PROTECTIVE LABEL ENHANCEMENT FOR LABEL PRIVACY

Anonymous authors
Paper under double-blind review

ABSTRACT

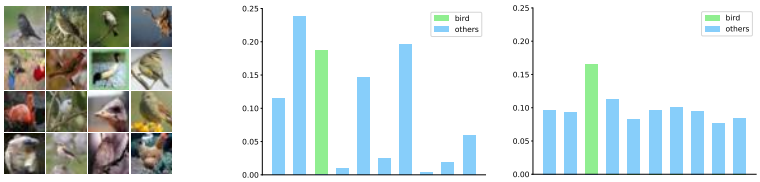
Much sensitive data is gathered from the individual device for commercial value without effective safeguards over the past decade, which would bring on serious privacy leakage. Here we review the label differential privacy (label DP), which means that only the labels are sensitive. The generated private labels of previous methods do not consider the label confidence corresponding to the feature and multiple sampling could be employed to identify the true labels. In the paper, a novel approach called Protective Label Enhancement (PLE) is proposed to mask the true label in the label distribution while ensuring that the protective label distribution is utility for training an effective predictive model on the server. Specifically, when we generate the label distribution, the true label is mixed up by choosing several random labels and punishment will be applied when the true is at the top of the label distribution. Meanwhile, if the true label almost vanishes, it will be compensated to keep the statistical effectiveness. Furthermore, we provide the corresponding theoretical guarantee that the predictive model is *classifier-consistent* and that learning with the protective label distribution is ERM *learnable*. Finally, experimental results clearly validate the effectiveness of the proposed approach for solving the label DP problem. The source code will be released for public access.

1 INTRODUCTION

With the advance of machine learning, there has been an ever-growing interest in collecting user data to improve products. Abuse of sensitive data would result in privacy leakage, which motivates the development of privacy-preserving machine learning Abadi et al. (2016); Bonawitz et al. (2017); Papernot et al. (2017). Therefore, a framework called Differential Privacy (DP) Dwork et al. (2006b) that provides provable protection for users has emerged as a popular privacy notion and has been deployed in industry Erlingsson et al. (2014); Greenberg (2016); Shankland (2014).

Actually, many real-world applications need not protect all data, but only labels are considered sensitive and need to be protected Smith et al. (2018); Ghazi et al. (2021); Malek Esmaeili et al. (2021), e.g. online advertising and user experience programs. Label differential privacy (label DP) Chaudhuri & Hsu (2011); Beimel et al. (2013); Ghazi et al. (2021) is proposed to protect the privacy of labels. In label DP, there are two unreconciled goals as follows. 1) Privacy: when sensitive labels are collected, the server is difficult to reveal private labels individually. 2) Utility: the labels collected by the server are utility and the trained predictive model has a good performance.

A classical algorithm of label DP is *Randomized Response* (RR), which is designed to eliminate evasive answer bias Warner (1965). In RR, the response is not always honest, and the sensitive label could be replaced by a sample drawn from its entire domain with a certain probability. However, RR is limited in two aspects. On the one hand, the probability distribution of the labels is often determined beforehand and is not combined with the features. On the other hand, since the probability of the true label is higher than other labels, the true label can be found by multiple sampling. *Additive Noise* is another essential algorithm of label DP Dwork et al. (2006a;b). Prior noise is added to the logical label such as Gaussian noise and Laplacian noise. The level of noise is directly related to the privacy budget. Nevertheless, *Additive Noise* is difficult to deploy in practice because high-level noise pollutes the true label severely and low-level noise could mask the true label. Same as RR, the noise is neither combined with the features.



(a) (b) (c)

Figure 1: Partial illustrations on CIFAR-10: (a) images of the birds, (b) an example of the protective label distribution, (c) the mean of the protective label distribution of all birds.

Motivated by the *label distribution learning* Geng (2016); Xu et al. (2019), we hide the true label in the protective label distribution as illustrated in Figure 1 (b). In the protective label distribution, the true label is probably not at the top so that the sensitive label is protected. In addition, the label distribution needs to be utility, which means the server can train a good predictive model with the protective label distribution. In this sense, we should keep the protective label distribution statistically effective as demonstrated in Figure 1 (c).

We propose an approach called Protective Label Enhancement (PLE), which converts the logical label to label distribution to hide the true label. Specifically, we apply punishment when the true label is at the top of the label distribution and utilize several random labels to mask the true label. Here we utilize random labels because they could be less correlated with the true label and high correlation could mislead the predictive model. After PLE, the server will collect the protective label distribution and train the predictive model. Since the label distribution contains a lot of irregular noise which prevents label leakage, learning from it directly could be ineffective. A regularization term Bachman et al. (2014); Wu et al. (2022), which is effective in handling the noise, is adopted for the predictive model. Furthermore, we prove that PLE is *classifier-consistent* and that learning with the high-noise label distribution is ERM *learnable* when the PLE model has high top-k accuracy (k is a constant selected in advance) and small ambiguity degree. Finally, the experiments verify that PLE is privacy-preserving and utility. Our contributions can be summarized as follows:

- A new approach called Protective Label Enhancement (PLE) is proposed, which employs the customized label distribution to hide the true label for the first time. The protective label distribution protects instance-level label privacy while maintains statistical effectiveness, which provides an excellent trade-off between privacy and utility.
- We prove that the predictive model is *classifier-consistent* and learning with the protective label distribution is ERM *learnable*. The protective label distribution is guaranteed to be effective when satisfying the conditions of unreliability degree and ambiguity degree in Theorem 2.
- The experiments also demonstrate that privacy protection can be achieved at a modest cost in the quality of the predictive model. When providing very strong privacy levels, our predictive model achieves 30% (20%) relative improvement on CIFAR-10 (CIFAR-100) over previous state-of-the-art.

2 Related Work

Label DP For algorithms on aggregate databases, DP constitutes a strong standard called privacy budget to protect privacy. Then, label differential privacy (label DP) Chaudhuri & Hsu (2011); Beimel et al. (2013); Ghazi et al. (2021) is proposed to measure the privacy of labels. It shows that when only the label needs to be protected, the model performance can be significantly improved because the constraint of label DP is slacker. Label DP was firstly studied in PAC setting and the sample complexity bounds are provided for label DP Chaudhuri & Hsu (2011); Beimel et al. (2013).

Two canonical algorithms, RR and additive noise, are applied to protect label privacy. RR was firstly proposed in Warner (1965) and employs a randomized strategy that could respond dishonestly with a certain probability. For example, Erlingsson et al. (2014) turned the binary bit vector in a certain

randomized manner. Kairouz et al. (2016) expanded the mechanism to k categories and showed that the optimal decoding algorithm depended on the underlying distribution. Recently, Ghazi et al. (2021) proposed Randomized Response with Prior (RRWithPrior) where incorrect labels collected from clients satisfies a prior distribution, and the server employed Multi-Stage Training (LP-MST) to progressively learns refined prior distribution of the instance. Additive noise mechanism is also widely adopted in label DP, such as additive Laplace Dwork et al. (2006b) and additive Gaussian Dwork et al. (2006a). In Smith et al. (2018), additive noise was combined with Gaussian processes in the linear regression. Malek Esmaeili et al. (2021) proposed applying additive Laplace noise to the logical label and the server de-noised the mechanism's output by performing Bayesian inference using the prior provided by the model itself.

