,\end{aligned}$$

where we used that ξ_i, ξ_j are independent for $i \neq j$, and independent from u_i's. In addition, $\mathbb{E}[u_i] \le 1$ and $\mathbb{E}[u_i^2] \le 2$, since u_i has unit sub-gaussian norm. Hence, we get

$$\mathbb{P}(\mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle) \neq y) \ge \Phi\left(-\frac{\sqrt{5}(1+\sqrt{n}(1-2p))\,|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle|}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right) \tag{B.2}$$

We next use a classical lower bound on Φ. Let $\Phi^c(t) = 1-\Phi(t) = \Phi(-t)$. Using (Abramowitz & Stegun, 1968, Equation (7.1.13)), we have for any $t > 0$:

$$\Phi^c(t) > \sqrt{\frac{2}{\pi}}\left(\frac{e^{-\frac{t^2}{2}}}{t+\sqrt{t^2+4}}\right).$$

Since $t+\sqrt{t^2+4} < 2(t+1)$ for $t > 0$, we get:

$$\Phi^c(t) > \frac{1}{\sqrt{2\pi}}\left(\frac{e^{-\frac{t^2}{2}}}{t+1}\right) > \frac{1}{2\sqrt{2\pi}}\exp(-t^2), \tag{B.3}$$

where we used that $t+1 \le t^2+2 \le 2e^{t^2/2}$. Combining (B.3) and (B.2) we get

$$\mathbb{P}(\mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle) \neq y) \ge \frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{5(1+\sqrt{n}(1-2p))^2\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right),$$

which completes the derivation of $\alpha_0(\boldsymbol{x})$.

B.2 Lower bound $\widetilde{\alpha}_0(\mathbf{x})$

We continue from (B.1), which reads

$$\mathbb{P}(\mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_0\rangle) \neq y) = \mathbb{E}\left[\Phi\left(-\frac{(\sum_{i\in[n]}\xi_i(1+u_i))\langle y\boldsymbol{x}, \boldsymbol{\mu}\rangle}{\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)\right], \tag{B.4}$$

where the expectation is with respect to the randomness in training data, namely u_i and ξ_0 for $i \in [n]$, while the test point $(\boldsymbol{x}, y)$ is fixed.

Note that under our data model, $\langle y\boldsymbol{x}, \boldsymbol{\mu}\rangle = (1+u)\left\|\boldsymbol{\mu}\right\|_{\ell_2}^2 = (1+u)\gamma^2 > 0$ since u is a non-negative random variables. We next the following event:

$$\mathcal{E}_0 := \left\{\sum_{i\in[n]}\xi_i(1+u_i) \ge n(1-2p)/2\right\}. \tag{B.5}$$

Note that by Hoeffding-type inequality for sub-gaussian random variables,

$$\mathbb{P}\left(\left|\sum_i \xi_i(1+u_i) - n(1-2p)(1+\mathbb{E}[u_i])\right| \ge t\right) \le e\exp\left(-\frac{ct^2}{n}\right),$$

for an absolute constant $c > 0$. (Note that $\|\xi_i(1+u_i)\|_{\psi_2} \le 1 + \|u_i\|_{\psi_2} = 2$ which is absorbed in the constant c.) By choosing $t = n(1-2p)/2$, and using the fact that $u_i > 0$ for all i we get

$$\mathbb{P}\left(\sum_i \xi_i(1+u_i) < n(1-2p)/2\right) \le e\exp\left(-cn(1-2p)^2\right).$$

Hence, $\mathbb{P}(\mathcal{E}_0^c) \le e\exp\left(-cn(1-2p)^2\right)$.

We then have

$$\begin{aligned}
\mathbb{E}\left[\Phi\left(-\frac{(\sum_{i\in[n]}\xi_i(1+u_i))\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)\right] &= \mathbb{E}\left[\Phi\left(-\frac{(\sum_{i\in[n]}\xi_i(1+u_i))\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)(\mathbb{1}_{\mathcal{E}_0}+\mathbb{1}_{\mathcal{E}_0^c})\right] \\
&\le \mathbb{E}\left[\Phi\left(-\frac{n(1-2p)\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{2\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)\mathbb{1}_{\mathcal{E}_0}\right] + \mathbb{P}(\mathcal{E}_0^c) \\
&= \Phi\left(-\frac{n(1-2p)\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{2\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)\mathbb{P}(\mathcal{E}_0) + \mathbb{P}(\mathcal{E}_0^c) \\
&\le \Phi\left(-\frac{n(1-2p)\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{2\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right) + \mathbb{P}(\mathcal{E}_0^c),
\end{aligned}$$

where in the first inequality we used the observation that $\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle \ge 0$ as explained above, the definition of $\mathcal{E}_0$ and that $\Phi(-z)$ is a decreasing function in z.

We next recall the tail bound of normal distribution (Abramowitz & Stegun, 1968, Equation (7.1.13)):

$$\Phi^c(t) \le \sqrt{\frac{2}{\pi}}\frac{e^{-t^2/2}}{t+\sqrt{t^2+\frac{8}{\pi}}} \le \frac{1}{2}e^{-t^2/2},$$

for all $t \ge 0$, where $\Phi^c(t) = 1-\Phi(t)$ is the complementary CDF of normal variables. Using this bound and the bound we derived on $\mathbb{P}(\mathcal{E}_0^c)$ we get

$$\mathbb{E}\left[\Phi\left(-\frac{(\sum_{i\in[n]}\xi_i(1+u_i))\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle}{\sqrt{n}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}}\right)\right] \le \frac{1}{2}\exp\left(-\frac{n(1-2p)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{4\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) + e\exp\left(-cn(1-2p)^2\right).$$

By invoking (B.4) we obtain

$$\mathbb{P}(\mathrm{sign}(\langle\boldsymbol{x},\widehat{\boldsymbol{\theta}}_0\rangle) = y) \ge 1 - \frac{1}{2}\exp\left(-\frac{n(1-2p)^2\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{4\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) - e\exp\left(-cn(1-2p)^2\right),$$

which completed the derivation of $\widetilde{\alpha}_0(\boldsymbol{x})$.

C Proof of Theorem 4.2

Using Theorem 4.1 we have:

$$\begin{aligned}
\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) &\le \mathbb{E}[\alpha_0(\boldsymbol{x})] \\
&= 1 - \frac{1}{2\sqrt{2\pi}}\mathbb{E}\left[\exp\left(-\frac{5(1+\sqrt{n}(1-2p))^2\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] \\
&\le 1 - \frac{1}{2\sqrt{2\pi}}\mathbb{E}\left[\exp\left(-\frac{5(1+\sqrt{n}(1-2p))^2\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\middle|\frac{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2} \le v^2\right]\mathbb{P}\left(\frac{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2} \le v^2\right) \\
&\le 1 - \frac{1}{2\sqrt{2\pi}}\exp\left(-5(1+\sqrt{n}(1-2p))^2v^2\right)\mathbb{P}\left(\frac{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2} \le v^2\right), \qquad \text{(C.1)}
\end{aligned}$$

for any value of $v > 0$. We next lower bound the probability on the right-hand side.

We write

$$\mathbb{P}\left(\frac{|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle|}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}} \le v\right) = \mathbb{P}\left(\frac{|y(1+u)\gamma^2|}{\|\boldsymbol{\Sigma}\boldsymbol{z}\|_{\ell_2}} \le v\right)$$
$$\ge \mathbb{P}\left(\frac{(1+u)\gamma^2}{\lambda_{\min}\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}} \le v\right)$$
$$= \mathbb{P}\left(\frac{(1+u)\gamma^2}{\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}} \le v\lambda_{\min}\right), \tag{C.2}$$

where the first step holds since under our data model we have $\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle = y(1+u)\left\|\boldsymbol{\mu}\right\|_{\ell_2}^2$ and $\boldsymbol{\Sigma}^{1/2}\boldsymbol{x} = y(1+u)\boldsymbol{\Sigma}^{1/2}\boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{z} = \boldsymbol{\Sigma}\boldsymbol{z}$, given that $\boldsymbol{\Sigma}^{1/2}\boldsymbol{\mu} = 0$. Continuing from (C.2) we write

$$\mathbb{P}\left(\frac{(1+u)\gamma^2}{\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}} \le v\lambda_{\min}\right) \ge \mathbb{P}\left((1+u)\gamma^2 \le v\lambda_{\min}\sqrt{d/2},\ \left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2} \ge \sqrt{d/2}\right)$$
$$= \mathbb{P}\left((1+u)\gamma^2 \le v\lambda_{\min}\sqrt{d/2}\right)\mathbb{P}\left(\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2} \ge \sqrt{d/2}\right), \tag{C.3}$$

given that $\boldsymbol{z}$ and u are independent. Next note that $\mathcal{P}_{\boldsymbol{\mu}}^{\perp}(\boldsymbol{z})$ is distributed as a Gaussian vector in a $(d-1)$-dimensional space, and therefore $\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}^2$ is χ^2 distribution with $(d-1)$-degree of freedom. We next use the following result about the tail bound of χ^2 distribution to control $\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}$.

