Xinyu Tang*, Milad Nasr, Saeed Mahloujifar, Virat Shejwalkar, Liwei Song, Amir Houmansadr, and Prateek Mittal

Machine Learning with Differentially Private Labels: Mechanisms and Frameworks

Abstract: Label differential privacy is a relaxation of differential privacy for machine learning scenarios where the labels are the only sensitive information that need to be protected in the training data. For example, imagine a survey from a participant in a university class about their vaccination status. Some attributes of the students are publicly available but their vaccination status is sensitive information and must remain private. Now if we want to train a model that predicts whether a student has received vaccination using only their public information, we can use label-DP. Recent works on label-DP use different ways of adding noise to the labels in order to obtain label-DP models. In this work, we present novel techniques for training models with label-DP guarantees by leveraging unsupervised learning and semi-supervised learning, enabling us to inject less noise while obtaining the same privacy, therefore achieving a better utility-privacy trade-off. We first introduce a framework that starts with an unsupervised classifier f_0 and dataset D with noisy label set Y, reduces the noise in Y using f_0, and then trains a new model f using the less noisy dataset. Our noise reduction strategy uses the model f_0 to remove the noisy labels that are incorrect with high probability. Then we use semi-supervised learning to train a model using the remaining labels. We instantiate this framework with multiple ways of obtaining the noisy labels and also the base classifier. As an alternative way to reduce the noise, we explore the effect of using unsupervised learning: we only add noise to a majority voting step for associating the learned clusters with a cluster label (as opposed to adding noise to individual labels); the reduced sensitivity enables us to add less noise. Our experiments show that these techniques can significantly outperform the prior works on label-DP.

Keywords: Differential Privacy; Label Differential Privacy

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

***Corresponding Author: Xinyu Tang:** Princeton University, E-mail: xinyut@princeton.edu

Saeed Mahloujifar, Liwei Song, Prateek Mittal: Princeton University, E-mail: {sfar, liweis, pmittal}@princeton.edu
Milad Nasr, Virat Shejwalkar, Amir Houmansadr: University of Massachusetts Amherst, E-mail: {milad, shejwalkar, amir}@cs.umass.edu

1 Introduction

With emerging applications of machine learning (ML), one of the important requirements of ML models is to preserve privacy of datasets used to train them. Differential privacy (DP) [16, 17] has become the gold standard of privacy where the training algorithm is required to output a model that will not be too different from the output of the same algorithm on a neighboring dataset (i.e., the value in one of its elements is swapped). This DP definition establishes a strong requirement that preserves the privacy of each individual example in the training set. However, the existing techniques that provably satisfy DP often suffer from a drop in utility compared to training algorithms without privacy. As a result, researchers study alternative definitions of privacy.

One recently proposed definition of privacy for machine learning is *label-DP* [23], which is also one kind of AttributeDP [29]. In label-DP, the privacy constraint is only imposed on the label set. That is, for a feature set X and label set Y, the ML algorithm L will output a model f which satisfies label-DP if $f(X, Y) \approx f(X, Y')$, where Y' is a neighboring label set. This definition is motivated by applications where the feature sets are publicly available and the labels are the only thing that have to be protected. For instance, the features could be the attributes of users of an online social network and the labels be if the users suffer from mental distress. Using (regular) DP for such a scenario may cause a drop in utility because DP will provide privacy for the features as well. For example, imagine a survey from a participant in a university class about their vaccination status. Some attributes of the students are publicly available but their vaccination status is sensitive information and must remain private. Now if we want to train a model to predict whether a student has received vaccination us-

ing only their public information, we can use label-DP because other attributes are already public and trying to protect their privacy will cause an unnecessary drop in utility.

As the main incentive for choosing label-DP over DP is improving the utility of the trained models, it is important to use the most recent advancements in ML for such problems. Since prior works [23, 35] for label-DP focus on noise addition algorithm design instead of how to improve the utility for label-DP, we explore classification algorithms that succeed with imperfect labels. In particular, the main question in this work is:

> How to optimize utility with label-DP, using classification algorithms that work with imperfect labels?

Specifically, we consider *unsupervised learning* and *semi-supervised learning* as ways for obtaining improved trade-offs between utility and privacy.

1.1 Our Contributions

Using unsupervised learning to add less noise and reduce the effect of noise: Because unsupervised learning does not use labels for training, it can be leveraged in the label-DP setting without incurring additional label privacy loss. The noise is only added when performing majority voting on the non-private labels in a cluster to assign a cluster label, and therefore the sensitivity is much reduced (ℓ_2 sensitivity is only $\sqrt{2}$ regardless of set size: changing one of the labels, the number of votes on one class increases by 1 and it decreases by 1 on the other class). We name this technique *NoiseCluster*, which would be effective when ε is extremely low as a result of much reduced sensitivity.

Another alternative way to leverage unsupervised learning is to generate cluster labels, which can be used as a component of *DenoiseSSL* (explained later) used to denoise any differentially private label set.

Using semi-supervised learning to reduce the effect of noise: Most of existing techniques proposed for label-DP use label noise. The label-DP mechanism (e.g., randomized response) adds noise to the labels and releases the features with the noisy labels. Then, using post-processing arguments, any model trained using standard learning algorithms on this data also enjoys label-DP. Leveraging semi-supervised learning (SSL), we propose a framework that performs a *label denoising* step on the dataset before applying the learning algorithm. Our denoising algorithm, *DenoiseSSL*, leverages the cluster label set from unsupervised learning to identify the labels that are correct with high probability and removes other labels from the label set. Then it applies SSL to train the model with the partial set of labels.

A comprehensive comparison with existing techniques: We instantiate our framework using noise mechanisms proposed in PATE-FM [35] and randomized response [23, 56] as well as our proposed techniques which enables less injected noise, and perform comprehensive experiments on mainstream datasets such as CIFAR10, CIFAR100, and CINIC10. Our evaluations show that our framework can significantly improve the utility of obtained models for the entire (reported) range of ε.

1.2 Technical Overview

Label differential privacy is proposed as a relaxation of differential privacy for scenarios when we only need the labels to remain private. In order to leverage this relaxation, existing techniques for label-DP use the fact that we do not need to manipulate the features and only add noise to the labels. That is, given a labeled dataset (X, Y) they apply a DP mechanism M on labels and then they apply a ML algorithm L on $(X, M(Y))$ to get the final label-DP model f.