Label Enhancement In the label distribution learning (LDL) Geng (2016), it is general to employ the label distribution to reflect the relationship between the label and the instance. However, due to the difficulty of obtaining the label distributions directly, label enhancement (LE) Xu et al. (2019) was proposed to recover label distributions from logical labels, where the implicit relative importance among different labels and the correlation between them could be mined. Many novel LE algorithms have been put forward in recent years and these approaches aim to improve the predictive model with the label distribution Xu et al. (2022); Zhao et al. (2022).

In this paper, the proposed PLE utilizes the label distribution to mask the true label while keeping the supervision information. In other words, while providing privacy guarantees, statistical effectiveness is kept as much as possible. However, different from traditional LE, the protective label distribution in PLE contains much noise and is not designed to describe the instance. Therefore, instead of leveraging the inherent correlation between the instances, PLE aims to decrease the correlation because similar labels increase the difficulty to distinguish the true label from the label distribution. Specifically, we apply punishment to the honest response that the true label is at the top of the label distribution and compensate for the true label when it almost vanishes. Besides, several random incorrect labels are employed in the target function to decrease the impact of similar labels.

After collecting protective data from clients, it is essential for the server to train an effective predictive model. For RR-based or noise-based data, semi-supervised learning techniques are often utilized for the predictive model. Inspired by the success of manifold regularization in semi-supervised learning, we employ a regularization term and progressively update the label distribution to disambiguate the label distribution. In detail, it encourages the prediction of the network to be similar in the vicinity of the observed training samples. Experimental results validate that it is effective to learn with the protective label distribution and our method significantly improves over the previous state-of-the-art private baselines.

3 METHOD

3.1 PRELIMINARY

7 Let $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ be the underlying probability distribution of the dataset. Let $\boldsymbol{x} \in \mathcal{X}$ denote the instance variable and $y \in \mathcal{Y}$ denote the corresponding true label. For a problem with c different labels, the label space can be denoted as $\mathcal{Y} = \{1, 2, \ldots, c\}$. To avoid ambiguity, we use y to denote the true label and l to denote any label in the label space, i.e. $l \in \mathcal{Y}$. Let $\boldsymbol{d}_x$ be the label distribution of $\boldsymbol{x}$. We will write it as $\boldsymbol{d}$ for simplicity when there is no ambiguity. Let $d_{\boldsymbol{x}}^l$ be the degree to which label l describes $\boldsymbol{x}$, where $d_{\boldsymbol{x}}^l \in [0, 1]$ and $\sum_l d_{\boldsymbol{x}}^l = 1$. Label distribution has the same formalization as the probability distribution and so we denote $\mathcal{P}$ as the space of label distribution. The top label of the distribution $\boldsymbol{d}_{\boldsymbol{x}}$ is denoted by $\operatorname{argmax}_{l \in \mathcal{Y}}(d_{\boldsymbol{x}}^l)$. In addition, $\Omega_{k_1}^{\boldsymbol{d}}(\boldsymbol{x}) = \{l \in \mathcal{Y} \mid l \text{ ranks top-}k_1 \text{ in } \boldsymbol{d}_{\boldsymbol{x}}\}$ denotes the top-k_1 labels of the distribution, where k_1 is a hyperparameter.

In our framework, the client and server each have a distinct model. We define $f : \boldsymbol{x} \to \boldsymbol{d}$ as the PLE model to protect label privacy on the client and $\Omega_{k_1}^f(\boldsymbol{x}) = \{l \in \mathcal{Y} \mid l \text{ ranks top-}k_1 \text{ in } f(\boldsymbol{x})\}$. We define $h : \boldsymbol{x} \to y$ as the predictive model to recover the true label from the protective label distribution on the server. For the instances $\boldsymbol{x}$, $f(\boldsymbol{x})$ is the probability distribution predicted by the model f and $f_i(\boldsymbol{x})$ is the degree of the i-th label. This also applies to $h(x)$. The space of label distribution generated by f is denoted by $\mathcal{P}_f$. After collecting data from the clients, the classifier h is trained on datasets sampled from $\mathcal{X} \times \mathcal{P}_f$.

3.2 PROTECTIVE LABEL ENHANCEMENT FOR LABEL PRIVACY

The PLE model f with parameters $\boldsymbol{\theta}_1$ is trained to hide the true label y in the distribution $\boldsymbol{d}$. In other words, the attacker is difficult to recover the true label y from $\boldsymbol{d}$. Intuitively, it means that the accuracy of the label enhancement model remains low. Therefore, the instances which are predicted correctly will be punished by a loss term $\mathcal{L}_{\text{pun}}$:

$$\mathcal{L}_{\text{pun}}(\boldsymbol{x}, y) = -\mathbb{I}(\underset{j \in \mathcal{Y}}{\operatorname{argmax}}(\boldsymbol{d}_j) = y)\ell(f(\boldsymbol{x}), y), \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\ell(\cdot, \cdot)$ is the cross entropy loss function. But in practice, the degree of the true label decreases significantly, resulting in low accuracy of the predictive model h. So the true label y is compensated when $y \notin \Omega^{\boldsymbol{d}}_{k_1}(\boldsymbol{x})$:

$$\mathcal{L}_{\text{comp}}(\boldsymbol{x}, y) = \mathbb{I}(y \notin \Omega^{\boldsymbol{d}}_{k_1}(\boldsymbol{x}))\ell(f(\boldsymbol{x}), y). \tag{2}$$

The compensation term is designed to improve the top-k accuracy of the PLE model, which keeps the statistical effectiveness of the protective label distribution. In fact, $\mathcal{L}_{\text{pun}}$ and $\mathcal{L}_{\text{comp}}$ can moderately balance the privacy protection and the performance of the subsequent predictive model. In practice, however, we discovered there was a strong correlation among the top-k labels, which leaded to the confusion between the true label and similar labels. For example, the images of *cat* are prone to be mistaken for *dog* than *apple*, and hence *dog* tend to be more weighted in the label distribution of *cat*. If the weight of *dog* is always higher than *cat* in general, the label *cat* may not be distinguishable.

To decrease this correlation, we propose an effective method to make the labels in $\Omega^{\boldsymbol{d}}_{k_1}(\boldsymbol{x})$ as independent as possible. Except for the true label, the other $k_1 - 1$ labels are selected randomly. Then the selected $k_1 - 1$ labels are employed as the learning objectives:

$$\mathcal{L}_{\text{rnd}}(\boldsymbol{x}, y) = \ell(f(\boldsymbol{x}), s(y)). \tag{3}$$

where $s(y)$ is the set of the $k_1 - 1$ random labels excluding y. Consequently, the objective of the label enhancement model is as follows:

$$\mathcal{L}_{\text{PLE}}(\boldsymbol{x}, y) = \mathcal{L}_{\text{ce}}(\boldsymbol{x}, y) + \alpha_1\mathcal{L}_{\text{pun}}(\boldsymbol{x}, y) + \alpha_2\mathcal{L}_{\text{comp}}(\boldsymbol{x}, y) + \alpha_3\mathcal{L}_{\text{rnd}}(\boldsymbol{x}, y) \tag{4}$$

where $\mathcal{L}_{\text{ce}}(\boldsymbol{x}, y)$ is the vanilla cross-entropy loss between the output and the true label, and α_1, α_2, α_3 are the tradeoff parameters.