Lemma C.1 *(Ghosh, 2021, Theorem 4) Suppose $X \sim \chi_d^2(\lambda)$. Then for $0 < \eta < d + \lambda$, we have*

$$\mathbb{P}(X < d + \lambda - \eta) \le \exp\left[-\frac{d\eta^2}{4(d+2\lambda)^2}\right].$$

As a corollary of Lemma C.1, by choosing $\eta = \lambda + d/2$, we obtain $\mathbb{P}(X < d/2) \le e^{-d/16}$ for any $\lambda \ge 0$. Using this we have

$$\mathbb{P}\left(\left\|\mathcal{P}_{\boldsymbol{\mu}}^{\perp}\boldsymbol{z}\right\|_{\ell_2}^2 < d/2\right) \le e^{-d/16}. \tag{C.4}$$

We next lower bound the other term on the right-hand side of (C.2). By Markov's inequality for any non-negative random variable X, we have $\mathbb{P}(X \le 2\,\mathbb{E}[X]) \ge 1/2$. Also $\mathbb{E}[(1+u)\gamma^2] \le 2\gamma^2$, since $\mathbb{E}[u] \le 1$ and therefore, by choosing $v = \frac{4\sqrt{2}\gamma^2}{\lambda_{\min}\sqrt{d}}$, we get

$$\mathbb{P}\left((1+u)\gamma^2 \le u\lambda_{\min}\sqrt{d/2}\right) \ge 1/2. \tag{C.5}$$

Combining (C.4), (C.5) and (C.2) with the bound (C.1) we arrive at

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) \le 1 - \frac{1}{4\sqrt{2\pi}}\exp\left(-160(1+\sqrt{n}(1-2p))^2\frac{\gamma^4}{\lambda_{\min}^2 d}\right)\left(1 - e^{-d/16}\right).$$

To derive the lower bound on $\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0)$ note that by using Theorem 4.1 we have

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) \ge \mathbb{E}[\widetilde{\alpha}_0(\boldsymbol{x})]$$
$$= 1 - \frac{1}{2}\,\mathbb{E}\left[\exp\left(-\frac{n(1-2p)^2\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{4\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] - e\exp\left(-cn(1-2p)^2\right).$$

Under our data model, $\boldsymbol{x} = y(1+u)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}$. Consider the probabilistic event $\mathcal{E}_2 := \{\boldsymbol{z} : \|\boldsymbol{z}\|_{\ell_2}^2 \le 2d\}$. For every $\eta > 0$, we have

$$\mathbb{E}\left[\exp\left(-\frac{\eta\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] = \mathbb{E}\left[\exp\left(-\frac{\eta\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)(\mathbb{1}_{\mathcal{E}_2} + \mathbb{1}_{\mathcal{E}_2^c})\right]$$
$$\le \exp\left(-\frac{\eta\gamma^4}{2\lambda_{\max} d}\right) + \mathbb{P}(\mathcal{E}_2^c)\,,$$

where the inequality holds because $\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle = y(1+u)\gamma^2$ and so $|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| \ge \gamma^2$ (given that u is a non-negative random variable). In addition,

$$\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2 = \|\boldsymbol{\Sigma}\boldsymbol{z}\|_{\ell_2}^2 \le \lambda_{\max}^2 \|\boldsymbol{z}\|_{\ell_2}^2 \le 2\lambda_{\max}^2 d$$

on the event $\mathcal{E}_2$. Further, $\|\boldsymbol{z}\|_{\ell_2}^2 \sim \chi_d^2$ and so using tail bounds for χ_d^2, see e.g., (Ghosh, 2021, Theorem 3), we have $\mathbb{P}(\mathcal{E}_2^c) \le e^{-d/8}$. Putting things together, we obtain

$$\mathbb{E}\left[\exp\left(-\frac{\eta\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle^2}{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] \le \exp\left(-\frac{\eta\gamma^4}{2\lambda_{\max}^2 d}\right) + e^{-d/8}\,.$$

Setting $\eta = n(1-2p)^2/4$, this implies that

$$\mathsf{acc}(\widehat{\boldsymbol{\theta}}_0) \ge 1 - \frac{1}{2}\exp\left(-\frac{n(1-2p)^2\gamma^4}{8\lambda_{\max}^2 d}\right) - e^{-d/8} - e\exp\left(-cn(1-2p)^2\right)\,.$$

D Proof of Theorem 4.6

Proof Since we are interested with dependence of sample size with respect to d and p, we assume $\gamma := \|\boldsymbol{\mu}\|_{\ell_2} = 1$ and $\boldsymbol{\Sigma} = \boldsymbol{I} - \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}$ the projection onto the orthogonal space of $\boldsymbol{\mu}$. Also note that without loss of generality, we can assume that our estimator $\widehat{\boldsymbol{\theta}}$ has unit norm, because its norm does not affect the sign of $\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}\rangle$. This way, $\boldsymbol{\mu}$ and $\widehat{\boldsymbol{\theta}}$ both belong to $\mathbb{S}^{d-1}$.

We first follow a standard argument to "reduce" the classification problem to a multi-way hypothesis testing problem. Let $\rho \in (0,1)$ be an arbitrary but fixed value which we can choose. We define a ρ-packing of $\mathbb{S}^{d-1}$ as a set $\mathcal{M} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M\} \subset \mathbb{S}^{d-1}$ such that $\langle \boldsymbol{\mu}_l, \boldsymbol{\mu}_k\rangle \le \rho$ for $l \ne k$. Also define the ρ-packing number of $\mathbb{S}^{d-1}$ as

$$M(\rho, \mathbb{S}^{d-1}) := \sup\{M \in \mathbb{N} : \text{there exists a } \rho\text{-packing } \mathcal{M} \text{ of } \mathbb{S}^{d-1} \text{ with size } M\}\,. \tag{D.1}$$

For convenience, we define the misclassification error of a classifier $\widehat{\boldsymbol{\theta}} \in \mathbb{S}^{d-1}$ as $\mathrm{err}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu}) = 1 - \mathrm{acc}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu})$. By our assumption,

$$\delta \ge \sup_{\boldsymbol{\mu}\in\mathbb{S}^{d-1}} \mathrm{err}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu}) \ge \sup_{\boldsymbol{\mu}\in\mathcal{M}} \mathrm{err}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu})\,. \tag{D.2}$$

In order to further lower bound the right hand side, we let I be a random variable uniformly distributed on the hypothesis set $\{1, 2, \dots, M\}$ and consider the case of $\boldsymbol{\mu} = \boldsymbol{\mu}_I$. We also define $\widehat{I}$ as the index of the element in $\mathcal{M}$ with maximum inner product with $\widehat{\boldsymbol{\theta}}$ (it does not matter how we break ties). Under our data model we have $\langle \boldsymbol{x}, \boldsymbol{\mu}_I\rangle = y(1+u)\|\boldsymbol{\mu}_I\|_{\ell_2}^2 = y(1+u)$ and since u is a non-negative random variable, $y = \mathrm{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I\rangle)$. We then have

$$\sup_{\boldsymbol{\mu}\in\mathcal{M}} \mathrm{err}(\widehat{\boldsymbol{\theta}}; \boldsymbol{\mu}) \ge \max_{i\in[M]} \mathbb{P}\left(\mathrm{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I\rangle) \ne \mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}\rangle)\middle| I = i\right)$$
$$\ge \frac{1}{M}\sum_{i=1}^{M} \mathbb{P}\left(\mathrm{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I\rangle) \ne \mathrm{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}\rangle)\middle| I = i\right)$$
$$\ge \Phi\left(-2\sqrt{\frac{1+\rho}{1-\rho}}\right)\frac{1}{M}\sum_{i=1}^{M} \mathbb{P}(\widehat{I} \ne i | I = i) \tag{D.3}$$
$$= \Phi\left(-2\sqrt{\frac{1+\rho}{1-\rho}}\right)\mathbb{P}(\widehat{I} \ne I)\,, \tag{D.4}$$

with Φ denoting the CDF of a standard normal variable. Equation (D.3) above follows from the lemma below.