In this work, we take one step beyond this and leverage *unsupervised learning* and *semi-supervised learning*.

Denoising framework through unsupervised learning and semi-supervised learning (DenoiseSSL): *DenoiseSSL* purifies the noisy labels and converts them to a smaller set of labels but with higher precision. DenoiseSSL uses a classifier f_0 (we consider f_0 trained using *unsupervised learning*) to generate a pseudo label set without incurring additional privacy cost. We only include labels for samples which have the same noisy label and pseudo label, and our analysis shows that the average correctness of the remaining labels could be much higher than that of all noisy labels. After obtaining this partial, high quality set of labels, we can apply *semi-supervised learning* on all the instances and the partial set of labels. Our experiments show that this technique can achieve better accuracy compared to training with a complete set of noisy labels.

Adding less noise through unsupervised learning (NoiseCluster): When assigning the cluster label by majority voting over non-private labels in the *unsupervised learning* model, the sensitivity is $\sqrt{2}$; we then add Gaussian noise to the voting statistics based on this

to get the desired privacy level (NoiseCluster). No noise will be added to individual samples.

Summary of experimental results: Our framework consists of two parts: noise addition mechanisms and learning with the private labels. The noise addition mechanisms include existing noise mechanisms [23, 35] and our proposed NoiseCluster. To improve the utility for label-DP, in addition to DenoiseSSL, we have also investigated algorithms in learning with noisy labels area and leveraged Aug-Descent [39]. Our framework significantly outperforms previous works (LP1ST (+mixup) [23] and ALIBI [35]) on CIFAR10, CIFAR100 and CINIC10 datasets at $\varepsilon = 0.5, 1, 2, 3, 4, 6$. For example, our framework achieves 56.2% higher accuracy than LP1ST and 60.8% higher accuracy than ALIBI on CIFAR10 at $\varepsilon = 0.5$. We also make specific recommendations regarding our framework based on the value of ε and other factors in Section 6.

2 Background

In this section, we briefly introduce the background on machine learning, privacy leakages in machine learning models, differential privacy and deep learning with differential privacy.

2.1 Machine Learning

In this paper, we consider machine learning (ML) models used for classification tasks, and below, we review the major learning paradigms used to train these models.

Supervised Learning Let $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}^k$ be a ML classifier (e.g., neural network) with d input features and k classes, which is parameterized by θ. For a given example $\mathbf{z} = (\mathbf{x}, y)$, $f_\theta(\mathbf{x})$ is the classifier's confidence vector for k classes and the predicted label is the corresponding class which has the largest confidence score, i.e., $\hat{y} = \text{argmax}_i\, f_\theta(\mathbf{x})$.

The goal of supervised machine learning is to learn the relationship between features and labels in given *labeled* training data D^l_{tr} and generalize this ability to unseen data. The model learns this relationship using empirical risk minimization (ERM) on the training set D^l_{tr}, where the risk is measured in terms of certain loss function, e.g., cross-entropy loss:

$$\min_\theta \frac{1}{|D^l_{tr}|} \sum_{\mathbf{z} \in D^l_{tr}} l(f_\theta, \mathbf{z})\big)$$

Here $|D^l_{tr}|$ is the size of the labeled training set and $l(f_\theta, \mathbf{z})$ is the loss function. When clear from the context, we use f instead of f_θ, to denote the target model.

Semi-supervised Learning When the labeling process is expensive, semi-supervised learning can alleviate the dependence of ML on labeled data, which changes the problem setup by introducing a new unlabeled dataset, D^{ul}_{tr}. Generally, the unlabeled data is drawn from a similar distribution as the labeled data.

There is a long line of research on semi-supervised learning, but in this work, we consider the state-of-the-art semi-supervised learning algorithm, *FixMatch* [48], which is based on the concepts of *pseudo-labeling* [32] and *consistency regularization* [3]. FixMatch is shown to achieve state-of-the-art classification performances on image classification tasks with less than 1% (of total data) labeled data.

Unsupervised Learning Unsupervised learning aims to learn the inherent task-agnostic patterns in unlabeled training data, i.e., D^{ul}_{tr}. It has been shown that these patterns are very useful for various downstream tasks such as classification, object detection [9, 10]. Unsupervised learning usually relies on representative features in the training data to perform well. More specifically, unsupervised learning first learns an encoder E using D^{ul}_{tr} that outputs representation $E(\mathbf{x})$ of input $\mathbf{x}$. Then E can be fine-tuned in a task-specific fashion, e.g., fine-tuning E by pseudo labeling and confidence.

In this paper, we use the state-of-the-art unsupervised learning algorithm *SCAN* [52] which is based on *contrastive learning SimCLR* proposed by Chen et al. [9, 10]. The intuition behind contrastive learning (SimCLR [9, 10], MoCo [25]) is that the model outputs should be similar for two different augmented versions of the same input. Unsupervised learning algorithms, e.g., SCAN [52], use these representative features to achieve comparable accuracy as fully supervised learning.

Learning with Noisy Labels Learning with noisy labels is a long-standing problem in machine learning because the labels in training data usually contain noisy labels. Our work in the learning from noisy labels domain utilizes the refining strategies for the training data; below we elaborate them. Han et al. [24] propose *co-teaching* that allows one model to learn from the other model's most confident outputs. Based on co-teaching, Li et al. [34] propose *DivideMix*, which uses Gaussian Mixture Model (GMM) to obtain more confident samples and leverages augmented samples to refine the noisy labels. Nishi et al. [39] apply more advanced augmentation strategies called *Aug-Descent* on DivideMix. The

latter two methods are specifically designed for image datasets while co-teaching is a general learning from noisy label technique applicable to all data domain.

2.2 Privacy Leakages in ML Models

ML models generally require large amounts of training data to achieve good performances. This data can be of sensitive nature, e.g., medical records and personal photographs, and without proper precautions, ML models may leak sensitive information about their private training data. Multiple previous works have demonstrated this via various *inference* attacks, e.g., membership inference, property or attribute inference, model stealing, and model inversion. Below, we review these attacks.