3.3 TRAINING THE PREDICTIVE MODEL

After collecting enhanced data from clients, the server aims to train the predictive model h with parameters $\boldsymbol{\theta}_2$ on the dataset $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^n$. Since the label distribution contains a lot of noise, we preprocess the label distribution and employ the strategy from the semi-supervised learning. Firstly, we zeroize labels which are not top-k_2 in the label distribution and normalize the distribution, where k_2 is a hyperparameter that can be different from k_1. Then, we employ a regularization term widely adopted in semi-supervised learning and it is suitable for such noisy label distribution. The basic idea behind the regularization is that, after a small modification of the feature, the prediction should be similar to the original label distribution. Meanwhile, in the training procedure, the label distribution is updated and becomes more accurate progressively. Therefore, we denote $\boldsymbol{p}(\boldsymbol{x})$ as the targeted label distribution in the training of the predictive model, which is initialized as $\boldsymbol{d}(\boldsymbol{x})$.

Optimize the Predictive Model Specifically, after initialization, we need to obtain the supervised loss first. Due to the unreliability of the label distribution, the model is optimized contrary to the negative labels as follows:

$$\mathcal{L}_{\text{neg}}(\boldsymbol{x}, \boldsymbol{p}) = -\sum_{j=1}^{c} \mathbb{I}(j \notin \Omega^{\boldsymbol{p}}_{k_2}(\boldsymbol{x}))\log(1 - h_j(\boldsymbol{x})). \tag{5}$$

where $h_j(\boldsymbol{x})$ denotes the output of the classifier h on label j given input $\boldsymbol{x}$. The optimization loss on incorrect labels has achieved excellent performance Gao & Zhang (2021).

Besides, it is essential to utilize the top-k_2 labels in the label distribution. We employ κ different data augmentations $\{A_j\}_{j=1}^{\kappa}$ instead of adding perturbation to the feature Wu et al. (2022). In detail, the

Algorithm 1 The overall of our methods

Input: Training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$; Initialized PLE model f with parameters $\boldsymbol{\theta}_1$; Initialized predictive model h with parameters $\boldsymbol{\theta}_2$; Training epoch T_1, T_2; The number of random labels k_1; The number of nonzeroize labels k_2.

Procedure:

Begin training PLE model with dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$;
Generate random label set s for each $\boldsymbol{x}$, where $|s| = k_1$;
for $t_1 = 1$ **to** T_1 **do**
    Calculate the loss according Eq.(4);
    Update the parameters $\boldsymbol{\theta}_1$ of the model f;
end for
Generate protective dataset $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^n$ by PLE model, where $\boldsymbol{d}_i = f(\boldsymbol{x}_i)$;
Begin training predictive model with dataset $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^n$;
Zeroize labels not in top-k_2 and then normalize $\boldsymbol{d}$;
Initialize $\boldsymbol{p} = \boldsymbol{d}$;
for $t_2 = 1$ **to** T_2 **do**
    Calculate the loss according Eq.(8);
    Update the parameters $\boldsymbol{\theta}_2$ of the model h;
    Update the label distribution by Eq.(9);
end for

Output: Predictive model h.

output of each instance after augmentation is aligned with the label distribution as the regularization. We denote ℓ as the cross-entropy loss and the augmentation loss can be written as:

$$\mathcal{L}_{\text{aug}}(\boldsymbol{x}, \boldsymbol{p}) = \sum_{j=1}^{\kappa} \ell(h(A_j(\boldsymbol{x})), \boldsymbol{p}(\boldsymbol{x})). \tag{6}$$

Otherwise, a dynamic balancing factor λ_t, whose maximum is λ, is applied to adjusting the tradeoff between $\mathcal{L}_{\text{neg}}$ and $\mathcal{L}_{\text{aug}}$:

$$\lambda_t = \min\left\{\frac{t}{T'}\lambda, \lambda\right\}, \tag{7}$$

where t is the epoch number and T' is the epoch when the balancing factor remains unvarying. Consequently, the overall objective of the predictive model is

$$\mathcal{L}_{\text{pred}}(\boldsymbol{x}, \boldsymbol{p}) = \mathcal{L}_{\text{aug}}(\boldsymbol{x}, \boldsymbol{p}) + \lambda_t \mathcal{L}_{\text{neg}}(\boldsymbol{x}, \boldsymbol{p}). \tag{8}$$

Update the Label Distribution However, the model would overfit the noisy labels in the label distribution. In noisy label learning, experiments show that the deep network firstly fits correct labels and then gradually fits incorrect labels through the learning phase, which is called memorization effect Bai et al. (2021). Correspondingly, a basic strategy is to reinforce the labels with high confidence. We update our label distribution with the output of the augmentations as Wu et al. (2020; 2022):

$$\boldsymbol{p}_i(\boldsymbol{x}) = \frac{(\prod_{j=1}^{\kappa} h_i(A_j(\boldsymbol{x})))^{\frac{1}{\kappa}}}{\sum_{i=1}^{c}(\prod_{j=1}^{\kappa} h_i(A_j(\boldsymbol{x})))^{\frac{1}{\kappa}}}. \tag{9}$$

The label distribution $\boldsymbol{p}(\boldsymbol{x})$ in Eq.(9) minimizes the overall loss $\mathcal{L}$. Then, we iteratively to optimize the model by (8) and update the label distribution $\boldsymbol{p}(\boldsymbol{x})$. The overall procedure of our method is presented in Algorithm 1.

4 Theoretical Analysis

In the proposed method, it is convinced that the true label is hidden in the label distribution. But there are still doubts about whether the predictive model can disambiguate the true label from the protective label distribution. It is conceivable that the distribution is unlearnable if the supervised information of the true label is completely erased, such as that all label distributions are uniform

across all categories. In the following, we give theoretical guarantees about the effectiveness of the protective label distribution. In particular, under certain conditions of the reliability degree and the ambiguity degree, the learning performance can be guaranteed as Theorem 2.

The crucial problem is how to measure the effectiveness of the label distribution when it is not at the top of the label distribution. Let me reiterate that $\Omega_k^f(\boldsymbol{x}) = \{l \in \mathcal{Y} \mid l \text{ ranks top-}k \text{ in } f(\boldsymbol{x})\}$ [1] as the set of the top-k labels in the label distribution and the rest of the labels are zeroized as section 3.3. Then, the effectiveness of $\boldsymbol{d}_{\boldsymbol{x}} = f(\boldsymbol{x})$ can be measured by the possibility $\Pr_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}(y \in \Omega_k^f(\boldsymbol{x}))$. It is obvious that, when k is larger, the possibility $\Pr_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}(y \in \Omega_k^f(\boldsymbol{x}))$ is higher, but the supervised information, i.e. zeroized $\boldsymbol{d}_{\boldsymbol{x}}$, could be weaker because distractor labels co-occur with the true labels more frequently.