Lemma D.1 *For any $i \in [M]$, we have*

$$\mathbb{P}\left(\operatorname{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I \rangle) \neq \operatorname{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}} \rangle) \middle| I = i\right) \geq \Phi\left(-2\sqrt{\frac{1+\rho}{1-\rho}} \right) \mathbb{P}(\widehat{I} \neq i | I = i) \,.$$

Combining (D.2) and (D.4), we obtain that

$$\delta_0 := \frac{\delta}{\Phi\left(-2\sqrt{\frac{1+\rho}{1-\rho}} \right)} \geq \mathbb{P}(\widehat{I} \neq I) \,. \tag{D.5}$$

Next recall the set $\mathcal{T} := \{(\boldsymbol{x}_j, \widehat{y}_j)\}_{j \in [n]}$, and let $\boldsymbol{X} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T]^T \in \mathbb{R}^{n \times d}$ and $\widehat{\boldsymbol{y}} = [\widehat{y}_1, \ldots, \widehat{y}_n]^T$. By an application of Fano's inequality, with conditioning on $\mathbf{X}$ (see e.g. (Scarlett & Cevher, 2019, Section 2.3)) we have

$$\mathcal{I}(I; \widehat{I} | \boldsymbol{X}) \geq (1 - \delta_0) \log(M(\rho, \mathbb{S}^{d-1})) - \log 2 \,, \tag{D.6}$$

where $\mathcal{I}(I; \widehat{I} | \boldsymbol{X})$ represents the conditional mutual information between I and $\widehat{I}$. Using the fact that $I \to \widehat{\boldsymbol{y}} \to \widehat{I}$ forms a Markov chain conditioned on $\mathbf{X}$, and by an application of the data processing inequality we have:

$$\mathcal{I}(I; \widehat{I} | \boldsymbol{X}) \leq \mathcal{I}(I; \widehat{\boldsymbol{y}} | \boldsymbol{X}) \tag{D.7}$$

We will now upper bound $\mathcal{I}(I; \widehat{\boldsymbol{y}} | \boldsymbol{X})$. Let $w_j := \frac{1}{2}\big(\operatorname{sign}(\langle \boldsymbol{x}_j, \boldsymbol{\mu}_I \rangle) + 1\big)$ be the 0-1 version of the actual label of $\boldsymbol{x}_j$, viz., $\operatorname{sign}(\langle \boldsymbol{x}_j, \boldsymbol{\mu}_I \rangle)$. As per our setting, we have $\widehat{y}_j = 2\big(w_j \oplus z_j\big) - 1$ where $z_j \sim \text{Bernoulli}(p)$ and $\oplus$ denotes modulo-2 addition. Since the noise variables z_j are independent and $\widehat{y}_j$ depends on $(I, \boldsymbol{X})$ only through $w_j = \frac{1}{2}\big(\operatorname{sign}(\langle \boldsymbol{x}_j, \boldsymbol{\mu}_I \rangle) + 1\big)$, by using the tensorization property of the mutual information (see e.g. (Scarlett & Cevher, 2019, Lemma 2, part (iii))), we have

$$\mathcal{I}(I; \boldsymbol{y} | \boldsymbol{X}) \leq \sum_{j=1}^{n} \mathcal{I}\big(w_j; \widehat{y}_j\big) \leq n(\log 2 - H_2(p)) \,, \tag{D.8}$$

where the second inequality follows since $\widehat{y}_j$ is generated by passing w_j through a binary symmetric channel, which has capacity $\log 2 - H_2(p)$ with $H_2(p) := -p \log p - (1-p) \log(1-p)$ denoting the binary entropy function.

We next use the lemma below to further upper bound the right-hand side of (D.8).

Lemma D.2 *For a discrete probability distribution, consider the entropy function given by*

$$H(p_1, \ldots, p_k) = \sum_{i=1}^{k} p_k \log(1/p_k) \,.$$

We have the following bound:

$$H(p_1, \ldots, p_k) \geq \log k - k \sum_{i=1}^{k} (p_i - 1/k)^2 \,.$$

Using Lemma D.2 with $k = 2$ we obtain $\log 2 - H_2(p) \leq 4(p - 1/2)^2 = (1 - 2p)^2$, which along with (D.8), (D.7) and (D.6) gives

$$\frac{n(1-2p)^2 + \log 2}{1 - \delta_0} \geq \log(M(\rho, \mathbb{S}^{d-1})) \,. \tag{D.9}$$

In our next lemma, we lower bound $M(\rho, \mathbb{S}^{d-1})$.

Lemma D.3 *Recall the definition of ρ-packing number of $\mathbb{S}^{d-1}$ given by (D.1). We have the following bound:*

$$M(\rho, \mathbb{S}^{d-1}) \geq \exp\left(\frac{d\rho^2}{2}\right) .$$

Using Lemma D.3 along with (D.9), we obtain the following lower bound on the sample complexity:

$$n \geq \frac{\frac{\rho^2}{2}(1-\delta_0)d - \log 2}{(1-2p)^2}, \text{ with } \delta_0 = \frac{\delta}{\Phi\Big(-2\sqrt{\frac{1+\rho}{1-\rho}}\Big)}.$$

Note that $\rho \in (0,1)$ can be set arbitrarily. Since our claim is on the order of sample size, the specific value of ρ does not matter. ■

We now prove Lemmas D.1, D.2 and D.3.

Proof of Lemma D.1. Let us consider the event $\widehat{I} \neq i$, given that $I = i$. We note that if $\widehat{I} \neq I$, then $\langle \widehat{\boldsymbol{\theta}}, \boldsymbol{\mu}_I \rangle \leq \sqrt{\frac{1+\rho}{2}}$ or equivalently, the angle between $\widehat{\boldsymbol{\theta}}$ and $\boldsymbol{\mu}_I$ is $\geq b := \cos^{-1}\left(\sqrt{\frac{1+\rho}{2}}\right)$. Otherwise, by definition of $\widehat{I}$ we have $\langle \widehat{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\widehat{I}} \rangle \geq \langle \widehat{\boldsymbol{\theta}}, \boldsymbol{\mu}_I \rangle > \sqrt{\frac{1+\rho}{2}}$, and therefore the angle between $\boldsymbol{\mu}_{\widehat{I}}$ and $\boldsymbol{\mu}_I$ is $< 2b$. Noting that $\cos(2b) = 2\cos^2(b) - 1 = \rho$, we would then have $\langle \boldsymbol{\mu}_{\widehat{I}}, \boldsymbol{\mu}_I \rangle > \rho$, which is a contradiction since $\mathcal{M}$ forms a ρ-packing. We proceed by writing

$$\mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I \rangle) \neq \text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}} \rangle) \Big| I = i\right) \geq \mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I \rangle) \neq \text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}} \rangle) \Big| I = i, \widehat{I} \neq i\right) \mathbb{P}(\widehat{I} \neq i | I = i)\,.$$

As discussed above, on the event that $I = i, \widehat{I} \neq i$, we have $\theta := \langle \widehat{\boldsymbol{\theta}}, \boldsymbol{\mu}_i \rangle \leq \sqrt{\frac{1+\rho}{2}}$. Consider the decomposition $\boldsymbol{x} = \langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i + \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \boldsymbol{x}$. We then have

$$\mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I \rangle) \neq \text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}} \rangle) \Big| I = i, \widehat{I} \neq i\right) = \mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle) \left(\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle \theta + \langle \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \boldsymbol{x}, \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \widehat{\boldsymbol{\theta}} \rangle\right) \leq 0 \Big| I = i, \widehat{I} \neq i\right).$$