Consider a target model f_θ trained on D_{tr} and a target sample $(\mathbf{x}, y)$. Membership inference attacks [2, 45, 46] aim to infer whether the target sample $(\mathbf{x}, y)$ was used to train the target model, i.e., whether $(\mathbf{x}, y) \in D_{tr}$. Property or attribute inference attacks [37, 50] aim to infer certain attributes of $(\mathbf{x}, y)$ based on model's inference time representation of $(\mathbf{x}, y)$. For instance, even if f_θ is just a gender classifier, $f_\theta(\mathbf{x})$ may reveal the race of the person in $\mathbf{x}$. Model stealing attacks [41, 51] aim to reconstruct the parameters θ of the original model f_θ based on black-box access to f_θ, i.e., using $f_\theta(\mathbf{x})$. Model inversion attacks [21] aim to reconstruct the whole training data D_{tr} based on white-box, i.e., using θ, or black-box, i.e., using $f_\theta(\mathbf{x})$, access to model.

2.3 Differential Privacy

Differential privacy [15, 17] is the gold standard for data privacy. It is formally defined as below:

Definition 1 (Differential Privacy)**.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ preserves (ε, δ)-differential privacy iff for any two neighboring datasets $D, D' \in \mathcal{D}$ and for any subset $S \subseteq \mathcal{R}$ we have:*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta \tag{1}$$

where ε is the privacy budget *and δ is the* failure probability.

Rényi Differential Privacy (RDP) is a commonly-used relaxed definition for differential privacy.

Definition 2 (Rényi Differential Privacy (RDP) [38])**.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ is (α, ε)-RDP with order $\alpha \in (1, \infty)$ iff for any two neighboring datasets $D, D' \in \mathcal{D}$:*

$$\begin{aligned} &D_\alpha(\mathcal{M}(D)||\mathcal{M}(D')) \\ :=&\frac{1}{\alpha - 1} \log \mathop{\mathbb{E}}_{\delta \sim \mathcal{M}(D')} [(\frac{Pr[\mathcal{M}(D) = \delta]}{Pr[\mathcal{M}(D') = \delta]})^\alpha] \leq \varepsilon \end{aligned} \tag{2}$$

Lemma 1 (Adaptive Composition of RDP [38])**.** *Consider two randomized mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ that provide (α, ε_1)-RDP and (α, ε_2)-RDP, respectively. Composing $\mathcal{M}_1$ and $\mathcal{M}_2$ results in a mechanism with $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.*

Lemma 2 (RDP to DP conversion [38])**.** *If $\mathcal{M}$ obeys (α, ε)-RDP, then $\mathcal{M}$ is $(\varepsilon + \log(\frac{1}{\delta})/(\alpha - 1), \delta)$-DP for all $\delta \in (0, 1)$.*

Lemma 3 (Post-processing of RDP [38])**.** *Given a randomized mechanism that is (α, ε)-RDP, applying a randomized mapping function on it does not increase its privacy budget, i.e., it will result in another (α, ε)-RDP mechanism.*

2.4 Deep Learning with Differential Privacy

Several works have used differential privacy in traditional machine learning to protect the privacy of the training data [5, 8, 20, 33, 60]. Many of these works [5, 8, 20] use properties such as convexity or smoothness for privacy analysis, which is not necessarily true in deep learning, therefore one cannot use many of such methods in practice. Here we briefly introduce DP-SGD [1] and PATE [42, 43] (See more in Section 7.1).

DP-SGD Abadi et al. [1] design a deep learning training algorithm, DP-SGD, where they use gradient clipping to limit the sensitivity, and then add noise to gradients proportional to its sensitivity.

PATE Private Aggregation of Teacher Ensembles [42, 43] (PATE) is a framework based on private knowledge aggregation of an ensemble model and knowledge transfer. PATE trains an ensemble of "teachers" on disjoint subsets of the private dataset. The ensemble's knowledge is then transferred to a "student" model via differentially private aggregation of the teachers' votes on samples from an unlabeled public dataset. Only the student model is released as the output of the training, as it accesses sensitive data via a privacy-preserving inter-

face. We use RDP to compute the bounds for PATE based framework and convert RDP to DP.

3 Problem Statement: Label Differential Privacy

As many services are using users' private data to train machine learning models, the use of privacy preserving machine learning with strong privacy guarantees is increasing. As we explained in Section 2.1, data $D = (X, Y)$ in machine learning have two main types of attributes: features X and labels Y. The research community has considered designing privacy frameworks [27, 30] which generalize differential privacy by restricting the secrets about individuals as well as enumerating the side information to the attacker. In certain practical settings, only the labels Y in training data are of sensitive nature, i.e., are private. Imagine a survey from a participant in a university class about their vaccination status. Some attributes of the students are publicly available but vaccination status is a sensitive information and must remain private. Now if we want to train a model that predicts whether a student has received vaccination only using their public information, we can use label-DP. To address this problem, we should only apply differential privacy to the labels. Such privacy protection is commonly called as *label differential privacy* (label-DP).

We start by formally defining label-DP and similarly label-Rényi DP.

Definition 3. *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ and range $\mathcal{R}$ preserves (ε, δ)-label-DP iff for any two label neighboring datasets $(X, Y), (X, Y') \in \mathcal{X} \times \mathcal{Y}$ and for any subset $S \subseteq \mathcal{R}$ we have:*

$$\Pr[\mathcal{M}(X, Y) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(X, Y') \in S] + \delta \qquad (3)$$

where ε is the privacy budget *and δ is the* failure probability*. In particular, when $\delta = 0$ $\mathcal{M}$ is ε-label-DP.*

Definition 4 (label-Rényi DP)**.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ is (α, ε)-label-RDP with order $\alpha \in (1, \infty)$ iff for any two label neighboring datasets $(X, Y), (X, Y') \in \mathcal{X} \times \mathcal{Y}$:*

$$\begin{aligned} &D_\alpha(\mathcal{M}(X, Y) || \mathcal{M}(X, Y')) \\ :=&\frac{1}{\alpha - 1} \log \mathop{\mathbb{E}}_{\delta \sim \mathcal{M}(X,Y')} [(\frac{Pr[\mathcal{M}(X, Y) = \delta]}{Pr[\mathcal{M}(X, Y') = \delta]})^\alpha] \leq \varepsilon \end{aligned} \qquad (4)$$

We can also use existing differential privacy mechanisms to achieve label-DP as proposed below.