Specifically, we define two criteria to measure supervised information. Firstly, we define Δ to measure the unreliability degree of $\boldsymbol{d}_{\boldsymbol{x}}$,

$$\Delta = \Pr_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}(y \notin \Omega_k^f(\boldsymbol{x})). \tag{10}$$

Secondly, the ambiguity degree is defined as the bound of the frequency of co-occurrence

$$\gamma = \sup_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}, i\in\mathcal{Y}, \Omega_k^f(\boldsymbol{x}\sim p(s|\boldsymbol{x},y,f), l\neq y} \Pr(l \in \Omega_k^f(\boldsymbol{x})). \tag{11}$$

In other words, if a problem exhibits ambiguity degree γ, then $\Pr(l \in \Omega_k^f(\boldsymbol{x}) \mid l \neq y, x, y) \leq \gamma$. The smaller Δ or γ is, the more supervised information the predictive model has. However, when k increases, Δ will decrease and γ is opposite, which means k should be selected cautiously.

Theorem 1 *If $\gamma < 1 - \frac{\Delta}{1-\Delta}$, which means the degree of the true label is large enough, the optimal bayesian classifier h^* satisfies $h^* = \arg\min_{h\in\mathcal{H}} R(h)$.*

The proof can be found in A.1. Theorem 1 ensures that the predictive model h is optimized towards the optimal model h^*. This property is also referred to as classifier-consistency in Feng et al. (2020), which is the statistical property of the predictive model over the entire data distribution $\mathcal{X} \times \mathcal{Y}$. However, it does not provide a guarantee for models which are trained on the practical dataset and cannot ensure the convergence of the predictive model.

Next we will prove our main result, the ERM learnability of learning with the protective label distribution. Firstly, we denote some common notations in machine learning. Let $\mathcal{H}$ denote the hypothesis space and each $h \in \mathcal{H}$ is a predictive model. The generalization error of h is defined as

$$\mathrm{Err}_{\mathcal{D}}(h) = E_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}\mathbb{I}(h(\boldsymbol{x}) \neq y).$$

Correspondingly, we define the generalization distribution error and the empirical distribution error as

$$\mathrm{Err}_{\mathcal{D}}^f(h) = E_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}\mathbb{I}(h(\boldsymbol{x}) \notin \Omega_k^f(\boldsymbol{x})),$$
$$\mathrm{Err}_{\mathbf{z}}^f(h) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left(h\left(\boldsymbol{x}_i\right) \notin \Omega_k^f(\boldsymbol{x}_i)\right),$$

where $\mathbf{z}$ is a dataset of size n. In the above equations, the set $\Omega_k^f(\boldsymbol{x})$ is determined by the label distribution $\boldsymbol{d}_{\boldsymbol{x}}$. Given that the true label y may not exist in the top-k of the $\boldsymbol{d}_{\boldsymbol{x}}$, we denote $H_\Delta = \{f : \Pr_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}(y \notin \Omega_k^f(\boldsymbol{x})) = \Delta\}$ as the set of PLE models with unreliability degree Δ. With the noisy label distribution, the task is to learn a predictive model that has good generalization. Empirical Risk Minimizing (ERM) is a common strategy. For the hypotheses space $\mathcal{H}$ and the empirical error $\mathrm{Err}_{\mathbf{z}}^f(h)$, an ERM learner $\mathcal{A}(\mathbf{z})$ returns the minimum empirical error on dataset $\mathbf{z}$.

$$\mathcal{A}(\mathbf{z}) = \arg\min_{h\in\mathcal{H}} \mathrm{Err}_{\mathbf{z}}^f(h).$$

Based on the unreliability degree Δ in (10) and the ambiguity degree γ in (11), we provide a sufficient condition that learning with the label distribution is ERM learnable. The main result is as follows.

[1] To be exact, k is equivalent to k_2 in section 3.3. Since k_1 does not appear in the theoretical analysis, k is used to denote k_2 for simplicity.

Theorem 2 *Suppose unreliability degree Δ and ambiguity degree γ, $0 < \Delta, \gamma < 1$ and $\Delta + \gamma < 1$. Let $\theta = \log \frac{2(1-\Delta)}{1-\Delta+\gamma}$ and suppose the Natarajan dimension of the hypothesis space $\mathcal{H}$ is $d_{\mathcal{H}}$. Define*

$$n_0(\mathcal{H}, \varepsilon, \delta) = \frac{2}{\frac{\theta\varepsilon}{2} + \log \frac{1}{2-2\Delta}} (d_{\mathcal{H}}(\log(2d_{\mathcal{H}}) + \log \frac{1}{\frac{\theta\varepsilon}{2} + \log \frac{1}{2-2\Delta}} + 2\log L) + \log \frac{1}{\delta} + 1).$$

Then when $n > n_0, \mathrm{Err}_{\mathcal{D}}(\mathcal{A}(\mathbf{z})) < \varepsilon$ with probability $1 - \delta$.

We follow the method of proving the ERM learnability of partial label learning Liu & Dietterich (2014) and the overall proof is in the appendix A.4. We define H_ε as the set of hypotheses with error at least ε, i.e. $H_\varepsilon = \{h \in \mathcal{H} : Err_{\mathcal{D}}(h) \geq \varepsilon\}$. Our target is to bound the H_ε, which ensures the convergence of the learner h. Since the entire data distribution is inaccessible, H_ε is evaluated by the mediator $R_{n,\varepsilon}$ as follows:

$$R_{n,\varepsilon} = \left\{\mathbf{z} \in (\mathcal{X} \times \mathcal{Y})^n : \exists h \in H_\varepsilon, Err^f_{\mathbf{z}}(h) = 0\right\}.$$

Then, our goal is to show that $\Pr(R_{n,\varepsilon} \mid f \in H_\Delta) \leq \delta$. In other words, the predictive model h, which is trained with the preprocessed noisy label distribution, has the generalization bound δ. Essentially, it should be clarified that the label set $\Omega^f_k(\boldsymbol{x})$ for instance $\boldsymbol{x}$ in our proof is induced by the label enhancement model f instead of artificial induction, so the true label may not be included in $\Omega^f_k(\boldsymbol{x})$, which is different from Liu & Dietterich (2014).

ERM here can be seen as picking out the most confident label from the top-k label set induced by f. Compared to directly optimizing the discrete loss function $Err^f_{\mathbf{z}}(h)$, many surrogate loss functions are proposed, and reweighting is also a common method. Otherwise, our predictive model preprocesses the protective label distribution and refines potential labels with different confidence generated by h. Therefore, the strategy of our predictive model is also an ERM learner.

Since the distribution $\mathcal{D}$ is unknown, it is very difficult to directly calculate the conditional probability $\Pr(R_{n,\varepsilon} \mid f \in H_\Delta)$. We bound it by introducing a testing set $\mathbf{z}'$. The overall proof can be divided into two parts. Lemma 1 is used in many learnability proofs.