Note that $\boldsymbol{x}$ is a test data point, independent of the training data $\mathcal{T}$ and so it is independent of $\widehat{\boldsymbol{\theta}}$. In addition, under our data model, $\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle$ is independent of $\mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \boldsymbol{x}$. Hence, $\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle) \langle \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \boldsymbol{x}, \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \widehat{\boldsymbol{\theta}} \rangle \sim \mathsf{N}(0, \|\mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \widehat{\boldsymbol{\theta}}\|^2_{\ell_2})$. Given that $\|\widehat{\boldsymbol{\theta}}\|_{\ell_2} = 1$ we also have $\|\mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \widehat{\boldsymbol{\theta}}\|^2_{\ell_2} = 1 - \langle \widehat{\boldsymbol{\theta}}, \boldsymbol{\mu}_i \rangle^2 = 1 - \theta^2$. In short, we can write

$$\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle) \langle \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \boldsymbol{x}, \mathcal{P}^{\perp}_{\boldsymbol{\mu}_i} \widehat{\boldsymbol{\theta}} \rangle = \sqrt{1-\theta^2} Z, \quad Z \sim \mathsf{N}(0,1)\,.$$

Using this characterization, we proceed by writing

$$\begin{aligned}
\mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_I \rangle) \neq \text{sign}(\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}} \rangle) \Big| I = i, \widehat{I} \neq i\right) &= \mathbb{P}\left(\text{sign}(\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle) \langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle \theta + \sqrt{1-\theta^2} Z \leq 0\right) \\
&= \mathbb{P}\left(Z \leq -\frac{|\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle| \theta}{\sqrt{1-\theta^2}}\right) \\
&= 1 - \mathbb{E}\left[\Phi\Big(\frac{\theta}{\sqrt{1-\theta^2}} |\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle|\Big)\right] \\
&\overset{(a)}{\geq} 1 - \mathbb{E}\left[\Phi\left(\sqrt{\frac{1+\rho}{1-\rho}} |\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle|\right)\right] \\
&\overset{(b)}{\geq} 1 - \Phi\left(\sqrt{\frac{1+\rho}{1-\rho}} \mathbb{E}\big[|\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle|\big]\right) \\
&\overset{(c)}{\geq} 1 - \Phi\left(2\sqrt{\frac{1+\rho}{1-\rho}}\right) = \Phi\left(-2\sqrt{\frac{1+\rho}{1-\rho}}\right),
\end{aligned}$$

where (a) follows from the fact that $\theta \leq \sqrt{\frac{1+\rho}{2}}$; (b) holds due to Jensen's inequality and concavity of $\Phi(\cdot)$ on the positive values, and (c) holds because under our data model $\mathbb{E}[|\langle \boldsymbol{x}, \boldsymbol{\mu}_i \rangle|] = \mathbb{E}[y(1+u)\|\boldsymbol{\mu}_i\|^2_{\ell_2}] = \mathbb{E}[1+u] \leq 2$ since $\|u\|_{\psi_2} = 1$. This completes the proof of claim.

Proof of Lemma D.2. Define $q_i = p_i - 1/k$. Note that q_i can be negative, and we have $\sum_{i=1}^k q_i = 0$. We write

$$\begin{aligned} H(p_1, \dots, p_k) &= -\sum_{i=1}^k p_i \log p_i \\ &= -\sum_{i=1}^k (1/k + q_i) \log(1/k + q_i) \\ &= -\sum_{i=1}^k (1/k + q_i)[\log(1/k) + \log(1 + kq_i)] \\ &\geq \log k - \sum_{i=1}^k (1/k + q_i) k q_i \qquad \text{(D.10)} \\ &= \log k - k \sum_{i=1}^k q_i^2 \\ &= \log k - k \sum_{i=1}^k (p_i - 1/k)^2 . \end{aligned}$$

Note that in eq. (D.10) we used the fact that $1 + kq_i \geq 0$ and $\log x \leq x - 1$ for all $x \geq 0$.

This completes the proof of the lemma.

Proof of Lemma D.3. Define a ρ-cover of $\mathbb{S}^{d-1}$ as a set of $\mathcal{V} := \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$ such that for any $\boldsymbol{\theta} \in \mathbb{S}^{d-1}$, there exists some $\boldsymbol{v}_i$ such that $\langle \boldsymbol{\theta}, \boldsymbol{v}_i \rangle \geq \rho$. The ρ-covering number of $\mathbb{S}^{d-1}$ is

$$N(\rho, \mathbb{S}^{d-1}) := \inf\{N \in \mathbb{N} : \text{there exists a } \rho\text{-cover } \mathcal{V} \text{ of } \mathbb{S}^{d-1} \text{ with size } N\} .$$

By a simple argument we have $M(\rho, \mathbb{S}^{d-1}) \geq N(\rho, \mathbb{S}^{d-1})$. Concretely, we construct a ρ-packing greedily by adding an element at each step which has inner product at most ρ with all the previously selected elements, until it is no longer possible. This means that any point on $\mathbb{S}^{d-1}$ has inner product larger than ρ by some of the elements in the constructed set (otherwise it contradicts its maximality). Hence, we have a set that is both a ρ-cover and a ρ-packing of $\mathbb{S}^{d-1}$, and by definition it results in $M(\rho, \mathbb{S}^{d-1}) \geq N(\rho, \mathbb{S}^{d-1})$.

We next lower bound $N(\rho, \mathbb{S}^{d-1})$ via a volumetric argument. Let $\mathcal{V} := \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$ be a ρ-cover of $\mathbb{S}^{d-1}$. For each element $\boldsymbol{v}_i \in \mathcal{V}$ we consider the cone around it with apex angle $\cos^{-1}(\rho)$. Its intersection with $\mathbb{S}^{d-1}$ defines a spherical cap which we denote by $\mathcal{C}(\boldsymbol{v}_i, \rho)$. Since $\mathcal{V}$ forms a ρ-cover of $\mathbb{S}^{d-1}$, we have

$$\text{Vol}(\mathbb{S}^{d-1}) \leq \text{Vol}(\cup_{i=1}^N \mathcal{C}(\boldsymbol{v}_i, \rho)) \leq \sum_{i=1}^N \text{Vol}(\mathcal{C}(\boldsymbol{v}_i, \rho)) .$$

We next use Lemma 2.2 from Ball et al. (1997) by which we have $\frac{\mathcal{C}(\boldsymbol{v}_i, \rho)}{\text{Vol}(\mathbb{S}^{d-1})} \leq e^{-d\rho^2/2}$. Using this above, we get

$$1 \leq N e^{-d\rho^2/2}$$

for any ρ-cover $\mathcal{V}$. Thus, we have $N(\rho, \mathbb{S}^{d-1}) \geq \exp(\frac{d\rho^2}{2})$, which completes the proof of the lemma.

E Proof of Theorem 4.8

We want to upper bound $\mathbb{P}(y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1 \rangle < 0)$. Plugging in for $\widehat{\boldsymbol{\theta}}_1$ from (4.7) we have

$$y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1 \rangle = \frac{y}{n} \langle \boldsymbol{x}, \sum_i \widetilde{y}_i \boldsymbol{x}_i \rangle = \frac{y}{n} \langle \boldsymbol{x}, \sum_i \text{sign}(\langle \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}_0 \rangle) \boldsymbol{x}_i \rangle$$

A major complication is that $\widehat{\boldsymbol{\theta}}_0$ depends on all of the data points in the training set. Expanding $\widehat{\boldsymbol{\theta}}_0$ we have

$$\widehat{\boldsymbol{\theta}}_0 = \frac{1}{n} \sum_i \widehat{y}_i \boldsymbol{x}_i = \frac{1}{n} \sum_i \xi_i y_i \boldsymbol{x}_i = \frac{1}{n} \left((\sum_i \xi_i (1 + u_i)) \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2} \sum_i \xi_i y_i \boldsymbol{z}_i \right) \qquad \text{(E.1)}$$

Here is the outline of our proof:

- For every $\ell \in [n]$ we define the label

$$\widetilde{y}'_\ell = \operatorname{sign}\left(\langle \boldsymbol{x}_\ell, (\sum_i \xi_i(1+u_i))\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell \boldsymbol{z}_\ell\rangle\right). \tag{E.2}$$

- We define the event $\mathcal{E} := \{\forall \ell : \widetilde{y}'_\ell = \operatorname{sign}(\langle \boldsymbol{x}_\ell, \widehat{\boldsymbol{\theta}}_0\rangle)\}$.
- We write

$$\begin{aligned}\mathbb{P}(y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle < 0) &= \mathbb{P}(y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle < 0; \mathcal{E}) + \mathbb{P}(y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle < 0; \mathcal{E}^c) \\ &\le \mathbb{P}(y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle < 0; \mathcal{E}) + \mathbb{P}(\mathcal{E}^c)\,. \end{aligned} \tag{E.3}$$

- We bound each of the term on the right-hand side separately. Note that on the event $\mathcal{E}$, we have

$$y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle = \frac{y}{n}\langle \boldsymbol{x}, \sum_\ell \widetilde{y}'_\ell \boldsymbol{x}_\ell\rangle$$

  We will control the probability of this quantity being negative using sum of i.i.d subgaussian random variables.