Proposition 1. *A mechanism $\mathcal{M}$ preserves (ε, δ)-label-DP if it also preserves (ε, δ)-DP.*

Proof. The notion of neighboring dataset in label-DP requires that datasets differ in one label. This implies that the pair datasets also satisfy the notion of neighboring where two datasets differ in one example. Therefore, if a mechanisms satisfies (ϵ, δ)-DP, then it also satisfies (ε, δ)-label-DP. □

Proposition 2. *If mechanism $\mathcal{M}'$ with domain $\mathcal{Y}$ and range $\mathcal{Y}$ satisfies (ε, δ)-DP (with replacement), then a mechanism M with domain $\mathcal{X} \times \mathcal{Y}$ that given (X, Y) outputs $f(X, M'(Y))$, for an arbitrary function f, satisfies (ε, δ)-label-DP.*

Proof. We first show that a mechanism M'' that outputs $(X, M'(Y))$ is (ϵ, δ)-label-DP. For any neighboring datasets X, Y and (X, Y') we have

$$\Pr[M''(X, Y) \in \mathcal{A}] = \Pr[(X, M'(Y)) \in \mathcal{A}].$$

Let $\mathcal{B} = \{Y''; (X, Y'') \in \mathcal{A}\}$. We have

$$\Pr[(X, M'(Y)) \in \mathcal{A}] = \Pr[M'(Y) \in \mathcal{B}]$$

Therefore we have

$$\begin{aligned} \Pr[M''(X, Y) \in \mathcal{A}] &= \Pr[M'(Y) \in \mathcal{B}] \\ &\leq e^{\varepsilon} \Pr[M'(Y') \in \mathcal{B}] + \delta \\ &= e^{\varepsilon} \Pr[M''(X, Y') \in \mathcal{A}] + \delta. \end{aligned}$$

Now since M'' is (ϵ, δ)-label-DP, M is also (ϵ, δ)-label-DP by a post-processing argument. □

Remark 1. *Note that the mechanism $\mathcal{M}'$ could be any differentially private mechanism. This includes mechanisms that use sub-sampling and shuffling for amplification of privacy. However, these kind of mechanisms that change the order of labels are not useful for label-DP. This is because we need the association of the labels and features to be preserved and changing the order of labels will remove this association.*

The next step is to design a ML mechanism which preserves label-DP. A basic idea is to use Proposition 1 and use similar mechanisms as used to train ML models with differential privacy [1]. However, label-DP is much more relaxed form of privacy and we can leverage the constraints in label-DP to design new methods and improve the utility of existing (traditional) DP approaches. In this work, we use Proposition 2 to provide label-DP.

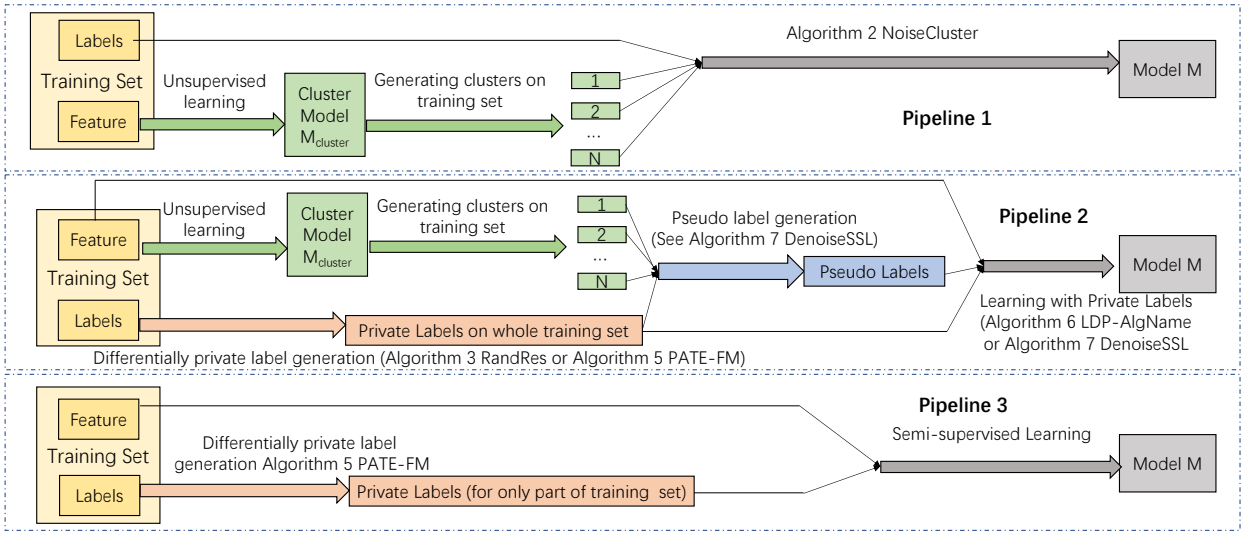

Fig. 1. Our end-to-end framework consists of three pipelines. Pipeline 1 leverages unsupervised learning to learn clusters and only adds noise when assigning cluster label with no further learning needed. Pipeline 2 will first generate the DP labels, then apply learning algorithms on the DP labels. Pipeline 3 leverages a variation of PATE by only querying part of training set instead of the whole set. Yellow boxes show the features and non-private labels. The green arrows show the computation related to unsupervised learning. The orange arrow shows the DP label generation process. Blue arrows show the pseudo label generation process. Gray arrows show the computation which finally outputs the models. Table 1 summarizes all algorithms in our framework.

In particular, we divide approaches to ML with label-DP into two components: (1) *obtain the labels with differential privacy* and then (2) *learn from the noisy (differentially private) labels*. In the following sections, we overview and compare with existing works on label-DP, provide our algorithms for each of the aforementioned label-DP components, and finally, discuss the advantages and disadvantages of each of the algorithms.