Lemma 1 *For a testing set $\mathbf{z}' \in (\mathcal{X} \times \mathcal{Y})^n$, we can define the set $S_{n,\varepsilon}$ as*

$$S_{n,\varepsilon} = \left\{(\mathbf{z}, \mathbf{z}') \in (\mathcal{X} \times \mathcal{Y})^{2n} : \exists h \in H_\varepsilon, \mathrm{Err}^f_{\mathbf{z}}(h) = 0, \mathrm{Err}^f_{\mathbf{z}'}(h) \geq \frac{\varepsilon}{2}\right\}.$$

Then $\Pr((\mathbf{z}, \mathbf{z}') \in S_{n,\varepsilon} \mid f \in H_\Delta) \geq \frac{1}{2} \Pr(\mathbf{z} \in R_{n,\varepsilon} \mid f \in H_\Delta)$ for $n > \frac{2\log 4}{\varepsilon^2}$.

By lemma 1, the estimation on $R_{n,\varepsilon}$ can be turned to estimation on $S_{n,\varepsilon}$. It seems more complicated but we can swap training/testing instance pairs, which is a classic method in the proof of learnability, to refine the distribution on $\mathcal{D}$ into a single instance.

Lemma 2 *On the same condition of theorem 2. If the hypothesis space $\mathcal{H}$ has Natarajan dimension $d_{\mathcal{H}}$, $\gamma < 1$ and $\Delta < 1$, then*

$$\Pr\left(S_{n,\varepsilon} \mid f \in H_\Delta\right) \leq (2n)^{d_{\mathcal{H}}} L^{2d_{\mathcal{H}}} \exp\left(-\frac{n\theta\varepsilon}{2}\right).$$

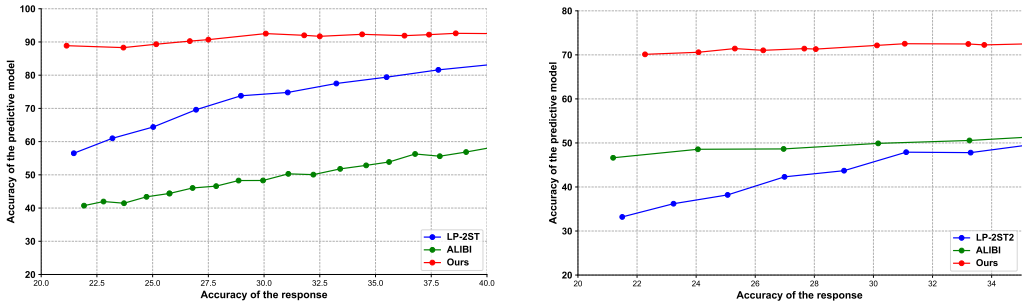
The proofs of Lemma 1 and 2 can be found in the appendix A.2, A.3. So far, with lemma 1 and lemma 2 we can prove theorem 2 with a bit of tricks, which is also detailed in the appendix A.4.

In this section, we prove two essential properties of the label distribution generated by the PLE model f. Classifier-consistency guarantees the effectiveness of the label distribution in a broad view and ERM learnability provides a generalization bound for the predictive model h and the convergence rate of the bound. In addition, Theorem 2 gives sufficient conditions which ensure the predictive model can learn from the label distribution. These conditions are instructive for the design of the protective label distribution.

5 Experiments

5.1 Experiment Setup

Evaluation measures In traditional label DP experiments, algorithms are compared based on the same privacy budget. However, our method cannot calculate privacy budget beacuse the noise in the



(a) (b)

Figure 2: The Privacy-Utility curve on CIFAR-10 (a) and CIFAR-100 (b).

Table 1: The accuracy of the response with certain privacy budgets in CIFAR-10.

ε	1	1.2	1.4	1.6	1.8	2	8
RRWithPrior	23.20	26.95	31.06	35.50	40.20	45.09	99.70
ALIBI	35.61	42.51	49.62	56.20	62.51	68.36	99.97

distribution is uncertain. Therefore, we review the original intention of privacy protection: *privacy* and *utility*. 1) *Privacy*: we use the *accuracy* of the response, a more accurate and intuitive indicator, to measure the degree of privacy protection. 2) *Utility*: the predictive model is trained with the data collected from the clients and better performance means more utility. In fact, more privacy means more dishonesty of the response and more utility means the response is more statistically consistent with the true label. Actually, the accuracy of the response is consistent with the privacy budget in protecting privacy as illustrated in Table 1. RRWithPrior Ghazi et al. (2021) and ALIBI Malek Esmaeili et al. (2021) employ randomized response and additive noise respectively to protect label privacy. It seems that measuring by accuracy is more intuitive and fairer.

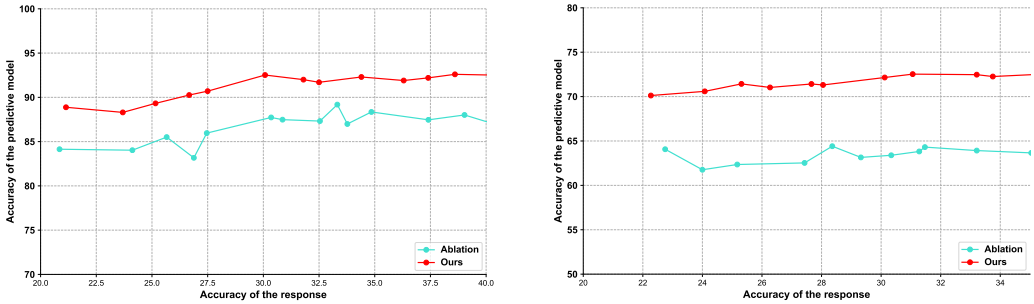
The predictive model is trained with the modified labels and aims to distinguish the true label. Top-1 accuracy is widely employed to measure the effectiveness of the predictive model. In summary, the fine strategy should have low accuracy of the response (*privacy*) while the predictive model has good performance (*utility*).

Datasets and Baselines We adopt two benchmark imgae datasets CIFAR-10 and CIFAR-100 Krizhevsky (2009), which is widely used in private machine learning tasks. These datasets are divided into training, validation, testing set in the ratio of 4:1:1. Our algorithm is compared to two state-of-the-art label DP baselines, LP-2ST Ghazi et al. (2021) and ALIBI Malek Esmaeili et al. (2021), where LP-2ST is trained on the response of RRWithPrior. It should be noted that Malek Esmaeili et al. (2021) also provides another algorithm PATE-FM which adapts the PATE (Private Aggregation of Teacher Ensembles) framework. As PATE cannot be deployed on the client and cost huge computational resources, it is not listed in our baselines. More implementation details can be found in the appendix A.5.

5.2 Data Accuracy

The trade-off hyperparameters in Eq.(4) play an important role in the privacy/utility trade-off of PLE. In detail, we observe the following:

- As mentioned in 3.2, we cannot take accuracy as the sole criterion to evaluate PLE. γ defined in (11) can be seen as the preliminary measure of the imbalance of the label. For the subsequent predictive model, the smaller γ could mean the better performance if the number of random labels is fixed.
- Both the punishment and the random labels are applied to decrease the top-1 accuracy within a reasonable range. For the simple dataset, large punishment are needed to protect



(a) (b)

Figure 3: Ablation study of the predictive model on CIFAR-10 (a) and CIFAR-100 (b).

privacy and for the complicated dataset like CIFAR-100, we enhance the level of random labels to reduce the effect of similar labels.