Next we get into details of this proof sketch. We start by bounding $\mathbb{P}(\mathcal{E}^c)$.

Lemma E.1 *Fix $\ell \in [n]$. Suppose that $nd \ge \frac{\gamma^4}{\lambda_{\max}^2}$ and define $p' := (1 + \frac{3\gamma^4}{8\lambda_{\max}^2 nd})p$. We then have*

$$\mathbb{P}\left(\widetilde{y}'_\ell \langle \boldsymbol{x}_\ell, \widehat{\boldsymbol{\theta}}_0\rangle < 0\right) \le \frac{1}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/16}\right)e^{d/n}\,.$$

Using Lemma E.1 along with union bounding over $\ell \in [n]$ (note that the events are dependent), we get

$$\mathbb{P}(\mathcal{E}^c) \le \frac{n}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/16}\right)e^{d/n}\,. \tag{E.4}$$

Next we note that on the event $\mathcal{E}$ we have

$$y\langle \boldsymbol{x}, \widehat{\boldsymbol{\theta}}_1\rangle = \frac{y}{n}\langle \boldsymbol{x}, \sum_\ell \widetilde{y}'_\ell \boldsymbol{x}_\ell\rangle = \frac{1}{n}\sum_{\ell\in[n]} y\operatorname{sign}\left(\langle \boldsymbol{x}_\ell, (\sum_i \xi_i(1+u_i))\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell \boldsymbol{z}_\ell\rangle\right)\langle \boldsymbol{x}, \boldsymbol{x}_\ell\rangle \tag{E.5}$$

We define the shorthand $\beta := \sum_i \xi_i(1+u_i)$ and condition on β. Then (E.5) will be sum of iid subgaussian variables

$$T_\ell := y\operatorname{sign}\left(\langle \boldsymbol{x}_\ell, \beta\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell \boldsymbol{z}_\ell\rangle\right)\langle \boldsymbol{x}, \boldsymbol{x}_\ell\rangle\,, \quad \ell \in [n]\,. \tag{E.6}$$

We continue by first characterizing its expectation.

Lemma E.2 *For $\ell \in [n]$ we have*

$$\mathbb{E}[T_\ell|\beta] \ge (1-2q)\langle y\boldsymbol{x}, \boldsymbol{\mu}\rangle > 0\,,$$

where

$$q = \exp\left(-\frac{\beta\gamma^2}{20\lambda_{\max}}\right) + \mathbb{1}\left(\frac{\beta\gamma^2}{2\lambda_{\max} d} < 1\right).$$

Recall that $\boldsymbol{x}_\ell = y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell$ and therefore

$$\begin{aligned}\|T_\ell - \mathbb{E}[T_\ell|\beta]\|_{\psi_2} \le 2\|T_\ell\|_{\psi_2} &= 2\|\langle \boldsymbol{x}, \boldsymbol{x}_\ell\rangle\|_{\psi_2} \\ &= 2\|\langle \boldsymbol{x}, y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\rangle\|_{\psi_2} \\ &\le 4|\langle \boldsymbol{x}, \boldsymbol{\mu}\rangle| + 2\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}\,.\end{aligned}$$

where we used the assumption $\|u_\ell\|_{\psi_2} = 1$ and the fact that $\|\boldsymbol{z}\|_{\psi_2} = 1$.

Next by using Hoeffding-type inequality for sum of sub-gaussian random variables (see e.g (Vershynin, 2010, Proposition 5.1)) we have for every $t \geq 0$,

$$\mathbb{P}\left(\Big|\sum_\ell T_\ell - \sum_\ell \mathbb{E}[T_\ell|\beta]\Big| \geq t \Big| \beta\right) \leq e\exp\left(-\frac{ct^2}{n(\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2)}\right),$$

where $c > 0$ is an absolute constant. Therefore,

$$\begin{aligned}\mathbb{P}(\sum_\ell T_\ell < 0\,|\beta) &\leq \mathbb{P}\left(|\sum_\ell T_\ell - \sum_\ell \mathbb{E}[T_\ell|\beta]| \geq \sum_\ell \mathbb{E}[T_\ell|\beta] \;\Big|\; \beta\right)\\ &\leq e\exp\left(-\frac{cn(\mathbb{E}[T_\ell|\beta])^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\\ &\leq e\exp\left(-\frac{cn(1-2q)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right), \end{aligned} \tag{E.7}$$

where the last step follows from Lemma E.2.

Our next step is to take expectation of the above with respect to β. Before proceeding we define the following event:

$$\mathcal{E}_0 := \{\beta \geq n(1-2p)/2\}\,. \tag{E.8}$$

Note that by Hoeffding-type inequality for sub-gaussian random variables,

$$\mathbb{P}\left(\Big|\sum_i \xi_i(1+u_i) - n(1-2p)(1+\mathbb{E}[u_i])\Big| \geq t\right) \leq e\exp\left(-\frac{ct^2}{n}\right),$$

for an absolute constant $c > 0$. (Note that $\|\xi_i(1+u_i)\|_{\psi_2} \leq 1 + \|u_i\|_{\psi_2} = 2$ which is absorbed in the constant c.) By choosing $t = n(1-2p)/2$, and using the fact that $u_i > 0$ for all i we get

$$\mathbb{P}\left(\sum_i \xi_i(1+u_i) < n(1-2p)/2\right) \leq e\exp\left(-cn(1-2p)^2\right).$$

Hence, $\mathbb{P}(\mathcal{E}_0^c) \leq e\exp\left(-cn(1-2p)^2\right)$. We now continue by taking expectation of (E.7) with respect to β.

$$\begin{aligned}\mathbb{P}(\sum_\ell T_\ell < 0) &\leq e\,\mathbb{E}\left[\exp\left(-\frac{cn(1-2q)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right]\\ &= e\,\mathbb{E}\left[\exp\left(-\frac{cn(1-2q)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)(\mathbb{1}_{\mathcal{E}_0} + \mathbb{1}_{\mathcal{E}_0^c})\right]\\ &\leq \mathbb{E}\left[\exp\left(-\frac{cn(1-2q)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\mathbb{1}_{\mathcal{E}_0}\right] + \mathbb{P}(\mathcal{E}_0^c)\,. \end{aligned} \tag{E.9}$$

On the event $\mathcal{E}_0$ we have

$$\frac{\beta\gamma^2}{2\lambda_{\max}d} \geq \frac{n(1-2p)\gamma^2}{4\lambda_{\max}d} > 1,$$

by our assumption. Therefore,

$$q = \exp\left(-\frac{\beta\gamma^2}{20\lambda_{\max}}\right) \leq \exp\left(-\frac{n(1-2p)\gamma^2}{40\lambda_{\max}}\right) := q'\,.$$

Since $1-2q \geq 1-2q' > 0$ this implies that

$$\mathbb{E}\left[\exp\left(-\frac{cn(1-2q)^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\mathbb{1}_{\mathcal{E}_0}\right] \leq \mathbb{E}\left[\exp\left(-\frac{cn(1-2q')^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\mathbb{1}_{\mathcal{E}_0}\right]$$
$$\leq \exp\left(-\frac{cn(1-2q')^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right).$$

Using the above bound along with the bound on $\mathbb{P}(\mathcal{E}_0^c)$ into (E.9), we arrive at

$$\mathbb{P}(\sum_\ell T_\ell < 0) \leq \exp\left(-\frac{cn(1-2q')^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) + e\exp\left(-cn(1-2p)^2\right). \tag{E.10}$$

Invoking (E.5) and the definition of T_ℓ given by (E.6) we obtain

$$\mathbb{P}(y\langle \boldsymbol{x},\widehat{\boldsymbol{\theta}}_1\rangle < 0;\mathcal{E}) \leq \exp\left(-\frac{cn(1-2q')^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right) + e\exp\left(-cn(1-2p)^2\right). \tag{E.11}$$

For the final step we combine (E.11), (E.4) and (E.3) to get

$$\mathbb{P}(y\langle \boldsymbol{x},\widehat{\boldsymbol{\theta}}_1\rangle < 0) \leq \exp\left(-\frac{cn(1-2q')^2\langle y\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle \boldsymbol{x},\boldsymbol{\mu}\rangle^2+\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)$$
$$+ e\exp\left(-cn(1-2p)^2\right) + \frac{n}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right)+e^{-d/16}\right)e^{d/n}. \tag{E.12}$$