Overview and comparison with related works

We propose a two-component modular framework to achieve the label-DP described above. The first component involves obtaining noisy (differentially private) labels and the second component involves learning on the differentially private labels. For the first component of obtaining differentially private labels, in addition to existing works including Randomized Response [56] based approach by Ghazi et al. [23] and PATE [42, 43] based approach by Malek et al. [35], we also design one more approach. This is an unsupervised learning based approach for generating differentially private labels (NoiseCluster), which adds noise to the counts of labels when assigning cluster labels, thus adding less noise compared to existing works (NoiseCluster does not need further learning with the private labels). For the second component of learning with differentially private labels, we focus on how to *denoise* the differentially private labels to improve the utility of the resulting models.[1] Unlike previous work ALIBI [35], which focuses on denoising the differentially private labels using only the label information, we also consider how to leverage the public feature information to denoise the private labels and therefore utility is much improved. Techniques in our framework include unsupervised learning, semi-supervised learning, and learning with noise labels. We systematically evaluate our framework and show the much improved utility compared to existing works.

4 Our Framework

In this section, we present our framework in detail. Specifically, there are three building blocks: NoiseCluster (i.e., adding noise to the cluster label, see Section 4.1), differentially private labels generation (i.e., adding noise to individual labels, see Section 4.2), and learning with differentially private labels (i.e., how to denoise private labels to improve utility, see Section 4.3). Our framework consists of three pipelines. Pipeline 1

1 We later noticed that Ghazi et al. [23] also investigated using self-supervised learning to improve utility of their LPMST algorithm. Our framework focuses on using unsupervised learning and semi-supervised learning to add less noise as well as denoising the differentially private labels.

Table 1. Summary of algorithms in our framework.

Algorithm name	Function	Intuitive explanation
Algorithm 1 LabelCount	Count the samples per class for each cluster in unsupervised learning model.	Only do counting, no noise added.
Algorithm 2 NoiseCluster	Generate differentially private label set by unsupervised learning.	Add Gaussian noise to the class count for each cluster and do majority voting, no noise added to individual samples.
Algorithm 3 RandRes	Generate differentially private label set directly from the label set.	Probabilistically replace the true label with a random label.
Algorithm 4 Confident-GNMax	Generate the differentially private label for a queried sample given n teacher models trained by PATE [42, 43].	Add Gaussian noise to the counting of teacher model's predictions and return the label when the noisy voting above a threshold.
Algorithm 5 PATE-FM	Generate differentially private label set given n teacher models by PATE [42, 43].	Use Confident-GNMax to generate either full or partial differential private label set.
Algorithm 6 LDP-AlgName	Given the differentially private labels, train the models using AlgName.	General framework, we investigate LDP-SSL and LDP-LNL (specifically LDP-FixMatch [48] and LDP-AugDescent [39]).
Algorithm 7 DenoiseSSL	Given the differentially private labels, train the model with purified label set by semi-supervised learning.	Filter the noisy label set by using pseudo labels from unsupervised learning to a smaller subset but with higher precision, then apply semi-supervised learning.

is based solely on NoiseCluster, Pipeline 2 is based on the combination of differentially private labels generation and learning with differentially private labels, and Pipeline 3 is a variation of PATE in the differentially private label generation block followed by semi-supervised learning. Figure 1 presents our end-to-end framework. Next we detail our three building blocks. We also summarize the algorithms of our framework in Table 1. For brevity, we assume there are N classes in total, i.e., the label cardinality is N, and denote $[N] = \{0, 1, ..., N-1\}$. We denote Y as clean label set, $\widetilde{Y}$ as noisy label set and $\hat{Y}$ as cluster label set from unsupervised learning model.

4.1 NoiseCluster

An easy approach to protect privacy of the private labels is to not use them at all. To this end, there are several unsupervised approaches that can be used to cluster the input space. Now we can apply a simple differentially private voting, called NoiseCluster to assign labels to the clusters with small privacy cost (See Algorithm 2). While this approach has a very low privacy cost, unfortunately, the utility of the clustering methods is significantly lower compared to the supervised methods. In Section 5, we show this trade-off. Specifically, we use SCAN [52] is this paper.[2]

Algorithm 1 LabelCount: count the samples per class for each cluster in unsupervised learning model

Require: dataset $D = (X, Y)$ or $(X, \widetilde{Y})$, an unsupervised model f_0 on X with N clusters.
  $D' = \{\}$
  for all $(x, y) \in D$ **do**
    $\hat{y} \leftarrow \text{argmax}_i\, f(x)$
    Add $(x, y, \hat{y})$ to D'
  end for
  $\hat{Y}$ is a $N \times N$ matrix with all values initialized as 0.
  for all $l \in [N]$ **do**
    $D_l = \{(x, y, \hat{y}) \colon (x, y, \hat{y}) \in D', \hat{y} = l\}$
    for all $(x, y, \hat{y}) \in D_l$ **do**
      $\hat{Y}[l, y] = \hat{Y}[l, y] + 1$
    end for
  end for
  return $(D', \hat{Y})$

Algorithm 2 NoiseCluster

Require: dataset $D = (X, Y)$, noise parameters σ.
  Train unsupervised model f_0 on X with N clusters
  Run $(D', \hat{Y}) \leftarrow$ Algorithm 1$((X, Y),\, f_0)$
  for all $l \in [N]$ **do**
    Assign cluster l with $\arg\max_i \{\hat{Y}[l, :] + \mathcal{N}(0, \sigma)\}$
  end for
  return model f with assigned clusters

2 Please note that this procedure is not limited to any specific unsupervised learning. We used SCAN [52], which was the state-of-the-art unsupervised learning algorithm (at the time of

Algorithm 3 Randomized Response Mechanism

Require: L the true label, N the label cardinality, and $1-p$ the probability of randomized response.
$z \sim$ Uniform(0,1)
if $z < p$ **then**
  return L
end if
return Uniform([N]\L)

Theorem 1. *Algorithm 2 is (ε, δ)-label-DP where:*

$$\varepsilon = \frac{2\sqrt{\ln(1.25/\delta)}}{\sigma} \tag{5}$$

Proof. The ℓ_2 sensitivity of $\hat{Y}$ in NoiseCluster is $\sqrt{(1)^2 + (-1)^2} = \sqrt{2}$ because by changing one of the labels, the number of votes on one class increases by 1 and it decreases by 1 on the other class. Therefore by Theorem 1 in Balle et al. [4] we conclude that $\hat{Y} + \mathcal{N}(0, \sigma)$ is (ε, δ)-DP. Then using a post-processing argument, argmax $\hat{Y}$ is (ε, δ)-DP as well. □

4.2 Differentially Private Labels

If we can get labels with differential privacy, then using Proposition 2 we can use normal machine learning methods on the differentially private labels and achieve label-DP. In this section we introduce two main approaches for obtaining differentially private labels.