Since both baselines use the privacy budget as the measure, we calculate their accuracy of the response. The accuracy of RRWithPrior (LP-2ST) with privacy budget ε is $\frac{e^{\varepsilon}}{e^{\varepsilon}+n-1}$, where n is the dimension of the label. The accuracy of ALIBI is calculated by argmax the noisy response, which is provided in their code. Figure 2 demonstrates the tradeoff between accuracy and utility on CIFAR-10 and CIFAR-100. As is shown, our method outperforms all compared methods on CIFAR-10 and CIFAR-100. The improvements are significant, especially when the accuracy of the response is low. When the accuracy of the response on CIFAR-10 (CIFAR-100) is 21.14% (22.27%), the accuracy of the predictive model is 88.88% (70.12%), which is a great improvement than previous state-of-the-art. The non-private baseline (training with the true labels) is 95.49% (78.33%) on CIFAR-10 (CIFAR-100). The result shows great trade-off between privacy and utility.

5.3 Effectiveness of the Label Distribution

There could be concern that our method works due to the strategy of the predictive model instead of the protective label enchancement. This section we show that using vanilla corss-entropy loss can also achieve a moderate result. As shown in Figure 3, if we train the predictive model with corss-entropy loss, the accuracy merely drops 5% (8%) on CIFAR-10 (CIFAR-100). It is amazing because the accuracy of protective label distribution is very low, which validates the effectiveness of the protective label distribution. Moreover, the result also outperforms the state-of-the-art with the corss-entropy loss. On the other side, we supply the mean of the protective label distribution generated by the PLE model in the appendix A.6. It is observed that the true label is dominant in the label distribution. The protective label distribution protects instance-level label privacy while maintains statistical effectiveness, that is why the predictive model can learn from it.

6 Conclusion

In this work, we propose a novel strategy named Protective Label Enhancement (PLE) to mask the true label in the label distribution. The protective label distribution effectively protects label privacy and is also utility for the predictive model. Since the label distribution is noisy, we employ a regularization term and progressively update the label distribution to disambiguate the true label from the label distribution. Meanwhile, we prove that the predictive model is *classifier-consistent* and learning with the enhanced label distribution is ERM *learnable*. These properties provide theoretical guarantees for PLE. Experimental results also validate the excellence of our methods.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24392–24403, 2021.

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

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

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.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006b.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in Neural Information Processing Systems*, 33: 10948–10960, 2020.

Yi Gao and Min-Ling Zhang. Discriminative complementary-label learning with weighted loss. In *International Conference on Machine Learning*, pp. 3587–3597. PMLR, 2021.

Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28 (7):1734–1748, 2016.

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.

Andy Greenberg. Apple's "differential privacy" is about collecting your data—but not your data. *Wired, June*, 13, 2016.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pp. 2436–2444. PMLR, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *International Conference on Machine Learning*, pp. 1629–1637. PMLR, 2014.

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.

Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Stephen Shankland. How google tricks itself to protect chrome user privacy. *CNET, October*, 2014.

Michael T Smith, Mauricio A Álvarez, Max Zwiessele, and Neil D Lawrence. Differentially private regression with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1195–1203. PMLR, 2018.

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

Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *International Conference on Machine Learning*, pp. 24212–24225. PMLR, 2022.

Jing-Han Wu, Xuan Wu, Qing-Guo Chen, Yao Hu, and Min-Ling Zhang. Feature-induced manifold disambiguation for multi-view partial multi-label learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 557–565, 2020.

N. Xu, J. Shu, R. Zheng, X. Geng, D. Meng, and M. Zhang. Variational label enhancement. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–15, sep 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3203678.

Ning Xu, Yun-Peng Liu, and Xin Geng. Label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1632–1643, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xingyu Zhao, Yuexuan An, Ning Xu, and Xin Geng. Fusion label enhancement for multi-label learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 3773–3779. ijcai.org, 2022.

A APPENDIX

A.1 PROOF OF THEOREM 1

$$\begin{aligned}R(h) =&E_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}[\min_{l\in\Omega_k^f(\boldsymbol{x})}\ell(h(x),l)\cdot p(l\in\Omega_k^f(\boldsymbol{x})\mid f(\boldsymbol{x}))\cdot p(f(\boldsymbol{x})\mid\boldsymbol{x})]\\ =&E_{(\boldsymbol{x},y)\sim\mathcal{X}\times\mathcal{Y}}[\min_{l\in\Omega_k^f(\boldsymbol{x})}\{\ell(h(x),l)\cdot p(l\in\Omega_k^f(\boldsymbol{x})\mid y\in\Omega_k^f(\boldsymbol{x}))\cdot p(y\in\Omega_k^f(\boldsymbol{x})\mid f(x))\\ &+\ell(h(x),l)\cdot p(l\in\Omega_k^f(\boldsymbol{x})\mid y\notin\Omega_k^f(\boldsymbol{x}))\cdot p(y\notin\Omega_k^f(\boldsymbol{x})\mid f(x))\}\cdot p(f(\boldsymbol{x})\mid\boldsymbol{x})].\end{aligned}$$

The coefficient of $\ell(h(x),y)$ is:

$$\begin{aligned}\text{Coff}[\ell(h(x),y)] =&p(y\in\Omega_k^f(\boldsymbol{x})\mid y\in\Omega_k^f(\boldsymbol{x}))\cdot p(y\in\Omega_k^f(\boldsymbol{x})\mid f(x))\\ =&1-\Delta.\end{aligned}$$

For $l\neq y$, there is

$$\begin{aligned}\text{Coff}[\ell(h(x),l)] =&p(l\in\Omega_k^f(\boldsymbol{x})\mid y\in\Omega_k^f(\boldsymbol{x}))\cdot p(y\in\Omega_k^f(\boldsymbol{x})\mid f(x))\\ &+p(l\in\Omega_k^f(\boldsymbol{x})\mid y\notin\Omega_k^f(\boldsymbol{x}))\cdot p(y\notin\Omega_k^f(\boldsymbol{x})\mid f(x))\\ =&\gamma(1-\Delta)+\Delta.\end{aligned}$$

Therefore, when $1-\Delta>\gamma(1-\Delta)+\Delta$, i.e. $\gamma<1-\frac{\Delta}{1-\Delta}$, we have $h^*=\arg\min_{h\in\mathcal{H}}R(h)$.