E.1 Proof of Lemma E.1

Define $\tilde{\xi}_i := \xi_i(1+u_i)$ By invoking (E.1) we have

$$\widetilde{y}'_\ell\langle \boldsymbol{x}_\ell,\widehat{\boldsymbol{\theta}}_0\rangle = \frac{\widetilde{y}'_\ell}{n}\langle \boldsymbol{x}_\ell,(\sum_i\tilde{\xi}_i))\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell\boldsymbol{z}_\ell+\boldsymbol{\Sigma}^{1/2}\sum_{i\neq\ell}\xi_i y_i\boldsymbol{z}_i\rangle$$

We have $\sum_{i\neq\ell}\xi_i y_i\boldsymbol{z}_i \sim \mathsf{N}(0,(n-1)\boldsymbol{I}_d)$ and so $\langle\widetilde{y}'_\ell\boldsymbol{x}_\ell,\boldsymbol{\Sigma}^{1/2}\sum_{i\neq\ell}\xi_i y_i\boldsymbol{z}_i\rangle$ is a zero mean Gaussian with variance $(n-1)\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}^2$. Let $\mathcal{F}_\ell$ be the σ-algebra generated by $(\boldsymbol{z}_\ell,y_\ell,\{\xi_i,u_i\}_{i\in[n]})$. Note that $\boldsymbol{z}_i$, for $i\neq\ell$, is independent of $\mathcal{F}_\ell$ and $\widetilde{y}'_\ell$ is $\mathcal{F}_\ell$-measurable. Hence,

$$\mathbb{P}\left(\widetilde{y}'_\ell\langle \boldsymbol{x}_\ell,\widehat{\boldsymbol{\theta}}_0\rangle<0\middle|\mathcal{F}_\ell\right) = \Phi\left(-\frac{\widetilde{y}_\ell\langle \boldsymbol{x}_\ell,(\sum_i\tilde{\xi}_i)\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell\boldsymbol{z}_\ell\rangle}{\sqrt{n-1}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}}\right)$$
$$\overset{(a)}{=} \Phi\left(-\frac{\left|\langle \boldsymbol{x}_\ell,(\sum_i\tilde{\xi}_i)\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell\boldsymbol{z}_\ell\rangle\right|}{\sqrt{n-1}\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}}\right)$$
$$\overset{(b)}{\leq} \frac{1}{2}\exp\left(-\frac{\left|\langle \boldsymbol{x}_\ell,(\sum_i\tilde{\xi}_i)\boldsymbol{\mu}+\boldsymbol{\Sigma}^{1/2}\xi_\ell y_\ell\boldsymbol{z}_\ell\rangle\right|^2}{2n\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}^2}\right)$$
$$\overset{(c)}{\leq} \frac{1}{2}\exp\left(-\frac{\langle \boldsymbol{x}_\ell,\boldsymbol{\mu}\rangle^2(\sum_i\tilde{\xi}_i)^2}{4n\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}^2}\right)\exp\left(\frac{\langle \boldsymbol{x}_\ell,\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\rangle^2}{2n\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}^2}\right)$$
$$\leq \frac{1}{2}\exp\left(-\frac{\langle \boldsymbol{x}_\ell,\boldsymbol{\mu}\rangle^2(\sum_i\tilde{\xi}_i)^2}{4n\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2}^2}\right)\exp\left(\frac{\|\boldsymbol{z}_\ell\|_{\ell_2}^2}{2n}\right). \tag{E.13}$$

Here, (a) follows by the choice of $\widetilde{y}'_\ell$; (b) holds by using (Abramowitz & Stegun, 1968, Equation (7.1.13)):

$$\Phi^c(t) \le \sqrt{\frac{2}{\pi}} \frac{e^{-t^2/2}}{t + \sqrt{t^2 + \frac{8}{\pi}}} \le \frac{1}{2} e^{-t^2/2},$$

for all $t \ge 0$. In addition (c) holds since $(a+b)^2 \ge \frac{a^2}{2} - b^2$.

We proceed by taking expectation of the right-hand side of (E.13) with respect to $\tilde{\xi}_1, \dots, \tilde{\xi}_n$. Define the shorthand $\lambda := \frac{\langle \boldsymbol{x}_\ell, \boldsymbol{\mu} \rangle^2}{4n \|\boldsymbol{\Sigma}^{1/2} \boldsymbol{x}_\ell\|_{\ell_2}^2}$. Fix an arbitrary $\lambda_0 > 0$ for now (we will determine its value later) and define truncated parameter $\bar{\lambda} = \min(\lambda, \lambda_0)$. We have

$$\begin{aligned}
\mathbb{E}\left[e^{-\lambda(\sum_i \tilde{\xi}_i)^2}\right] &\le \mathbb{E}\left[e^{-\bar{\lambda}(\sum_i \tilde{\xi}_i)^2}\right] \qquad \text{(E.14)} \\
&= \mathbb{E}\left[e^{-\bar{\lambda}\sum_i (1+u_i)^2} e^{-\bar{\lambda}\sum_{i\neq j} \xi_i \xi_j (1+u_i)(1+u_j)}\right] \\
&\le e^{-\bar{\lambda} n} \prod_{i \neq j} \mathbb{E}[e^{-\bar{\lambda}\xi_i(1+u_i)}]\, \mathbb{E}[e^{-\bar{\lambda}\xi_j(1+u_j)}] \\
&= e^{-\bar{\lambda} n} \big(\mathbb{E}[e^{-\bar{\lambda}\xi(1+u)}]\big)^{n(n-1)} \\
&= e^{-\bar{\lambda} n}\, \mathbb{E}\big[(1-p)e^{-\bar{\lambda}(1+u)} + p e^{\bar{\lambda}(1+u)}\big]^{n(n-1)} \\
&\le e^{-\bar{\lambda} n} \Big((1-p)e^{-\bar{\lambda}} + p e^{\bar{\lambda}}\, \mathbb{E}[e^{\bar{\lambda} u}]\Big)^{n(n-1)} \\
&\le e^{-\bar{\lambda} n} \Big((1-p)e^{-\bar{\lambda}} + p e^{\bar{\lambda}} e^{\bar{\lambda}^2/2}\Big)^{n(n-1)} \\
&= e^{-\bar{\lambda} n^2} \Big(1 - p + p e^{2\bar{\lambda} + \bar{\lambda}^2/2}\Big)^{n(n-1)}. \qquad \text{(E.15)}
\end{aligned}$$

Here, the first and the second inequality follows from $u_i \ge 0$. The third inequality holds since u has unit sub-gaussian norm and so it moment-generating function is bounded as $\mathbb{E}[e^{\lambda u}] \le e^{\lambda^2/2}$.

We next further upper bound the right-hand side of E.15 to get a simpler expression. In doing that we use the following lemma.

Lemma E.3 *For $t \le 1/2$ we have $e^t - 1 \le t + t^2$.*

By choosing $\lambda_0 \le 0.2$ we have $2\lambda_0 + \lambda_0^2/2 \le 1/2$ and so $2\bar{\lambda} + \bar{\lambda}^2/2 \le 1/2$. By virtue of the above lemma we have

$$\begin{aligned}
e^{2\bar{\lambda} + \bar{\lambda}^2/2} - 1 &\le 2\bar{\lambda} + \bar{\lambda}^2/2 + \big(2\bar{\lambda} + \bar{\lambda}^2/2\big)^2 \\
&= 2\bar{\lambda} + \bar{\lambda}^2/2 + 4\bar{\lambda}^2 + \bar{\lambda}^4/4 + 2\bar{\lambda}^3 \\
&\le 2\bar{\lambda} + \bar{\lambda}\big(\lambda_0/2 + 4\lambda_0 + \lambda_0^3/4 + 2\lambda_0^2\big) \\
&\le (2 + 6\lambda_0)\bar{\lambda},
\end{aligned}$$

where we used that $\lambda_0 \le 0.2$ in the last step. Next using the inequality $1 + x \le e^x$ for $x \ge 0$, we get

$$(e^{2\bar{\lambda} + \bar{\lambda}^2/2} - 1)p + 1 \le (2 + 6\lambda_0)p\bar{\lambda} + 1 \le e^{(2+6\lambda_0)\bar{\lambda} p}.$$

By using this bound in (E.15) we obtain

$$\mathbb{E}\left[e^{-\lambda(\sum_i \tilde{\xi}_i)^2}\right] \le e^{-\bar{\lambda} n^2} e^{(2+6\lambda_0)\bar{\lambda} p n^2} = e^{-(1-2p')\bar{\lambda} n^2}, \qquad \text{(E.16)}$$

with $p' := (1 + 3\lambda_0)p$.