4.2.1 RandRes

One of the main approaches to achieve differential privacy is to add noise to the data. Differential privacy usually works well on mechanisms that aggregate several data points, however, supervised learning needs the label of each instance. Therefore, we cannot easily use aggregation mechanisms. Instead, we can use Randomized Response mechanisms [56] (RandRes) to preserve the differential privacy and provide plausible deniability.

In a nutshell, RandRes for label-DP proposed by Ghazi et al. [23] returns the true label with probability p and a uniformly random label other than true label with probability $1-p$ (See Algorithm 3).

Theorem 2 (Randomized Response Mechanism)**.** *Algorithm 3 with label cardinality N and probability of true answer $p > 1/N$ satisfies $(\varepsilon, 0)$-label-DP where:*

$$\varepsilon = \ln\left(\frac{(N-1)p}{1-p}\right). \tag{6}$$

Proof. For any label y, we know that $\Pr[M(y) = y] = p$. Therefore if $p > 1/N$, we have: $\forall y', y'' \neq y$,

$$\frac{\Pr[M(y) = y']}{\Pr[M(y'') = y']} \leq \frac{\Pr[M(y) = y]}{\Pr[M(y'') = y]} = \frac{p(N-1)}{(1-p)} \tag{7}$$

□

4.2.2 PATE-FM

As mentioned earlier most of the DP mechanisms work best when applied on an aggregation of several data points. To take advantage of this, we can use the idea of ensemble learning [14]. The approach is similar to PATE [42, 43] approach where we utilize the noisy aggregation with differential privacy. However, unlike PATE [42, 43], we only need to provide differential privacy for the labels. Based on this observation, Malek et al. [35] propose PATE-FM. Figure 2 illustrates PATE-FM. Below, we describe each component of this method.

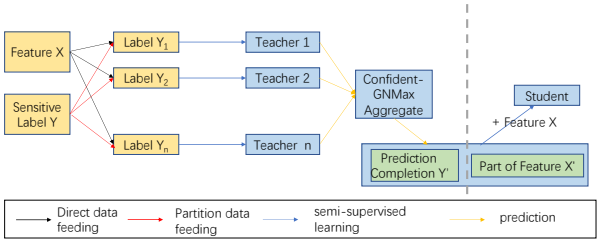

Fig. 2. PATE-FM: adaption of PATE for label-DP.

Data partitioning and training teachers: Consider a dataset $D = (X, Y)$ with features X and labels Y. We partition D into n subsets (X, Y_n) such that the features are part of each sub-dataset but we only have labels for a limited set of the inputs and Y_n are disjoint (i.e. $\forall_{i \neq j} Y_i \cap Y_j = \emptyset$). We then train a teacher model on each subset (X, Y_n). Instead of discarding the inputs without the labels and train the teachers in supervised fashion, we can use the unlabeled data to improve the accuracy of each teacher using semi-supervised learning (SSL), e.g. FixMatch [48]. Please note that this procedure is not limited to any specific SSL.

conducting this research) that achieved comparable accuracy as standard training algorithms on benchmark image datasets.

Algorithm 4 Confident-GNMax Aggregator

Require: input x, threshold T, noise parameters (σ_1, σ_2), n teacher models $f_1, \cdots, f_n$.
if $\max\{f_i(x)\} + \mathcal{N}(0, \sigma_1) \geq T$ **then**
return $\arg\max\{f_i(x) + \mathcal{N}(0, \sigma_2)\}$
end if
return $\perp$

Algorithm 5 Generating Private Labels by PATE-FM

Require: dataset $D = (X, Y)$, number of teachers n, number of labels for student model L, threshold T, noise parameters (σ_1, σ_2).
Train Teacher Models:
Divide the label set Y into n disjoint datasets $Y_1, \cdots, Y_n$
Train n teacher models (model f_i will be trained on Y_i and X via semi-supervised learning)
Train Student Model:
$X_s \leftarrow$Select L random samples from X
$Y_s \leftarrow$Get the noisy labels using Algorithm 4 for X_s
Train student model on (X_s, Y_s) and X

Aggregating the teachers: After training each sub-model, we need to aggregate the outputs of each sub-model and publish the output while preserving privacy. In particular, we will use Confident-GNMax Aggregator [43] (see Algorithm 4). The algorithm only releases private labels for the data points for which the teacher votes exceed a noisy threshold. Also, we only consider the privacy cost for the points that the algorithm decides to release. Moreover, Papernot et al. [43] showed that by using smooth sensitivity [40] instead of the global sensitivity it is possible to improve the privacy analysis (Please see Appendix 1 in Papernot et al. [43] for the privacy analysis of PATE). We use the same analysis as in Papernot et al. [43] to calculate the privacy cost of our approach. The main difference in our setting is that since the features of the instances are known we do not need public data for the student model.

Training the student: While we can use the aggregated teachers to answer queries, each query will increase the privacy cost. Therefore, instead we train a student model. In particular, we can use the idea of the SSL algorithms to get private labels for a small subset of the training dataset. Please note unlike the existing works [42, 43], since we only require label-DP, we do not need a public dataset. Algorithm 5 summarizes the overall steps of PATE-FM.

4.3 Learning with Private Labels

Using the approach in Section 4.2, we can obtain labels with differential privacy and we can then use the private labels for the main learning task. However, normal ML methods are designed to train on true labels while the private labels are generally noisy. Below, we discuss different methods for learning with private labels.

4.3.1 Normal

Given the noisy (differentially private) labels, the next step is to train the model. Algorithm 6 illustrates the general framework to train a model with the differentially private labels.