A.2 PROOF OF LEMMA 1

Lemma 1 is a trick widely used in the proof of learnability. Consider the training set z, the testing set z' and each of them is of size n. We can bound $\Pr\left(z\in R_{n,\varepsilon}\mid f\in H_\Delta\right)$ with $\Pr\left((z,z')\in S_{n,\varepsilon}\mid f\in H_\Delta\right)$ as follows.

$$\begin{aligned}&\Pr\left((z,z')\in S_{n,\varepsilon}\mid f\in H_\Delta\right)\\ =&\Pr\left((z,z')\in S_{n,\varepsilon}\mid z\in R_{n,\varepsilon},f\in H_\Delta\right)\\ =&\Pr\left(\{\exists h\in H_\varepsilon\cap H(z),Err_{z'}(h)\geq\frac{\varepsilon}{2}\}\mid z\in R_{n,\varepsilon},f\in H_\Delta\right)\\ \geq&\Pr\left(h\in H_\varepsilon\cap H(z),Err_{z'}(h)\geq\frac{\varepsilon}{2}\mid z\in R_{n,\varepsilon},f\in H_\Delta\right)\\ \geq&1-\exp\left(-\cdot\frac{\varepsilon n}{8}\right)\end{aligned}$$

When $n>\frac{8\log 2}{\varepsilon}$, we have $\Pr\left((z,z')\in S_{n,\varepsilon}\mid f\in H_\Delta\right)\geq\frac{1}{2}\Pr\left(z\in R_{n,\varepsilon}\mid f\in H_\Delta\right)$, which completes the proof.

A.3 PROOF OF LEMMA 2

Here we need to bound $\Pr\left(S_{n,\varepsilon}\mid f\in H_\Delta\right)$. The key behind the proof is to refine $Err_z^s(h)$ and $Err_{z'}^s(h)$. We use a classic method, i.e. swap, to refine the single instance. A swap $\sigma(z,z')=(z^\sigma,z'^\sigma)$ means exchanging some instances between the training set z and testing set z' while keeping size n unchanged. There are 2^n different swaps in total and we define G as the set of all swaps. Firstly, we use swap to describe $\Pr\left(S_{n,\varepsilon}\mid f\in H_\Delta\right)$.

$$\begin{aligned}2^n\Pr\left(S_{n,\varepsilon}\mid f\in H_\Delta\right)=&\sum_{\sigma\in G}E\left[\Pr((z,z')\in S_{n,\varepsilon}\mid x,y,x',y',f\in H_\Delta)\right]\\ =&\sum_{\sigma\in G}E\left[\Pr(\sigma(z,z')\in S_{n,\varepsilon}\mid x,y,x',y',f\in H_\Delta)\right]\\ =&E\left[\sum_{\sigma\in G}\Pr(\sigma(z,z')\in S_{n,\varepsilon}\mid x,y,x',y',f\in H_\Delta)\right].\end{aligned}$$

To further refine $S_{n,\varepsilon}$, we define $S^h_{n,\varepsilon}$ for a certain classifier h as

$$S^h_{n,\varepsilon} = \{(z, z') : Err^s_z(h) = 0, Err_{z'}(h) \geq \frac{\varepsilon}{2}\}.$$

Next, we have the bound

$$\sum_{\sigma\in G} \Pr(\sigma(z, z') \in S_{n,\varepsilon} \mid x, y, x', y', f \in H_\Delta) \leq \sum_{h\in H|(x,x')} \sum_{\sigma\in G} \Pr(\sigma(z, z') \in S^h_{n,\varepsilon} \mid x, y, x', y', f \in H_\Delta).$$

By Natarajan (1989), the hypotheses space $H \mid (x, x')$ can be bounded as

$$|H \mid (x, x')| \leq (2n)^{d_H} L^{2d_H}.$$

Then,

$$\begin{aligned}
&\Pr\left(\sigma(z, z') \in S^h_{n,\varepsilon} \mid x, y, x', y', f \in H_\Delta\right) \\
=&I\left(Err_{z'}\sigma(h) \geq \frac{\varepsilon}{2} \mid f \in H_\Delta\right) \cdot \Pr\left(h(x^\sigma_i) \in S^\sigma_i, 1 \leq i \leq n \mid x^\sigma, y^\sigma, f \in H_\Delta\right) \\
=&I\left(Err_{z'}\sigma(h) \geq \frac{\varepsilon}{2} \mid f \in H_\Delta\right) \cdot \prod_{i=1}^{n} \Pr\left(h(x^\sigma_i) \in S^\sigma_i \mid x^\sigma, y^\sigma, f \in H_\Delta\right).
\end{aligned}$$

For the pair of (x, y, x', y'), we consider the number of instances of all cases. Specifically, let u_1, u_2 and u_3 represent the number of both incorrectly predicted instances, one incorrectly predicted instances and both correctly predicted instances. Besides, we define u_σ as the number of instances where (x^σ, y^σ) is incorrectly predicted while (x'^σ, y'^σ) is correctly predicted. Afterwards, the number of incorrectly predicted instances in the tesging set is $u_1 + u_2 - u_\sigma$.

$$\begin{aligned}
I\left(Err_{z'}\sigma(h) \geq \frac{\varepsilon}{2} \mid f \in H_\Delta\right) =& I(u_1 + u_2 - u_\sigma \geq \frac{\varepsilon}{2}n) \\
\leq& I(u_1 + u_2 \geq \frac{\varepsilon}{2}n).
\end{aligned}$$

For $\Pr\left(h(x^\sigma_i) \in S^\sigma_i \mid x^\sigma, y^\sigma, f \in H_\Delta\right)$, we count instances which have been swapped. On the one side, there are $u_2 + u_3 - u_\sigma$ instances satisfying $h(x^\sigma_i) = y^\sigma_i$ where we have $\Pr\left(h(x^\sigma_i) \in S^\sigma_i \mid f \in H_\Delta\right) = 1 - \Delta$. On the other side, there are $u_1 + u_\sigma$ instances satisfying $h(x^\sigma_i) \neq y^\sigma_i$ where we have $\Pr\left(h(x^\sigma_i) \in S^\sigma_i \mid f \in H_\Delta\right) \leq \gamma$. So, for any i, we have

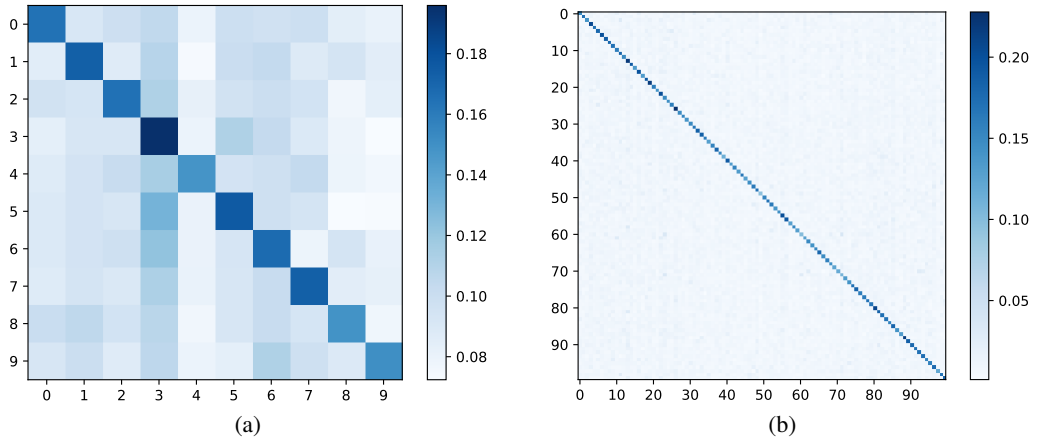
$$\Pr\left(h(x^\sigma_i) \in S^\sigma_i \mid x^\sigma, y^\sigma, f \in H_\Delta\right) \leq (1 - \Delta)^{u_2+u_3-u_\sigma} \cdot \gamma^{u_1+u_\sigma}.$$