Using (E.15) into (E.13) and then taking expectation with respect to $\boldsymbol{z}_\ell$ we get

$$\mathbb{P}\left(\widetilde{y}'_\ell\langle \boldsymbol{x}_\ell, \widehat{\boldsymbol{\theta}}_0\rangle < 0\right) \le \frac{1}{2}\,\mathbb{E}\left[\exp\left(-(1-2p')\bar{\lambda}n^2\right)\exp\left(\frac{\|\boldsymbol{z}_\ell\|_{\ell_2}^2}{2n}\right)\right]$$
$$\le \frac{1}{2}\,\mathbb{E}\left[\exp\left(-2(1-2p')\bar{\lambda}n^2\right)\right]^{1/2}\mathbb{E}\left[\exp\left(\frac{\|\boldsymbol{z}_\ell\|_{\ell_2}^2}{n}\right)\right]^{1/2}, \tag{E.17}$$

by applying Cauchy–Schwarz inequality.

Using the moment generating function of χ_d^2 distribution, we have

$$\mathbb{E}\left[\exp\left(\frac{\|\boldsymbol{z}_\ell\|_{\ell_2}^2}{n}\right)\right] = \left(1-\frac{2}{n}\right)^{-d/2} \le e^{2d/n}, \tag{E.18}$$

for $n \ge 4$. Here, we use the inequality $(1-x)^{-1} \le e^{2x}$ for $x \in [0, 1/2]$.

We continue by bounding the first term on the right-hand side of (E.17). Note that $|\langle \boldsymbol{x}_\ell, \boldsymbol{\mu}\rangle| = |y(1+u_\ell)\,\|\boldsymbol{\mu}\|_{\ell_2}^2| \ge \gamma^2$ because $u_\ell \ge 0$. Define the event $\mathcal{E}_1$ as follows:

$$\mathcal{E}_1 := \left\{\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_\ell\right\|_{\ell_2} \le \lambda_{\max}\sqrt{2d}\right\} \tag{E.19}$$

We then have

$$\begin{aligned}
\mathbb{E}\left[\exp\left(-2(1-2p')\bar{\lambda}n^2\right)\right] &= \mathbb{E}\left[\exp\left(-2(1-2p')\bar{\lambda}n^2\right)(\mathbb{1}_{\mathcal{E}_1} + \mathbb{1}_{\mathcal{E}_1^c})\right] \\
&\le \mathbb{E}\left[\exp\left(-2(1-2p')\bar{\lambda}n^2\right)\mathbb{1}_{\mathcal{E}_1}\right] + \mathbb{E}[\mathbb{1}_{\mathcal{E}_1^c}] \\
&\overset{(a)}{=} \mathbb{E}\left[\exp\left(-\frac{\gamma^4}{4\lambda_{\max}^2 d}(1-2p')n\right)\mathbb{1}_{\mathcal{E}_1}\right] + \mathbb{P}(\mathcal{E}_1^c) \\
&\le \exp\left(-\frac{\gamma^4}{4\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + \mathbb{P}(\mathcal{E}_1^c)
\end{aligned} \tag{E.20}$$

Note that in step (a), we used the fact that on the event $\mathcal{E}_1$, we have $\lambda \ge \gamma^4/(8\lambda_{\max}^2 nd)$ and we choose $\lambda_0 = \gamma^4/(8\lambda_{\max}^2 nd)$. This way, we have $\bar{\lambda} = \min(\lambda, \lambda_0) = \gamma^4/(8\lambda_{\max}^2 nd)$. Also note that by our assumption in the statement of the lemma, we have $\lambda_0 \le 0.2$ which is the condition assumed in deriving (E.16).

We next bound $\mathbb{P}(\mathcal{E}_1^c)$. Under our data model we have $\boldsymbol{\Sigma}^{1/2}\boldsymbol{x} = \boldsymbol{\Sigma}\boldsymbol{z}$ and so $\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2} \le \lambda_{\max}\|\boldsymbol{z}\|_{\ell_2}$. Also since $\boldsymbol{z} \sim \mathsf{N}(0, \boldsymbol{I}_d)$, $\|\boldsymbol{z}\|_{\ell_2}^2$ is χ^2 distribution with d-degree of freedom using the tail bound of χ^2 distribution (see e.g, (Ghosh, 2021, Theorem 3)) we have $\mathbb{P}(\|\boldsymbol{z}\|_{\ell_2}^2 \ge 2d) \le e^{-d/8}$ and so

$$\mathbb{P}(\mathcal{E}_1^c) \le \mathbb{P}(\|\boldsymbol{z}\|_{\ell_2}^2 \ge 2d) \le e^{-d/8}.$$

Using the above in (E.20) we obtain

$$\mathbb{E}\left[\exp\left(-2(1-2p')\bar{\lambda}n^2\right)\right] \le \exp\left(-\frac{\gamma^4}{4\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/8} \tag{E.21}$$

By combining (E.21) and (E.18) with (E.17) we arrive at

$$\begin{aligned}
\mathbb{P}\left(\widetilde{y}'_\ell\langle \boldsymbol{x}_\ell, \widehat{\boldsymbol{\theta}}_0\rangle < 0\right) &\le \frac{1}{2}\left(\exp\left(-\frac{\gamma^4}{4\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/8}\right)^{1/2} e^{d/n} \\
&\le \frac{1}{2}\left(\exp\left(-\frac{\gamma^4}{8\lambda_{\max}^2}(1-2p')\frac{n}{d}\right) + e^{-d/16}\right) e^{d/n},
\end{aligned}$$

which completes the proof of lemma.

E.2 Proof of Lemma E.2

Under our data model we have $\boldsymbol{x}_\ell = y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell$. Substituting for $\boldsymbol{x}_\ell$ in the expression of T_ℓ, we have

$$\begin{aligned} T_\ell &= \operatorname{sign}\left(\langle y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell, \beta\boldsymbol{\mu} + \xi_\ell y_\ell \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\rangle\right)\langle y\boldsymbol{x}, y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\rangle \\ &= \operatorname{sign}\left(y_\ell(1+u_\ell)\beta\gamma^2 + \xi_\ell y_\ell \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\right\|_{\ell_2}^2\right)\langle y\boldsymbol{x}, y_\ell(1+u_\ell)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\rangle\,, \end{aligned}$$

using the fact that $\boldsymbol{\Sigma}^{1/2}\boldsymbol{\mu} = 0$. Taking expectation we get

$$\mathbb{E}[T_\ell|\beta] = \mathbb{E}\left[\operatorname{sign}\left(y_\ell(1+u_\ell)\beta\gamma^2 + \xi_\ell y_\ell \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\right\|_{\ell_2}^2\right)\langle y\boldsymbol{x}, y_\ell(1+u_\ell)\boldsymbol{\mu}\rangle \,\Big|\beta\right]$$

since the other term will be an odd function of $\boldsymbol{z}_\ell$. Letting

$$q_0 := \mathbb{P}\left((1+u_\ell)\beta\gamma^2 + \xi_\ell \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_\ell\right\|_{\ell_2}^2 < 0\right),$$

we get

$$\mathbb{E}[T_\ell|\beta] = (1-2q_0)\langle y\boldsymbol{x}, (1+u_\ell)\boldsymbol{\mu}\rangle \geq (1-2q_0)\langle y\boldsymbol{x}, \boldsymbol{\mu}\rangle\,,$$

given that $u_\ell > 0$ and $\langle y\boldsymbol{x}, \boldsymbol{\mu}\rangle > 0$ given the positive margin in our data model. So what remains is to show that $q_0 \leq q$. We write

$$\begin{aligned} q_0 &\leq \mathbb{P}\left((1+u_\ell)\beta\gamma^2 - \lambda_{\max}\left\|\boldsymbol{z}_\ell\right\|_{\ell_2}^2 < 0\right) \\ &\leq \mathbb{P}\left(\frac{\beta\gamma^2}{\lambda_{\max}} < \left\|\boldsymbol{z}_\ell\right\|_{\ell_2}^2\right) < \exp\left(-\frac{\beta\gamma^2}{20\lambda_{\max}}\right), \end{aligned}$$

if $\frac{\beta\gamma^2}{2d\lambda_{\max}} > 1$, where the last step follows from the observation that $\left\|\boldsymbol{z}_\ell\right\|_{\ell_2}^2 \sim \chi_d^2$ and using the tail bound of χ_d^2 ((Ghosh, 2021, Theorem 3)). This can be alternatively written as

$$q_0 \leq q = \exp\left(-\frac{\beta\gamma^2}{20\lambda_{\max}}\right) + \mathbb{1}\left(\frac{\beta\gamma^2}{2d\lambda_{\max}} < 1\right),$$

which completes the proof of lemma.