Algorithm 6 General framework for learning with differentially private labels: LDP-AlgName

Require: dataset with differentially private labels $D' = (X, \widetilde{Y})$ by Algorithm 3 or 5 on D.
train a model f on D' by AlgName.
return return f

Proposition 3. *By Proposition 2, Algorithm 6 does not have additional label privacy cost besides Algorithm 3 or 5.*

One simple method is to directly train the machine learning model with such noisy labels as training with the (sensitive) true labels by naive machine learning, i.e., supervised learning (directly calculated the gradients with respect to the noisy labels and update weights accordingly). We denote this as Algorithm 6: LDP-Normal. Ghazi et al. [23] propose a Label Privacy Multi-Stage Training (LP-MST) algorithm to preserve label privacy for deep learning, and LP1ST is equivalent to normal training (Algorithm 6: LDP-Normal) with private label set by RandRes (Ghazi et al. [23] considers mixup [59] to improve utility).

Next, we present two methods to improve the utility-privacy trade-off in label-DP.

4.3.2 DenoiseSSL: Denoising Assisted Semi-supervised Learning for Label-DP

Suppose that we obtain another set of labels for the training set without sacrificing the privacy of the la-

bels (denoted as pseudo label set and corresponding accuracy as Acc_0). We then only use the label information of the samples for which have the same differentially private label and pseudo label. We next show that this gives a subset with significantly more accurate labels.

Let us walk through the above denoising idea for RandRes where $p = e^\varepsilon/(e^\varepsilon + N - 1)$. Our strategy will select $Acc_0 \times p$ samples of which the differentially private labels are correct and $(1 - Acc_0) \times (1-p)/(N-1)$ which have wrong labels as their differentially private labels.[3] Therefore, the accuracy of the selected subset is

$$\begin{aligned} Acc' &= \frac{Acc_0 \times p}{Acc_0 \times p + (1 - Acc_0) \times \frac{1-p}{N-1}} \\ &= \frac{p}{p + \frac{1-Acc_0}{Acc_0} \times \frac{1-p}{N-1}} = \frac{e^\varepsilon}{e^\varepsilon + \frac{1-Acc_0}{Acc_0}} \end{aligned} \tag{8}$$

In the extreme case where the new label set cannot provide any label information and $Acc_0 = 1/N$, we can see that $(1 - Acc_0)/Acc_0 = N - 1$ and we have $Acc' = p$. If we increase Acc_0 from $1/N$, we have $(1-Acc_0)/Acc_0 < N - 1$ and therefore $Acc' > p$.

Algorithm 7 DenoiseSSL

Require: dataset with differentially private labels $D' = (X, \widetilde{Y})$ by Algorithm 3 or 5 on D.
$Z_{pseudo} = \{\}, D_{denoise} = \{\}$
Train unsupervised model f_0 on X with N clusters.
Run $(D'', \hat{Y}) \leftarrow$ Algorithm 1$((X, \widetilde{Y}), f_0)$
for all $l \in [N]$ **do**
  assign cluster of l with label $\arg\max\{\hat{Y}[l, :]\}$, i.e., add $(l, \arg\max \hat{Y}[l, :])$ to Z_{pseudo}
end for
for all $(x, \widetilde{y}, \hat{y}) \in D''$ **do**
  get y_{pseudo} from $\hat{y}$ according to the pseudo label look-up set Z_{pseudo}
  if $y_{pseudo} == \widetilde{y}$ **then**
    add $(x, \widetilde{y})$ to $D_{denoise}$
  else
    add $(x, \perp)$ to $D_{denoise}$
  end if
end for
train a model f on $D_{denoise}$ by SSL
return return f

We next introduce how to generate the pseudo label set without sacrificing the label privacy. As our discussion in Section 4.1, unsupervised learning is immune to label noise and we apply majority voting on differentially private labels to get cluster labels as pseudo labels (See Algorithm 7). Following Proposition 2, we do not incur additional privacy cost.

Next, we only use label information of training samples which have same pseudo label and differentially private label. As we now get a subset of the original training set with more accurate but fewer labels, we apply semi-supervised learning on the whole data set but only with the selected labels (See Algorithm 7).

Proposition 4. *By Proposition 2, Algorithm 7 does not have additional label privacy cost than Algorithm 6.*

Although the model is trained with fewer correct labels (as we will fill out $p \times (1 - Acc_0)$ samples which have correct labels), we will show that the model will get higher accuracy than directly applying SSL on the raw differentially private label set; note that the latter set has more labels but with lower accuracy, when we want strong label differential privacy, i.e., low ε. However, when ε is high, e.g., $\varepsilon = 8$ for $N = 10$, we will get differentially private label set with $p \geq 99.7\%$. The performance of denoising will be limited by the performance of unsupervised learning as it will filter out $p \times (1 - Acc_0) \approx (1 - Acc_0)$ correctly labeled samples. Therefore, we also consider applying semi-supervised learning with all training samples and noisy labels. We denote this as Algorithm 6: LDP-SSL and its privacy analysis is provided by Proposition 3.[4]

4.3.3 Aug-Descent: Dynamic Denoising in Training

One limitation of DenoiseSSL is that this method divides the clean label set and noisy label set before starting semi-supervised learning and keeps the labeled set fixed. The model can obtain more label information within the training procedure. To leverage more valid label information, we need to consider dynamically dividing the dataset during the training process and we adapt techniques from fields in learning from noisy labels. If

3 The denoising algorithm should also work for private label set generated by PATE-FM, but the private label by PATE-FM is dependent on PATE and cannot be directly estimated.

4 Please note that DenoiseSSL and LDP-SSL are not limited to any specific SSL. We used FixMatch [48], which was the state-of-the-art SSL algorithm (at the time of conducting this research) that achieved high accuracy on benchmark image datasets with very few labeled examples.

we view the differentially private labels as noisy labels generated in the label collection process, we can adapt existing learning with noisy labels technique to improve the utility-privacy trade-off in learning with label-DP guarantee. Specifically, we consider Aug-Descent [39], which applies Gaussian Mixture Model in each iteration to identify high confident samples as labeled samples and others as unlabeled samples, does label-refinement on labeled samples and labeled-sharpening on unlabeled samples, then learns on the refined and sharpened labels (See Algorithm 4 in Aug-Descent [39] for more details). We denote this as Algorithm 6: LDP-Aug-Descent and its privacy analysis is provided by Proposition 3.