And then, the conditional probability can be bounded as follow:

$$\Pr\left(\sigma(z, z') \in S^h_{n,\varepsilon} \mid x, y, x', y'\right) \leq I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right)(1 - \Delta)^{u_2+u_3-u_\sigma} \cdot \gamma^{u_1+u_\sigma}.$$

There are 2^n different swaps and we sum all.

$$\begin{aligned}
&\sum_{\sigma\in G} I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right)(1 - \Delta)^{u_2+u_3-u_\sigma} \cdot \gamma^{u_1+u_\sigma} \\
\leq& 2^{u_1+u_3} \cdot I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right) \cdot \sum_{j=0}^{u_2} \binom{u_2}{j} (1 - \Delta)^{u_2+u_3-j} \cdot \gamma^{u_1+j} \\
=& 2^{n-u_2} \cdot (1 - \Delta)^{u_2+u_3} \cdot \gamma^{u_1} \cdot I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right) \cdot \sum_{j=0}^{u_2} \binom{u_2}{j} (\frac{\gamma}{1 - \Delta})^j \\
=& 2^{n-u_2} \cdot (1 - \Delta)^{n-u_1} \cdot \gamma^{u_1} \cdot I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right) \cdot (1 + \frac{\gamma}{1 - \Delta})^{u_2} \\
=& I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right) \cdot 2^{n-u_2} \cdot (1 - \Delta)^{n-u_1} \cdot \gamma^{u_1} \cdot (\frac{1 - \Delta + \gamma}{1 - \Delta})^{u_2} \\
=& I\left(u_1 + u_2 \geq \frac{\varepsilon}{2}n\right) \cdot (2 - 2\Delta)^n \cdot (\frac{\gamma}{1 - \Delta})^{u_1} \cdot (\frac{1 - \Delta + \gamma}{2(1 - \Delta)})^{u_2}.
\end{aligned}$$



(a) (b)

Figure 4: Mean of the Protective Label Distribution on CIFAR-10 (a) and CIFAR-100 (b).

where $n = u_1 + u_2 + u_3$. The $2^{u_1+u_3}$ in the first step means ways of swapping both correct instances and both incorrect instances. The index of j equals u_2 different swaps of one correctly predicted instances. According to the assumption $\Delta + \gamma \leq 1$, we have $0 < \frac{\gamma}{1-\Delta} < \frac{1-\Delta+\gamma}{2(1-\Delta)} < 1$. For $u_1 + u_2 \geq \frac{\varepsilon}{2}n$, when $u_1 = 0$ and $u_2 = \frac{\varepsilon}{2}n$, the right side reaches its maximum as follows:

$$\Pr\left(S_{n,\varepsilon} \mid f \in H_\Delta\right) \leq (2n)^{d_H} \cdot L^{2d_H} \cdot (2-2\Delta)^n \cdot \left(\frac{1-\Delta+\gamma}{2(1-\Delta)}\right)^{\frac{n\varepsilon}{2}}.$$

We have proved the Lemma 2.

A.4 Proof of Theorem 2

With Lemma 1 and Lemma 2, we have

$$\Pr\left(R_{n,\varepsilon} \mid f \in H_\Delta\right) \leq 2^{d_H+1} \cdot n^{d_H} \cdot L^{2d_H} \cdot (2-2\Delta)^n \cdot \left(\frac{1-\Delta+\gamma}{2(1-\Delta)}\right)^{\frac{n\varepsilon}{2}}.$$

We set θ as

$$\theta = \log \frac{2(1-\Delta)}{1-\Delta+\gamma}.$$

Since $\Delta + \gamma < 1$, we get $\theta > 0$. We need to bound $\Pr\left(R_{n,\varepsilon} \mid f \in H_\Delta\right)$ with δ, which means

$$(d_H+1) \cdot \log 2 + d_H \log n + 2d_H \log L + n \log(2-2\Delta) - \frac{\theta\varepsilon n}{2} \leq \log \delta.$$

Note that the function $f(x) = log\frac{1}{a} + ax - logx - 1 \geq 0$. Let $a = \frac{\frac{\theta\varepsilon}{2} + log(\frac{1}{2-2\Delta})}{d_H}$ and $x = n$. It can be inferred that

$$logn \leq \frac{\frac{\theta\varepsilon}{2} + log(\frac{1}{2-2\Delta})}{d_H} n - log\frac{\frac{\theta\varepsilon}{2} + log(\frac{1}{2-2\Delta})}{d_H} - 1.$$

With the bound of $logn$, we get the linear inequality of n. Let

$$n_0(\mathcal{H}, \varepsilon, \delta) = \frac{2}{\frac{\theta\varepsilon}{2} + \log\frac{1}{2-2\Delta}} \left(d_{\mathcal{H}}(\log(2d_{\mathcal{H}}) + \log\frac{1}{\frac{\theta\varepsilon}{2} + \log\frac{1}{2-2\Delta}} + 2\log L) + \log\frac{1}{\delta} + 1\right).$$

When $n > n_0$, we get $\Pr\left(R_{n,\varepsilon} \mid f \in H_\Delta\right) < \delta$ and the proof is finished.

A.5 Implementation Details

For the fairness of the experiments, all predictive models use WideResNet28×2 architecture Zagoruyko & Komodakis (2016) on each dataset. The hyperparameters of the compared models

are set as the author recommends. Two baselines and our method all employ data augmentation on the predictive model. In our proposed method, we use mini-batch SGD Robbins & Monro (1951) with a batch size of 128 and a momentum of 0.9. Otherwise, each model is trained with maximum epochs $T = 300$ and employs early stopping strategy with patience 20. In other words, if the accuracy does not rise in validation set for 20 epochs, tche training process will be stopped. Finally, we report final performance using the test accuracy corresponding to the best accuracy on validation set. The PLE model is only trained for no more than 20 epochs without data augmentation. The learning rate and weight decay for PLE are both 0.001 while for the predictive model, they are 0.05 and 0.001 separately. Usually, the number of the random labels k_1 is set as 3 and the number of non-zeroized labels k_2 is set to 4 or 5. The punishment factor α_1 ranges from 0 to 0.4 and the compensation factor α_2 ranges from 0.9 to 1.3. The weight of the random labels α_3 is from 1.6 to 2.3. Different settings will result in different privacy and utility, and we will release the hyperparameters of the experiments in our code for public access.

A.6 Effectiveness of the Protective Label Distribution

As illustrated in Figure 4, the horizontal axis represents the true label while the vertical axis represents the mean of the label distribution. The diagonal can be seen as the degree of correctly predicted labels. We can see that the true label is dominant in the label distribution, that is why the label distribution is effective. On the other side, the figure can be seen as a simple measure of the similarity between labels.