F Proof of Theorem 4.9

Using Theorem 4.8, we have $\mathsf{acc}(\widehat{\boldsymbol{\theta}}_1) \geq \mathbb{E}[\alpha_1(\boldsymbol{x})]$. Based on our data model $\boldsymbol{x} = y(1+u)\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}$. Consider the probabilistic event $\mathcal{E}_2 := \{\boldsymbol{z} : \|\boldsymbol{z}\|_{\ell_2}^2 \leq 2d\}$. For every $\eta > 0$, we have

$$\begin{aligned} \mathbb{E}\left[\exp\left(-\frac{\eta\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] &= \mathbb{E}\left[\exp\left(-\frac{\eta\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)(\mathbb{1}_{\mathcal{E}_2} + \mathbb{1}_{\mathcal{E}_2^c})\right] \\ &\leq \exp\left(-\frac{\eta\gamma^4}{\gamma^4 + 2\lambda_{\max}^2 d}\right) + \mathbb{P}(\mathcal{E}_2^c)\,, \end{aligned}$$

where the inequality holds because $\langle\boldsymbol{x},\boldsymbol{\mu}\rangle = y(1+u)\gamma^2$ and so $\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2 \geq \gamma^4$ (since u is a non-negative random variable). In addition,

$$\left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2 = \|\boldsymbol{\Sigma}\boldsymbol{z}\|_{\ell_2}^2 \leq \lambda_{\max}^2\|\boldsymbol{z}\|_{\ell_2}^2 \leq 2\lambda_{\max}^2 d$$

on the event $\mathcal{E}_2$. Further, $\|\boldsymbol{z}\|_{\ell_2}^2 \sim \chi_d^2$ and so using tail bounds for χ_d^2, see e.g., (Ghosh, 2021, Theorem 3), we have $\mathbb{P}(\mathcal{E}_2^c) \leq e^{-d/8}$. Putting things together, we obtain

$$\mathbb{E}\left[\exp\left(-\frac{\eta\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2}{\langle\boldsymbol{x},\boldsymbol{\mu}\rangle^2 + \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right\|_{\ell_2}^2}\right)\right] \leq \exp\left(-\frac{\eta\gamma^4}{\gamma^4 + 2\lambda_{\max}^2 d}\right) + e^{-d/8}\,.$$

The claim of theorem follows by setting $\eta = cn(1-2q')^2$.

G Remaining Experimental Details

Here we provide the remaining details about the experiments in Section 5. Our experiments were done using TensorFlow and JAX on one 40 GB A100 GPU (per run). In all the cases, we retrain starting from random initialization rather than the previous checkpoint we converged to before RT; the former worked better than the latter. We list training details for each individual dataset next.

CIFAR-10. Optimizer is SGD with momentum = 0.9, batch-size = 32, number of gradient steps in each stage of training (i.e., both stages of baseline, full RT and consensus-based RT) = 21k. We use the cosine one-cycle learning rate schedule with initial learning rate = 0.1 for each stage of training. The number of gradient steps and initial learning rate were chosen based on the performance of the baseline method and *not* based on the performance of full or consensus-based RT. Standard augmentations such as random cropping, flipping and brightness/contrast change were used.

CIFAR-100. Details are the same as CIFAR-10 except that here the number of gradient steps in each stage of training = 28k and initial learning rate = 0.005.

AG News Subset. Small BERT model link: https://www.kaggle.com/models/tensorflow/bert/frameworks/tensorFlow2/variations/bert-en-uncased-l-4-h-512-a-8/versions/2?tfhub-redirect=true, BERT English uncased preprocessor link: https://www.kaggle.com/models/tensorflow/bert/frameworks/tensorFlow2/variations/en-uncased-preprocess/versions/3?tfhub-redirect=true. Optimizer is Adam with fixed learning rate = $1e-5$, batch size = 32, number of epochs in each training stage = 5.

The test accuracies *without* label DP for CIFAR-10, CIFAR-100 and AG News Subset are $94.13 \pm 0.05\%$, $74.73 \pm 0.34\%$ and $91.01 \pm 0.25\%$, respectively.

H Consensus-Based Retraining Does Better than Confidence-Based Retraining

Here we compare full and consensus-based RT against another strategy for retraining which we call **confidence-based retraining (RT)**. Specifically, we propose to retrain with the predicted labels of the samples with the top 50% margin (i.e., highest predicted probability - second highest predicted probability); margin is a measure of the model's confidence. This idea is similar to self-training's method of sample selection in the semi-supervised setting (Amini et al., 2022). In Tables 8 and 9, we show results for CIFAR-10 and CIFAR-100 (in the same setting as Section 5 and Appendix G) with the smallest value of ϵ from Tables 1 and 2, respectively. Notice that *consensus-based RT is clearly better than confidence-based RT*.

Table 8: **CIFAR-10.** Test set accuracies (mean ± standard deviation). *Consensus-based RT performs the best.*

ϵ	Baseline	Full RT	Consensus-based RT	Confidence-based RT
1	57.78 ± 1.13	60.07 ± 0.63	**63.84** ± 0.56	62.09 ± 0.55

Table 9: **CIFAR-100.** Test set accuracies (mean ± standard deviation). Again, *consensus-based RT performs the best.*

ϵ	Baseline	Full RT	Consensus-based RT	Confidence-based RT
3	23.53 ± 1.01	24.42 ± 1.22	**29.98** ± 1.11	24.99 ± 1.25

I Retraining is Beneficial Even Without a Validation Set

In all our previous experiments, we assumed access to a small clean validation set. Here we consider training ResNet-34 on CIFAR-100 when we do *not* have access to a validation set and keep training for a fixed number of epochs (100). The point of these experiments is to show that retraining can offer gains even without a validation set, in which case we can expect severe overfitting. Please see the the results and discussion in Table 10.

Table 10: **No validation set (CIFAR-100 w/ ResNet-34).** Test set accuracies (mean ± standard deviation). Just like all our previous results in the presence of a small validation set, consensus-based RT is the clear winner here. However, due to a higher degree of overfitting here, the performance and amount of improvement obtained with retraining is worse than the corresponding experiment using a validation set (Table 4). This is not surprising.

ϵ	Baseline	Full RT	Consensus-based RT
3	14.53 ± 0.48	15.20 ± 0.71	**17.30** ± 0.43
4	26.03 ± 1.75	28.33 ± 1.76	**30.47** ± 1.31
5	42.50 ± 1.14	44.43 ± 1.27	**46.27** ± 1.72

J Beyond Label DP: Evaluating Retraining in the Presence of Human Annotation Errors

Even though our empirical focus in this paper has been label DP training, retraining (RT) can be employed for general problems with label noise. Here we evaluate RT in a setting with "real" label noise due to *human annotation*. Specifically, we focus on training a ResNet-18 model (*without* label DP to be clear) on the *CIFAR-100N* dataset introduced by Wei et al. (2021) and available on the TensorFlow website. CIFAR-100N is just CIFAR-100 labeled by humans; thus, it has real human annotation errors. The experimental setup and details are the same as CIFAR-100 (as stated in Section 5 and Appendix G); the only difference is that here we use initial learning rate = 0.01.

In Table 11, we list the test accuracies of the baseline which is just vanilla training with the given labels, full RT and consensus-based RT, respectively. Even here *with human annotation errors*, consensus-based RT is beneficial.

Table 11: **CIFAR-100N.** Test set accuracies (mean ± standard deviation). *So even with real human annotation errors, consensus-based RT improves performance*.

Baseline	Full RT	Consensus-based RT
55.47 ± 0.18	56.88 ± 0.35	**57.68** ± 0.35