5 Experiments

In this section, we first briefly introduce the datasets and model architectures used to train the classification models in Section 5.1. Next we present the performance of RandRes and PATE-FM in Section 5.2. Then we present the overall performance of our framework with each specific algorithms in Section 5.3. Finally we compare our framework with previous works (LP1ST (+mixup) [23] and ALIBI [35]) in Section 5.4.

5.1 Experimental Setup

Dataset. We use three benchmark datasets for image classification and target models which are widely used in the study of prior non-private learning algorithms.

CIFAR10-100 CIFAR10 and CIFAR100 [31] both contain 60,000 32×32 color (RGB) images (50,000 images as training set and 10,000 images as test set). CIFAR10 contains 10 classes and CIFAR100 contains 100 classes.

CINIC10 CINIC10 [11] contains 60,000 CIFAR10 images and 210,000 color (RGB) images rescaled to 32×32 from ImageNet [12]. This dataset uses 90,000 images for training/validation/test set. CINIC10 contains 10 classes, which is consistent with CIFAR10, but is more complicated than CIFAR10 as it has much more images.

Models. We use ResNet-18 [26] for all three datasets and all algorithms. We also vary model architectures on the comparison between our framework and previous works in Appendix A to show the effectiveness of our framework. We instantiate our unsupervised learning algorithms by SCAN [52], semi-supervised learning algorithms by FixMatch [48].

5.2 Evaluation of Differentially Private Labels

Table 2 presents the private label accuracy of the training set by RandRes and PATE-FM on the whole training set.[5] We can compute the theoretical private label accuracy for RandRes[6] by $acc(\varepsilon) = \frac{e^{\varepsilon}}{e^{\varepsilon}+N-1}$, while for PATE-FM, the private label accuracy is determined by the performance of PATE-FM and composition theorem of differential privacy. In addition, PATE-FM uses Gaussian noise and has an additional private parameter δ compared to RandRes. In this work, we set $\delta = 10^{-5}$.

Table 2. The accuracy of the differentially private label set (i.e. for training set) by RandRes and PATE-FM.

DP label. Method	label-DP level			
CIFAR10				
	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$
RandRes ($\delta = 0$)	15.5%	23.2%	45.2%	85.9%
PATE-FM ($\delta = 10^{-5}$)	93.2%	93.7%	94.9%	94.9%
CIFAR100				
	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 6$
RandRes ($\delta = 0$)	2.7%	6.9%	35.8%	80.4%
PATE-FM ($\delta = 10^{-5}$)	42.5%	48.1%	61.9%	69.3%
CINIC10				
	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$
RandRes ($\delta = 0$)	15.4%	23.2%	45.0%	85.9%
PATE-FM ($\delta = 10^{-5}$)	30.2%	77.3%	79.9%	85.1%

For CIFAR10, PATE-FM is significantly better than RandRes: even at $\varepsilon = 0.5$, PATE-FM provides 93.2% accuracy of differentially private labels (A single model in PATE-FM can have accuracy higher than 90%). For CIFAR100, PATE-FM is better than RandRes for $\varepsilon = 1, 2, 4$, (RandRes adds large noise for $\varepsilon = 1$ and the label accuracy is just 2.7%), but worse for $\varepsilon = 6$. This is because PATE-FM is limited by the performance of PATE-FM framework: CIFAR100 is a more difficult dataset compared to CIFAR10 as it has 100 classes and fewer samples (500 per class) and any single model in

5 NoiseCluster does not need any further learning process after generating the differentially private cluster and we will evaluate its performance in the next subsection.

6 Both of CIFAR10 and CINIC10 have 10 classes, so they have same theoretical accuracy $acc(\varepsilon)$ for same ε, though the reported accuracy in Table 2 differs a little due to randomness.

PATE-FM can not achieve accuracy higher than 75%. Therefore, even when no noise is added in Algorithm 5, the label generated from PATE-FM may be worse than RandRes (80.4%). RandRes is not affected with this fewer samples fact. With a fixed N, the value of p increases as ε increases and RandRes outputs a label set with higher accuracy (80.4% at $\varepsilon = 6$ by Equation (6)) accuracy. For CINIC10, PATE-FM is much better than RandRes for $\varepsilon = 0.5, 1, 2$ while a litter worse for $\varepsilon = 4$.

5.3 Evaluation of Our Whole Framework

Tables 3, 4 and 5, summarize the main results of the paper. We also include the non-private baseline in these Tables by running our framework on the clean label set and reporting the respective highest accuracy on each dataset as the non-private baseline. For each ε column, the bold number is the highest accuracy.

Overall Performance. We can see that most of the highest accuracy model are from Aug-Descent except for $\varepsilon = 0.5$ on CINIC10 (by RandRes+DenoiseSSL) and $\varepsilon = 2$ on CIFAR100 (by PATE-FM+SSL). Moreover, for highest accuracy model from Aug-Descent, we can find that when ε is low, the highest accuracy are by private label set generated by PATE-FM, and when ε becomes higher, the highest accuracy are by private label set generated by RandRes. Our analysis is that, for relative low ε where the private label set does not include much noise, e.g., CIFAR10 at $\varepsilon = 1$ by RandRes or PATE-FM, Aug-Descent can perform better than other algorithms. In addition, PATE-FM may be limited to the feature information as ε increases, while the value of p for label set by RandRes increases as ε increases with fixed N, e.g., CIFAR100 at $\varepsilon = 6$. Next we analyze the components of our framework in detail.

NoiseCluster performs well at low ε. Even without any specific individual label information, unsupervised learning can perform well to cluster samples of the same class to a single cluster. Besides, to get the label information of each cluster, NoiseCluster only needs to add noise to the counting of samples per class for each cluster instead of adding noise to each individual sample, therefore, it adds less noise to the label set in total compared to RandRes and PATE-FM. Please note that the same as PATE-FM, NoiseCluster also has a δ parameter. In this work, we set $\delta = 10^{-5}$. Though we only provide $\varepsilon = 0.5$ for CIFAR10/CINIC10 and $\varepsilon = 1$ for CIFAR100 in Tables, the lowest ε that NoiseCluster can provide without sacrificing utility is 0.007 for CIFAR10, 0.5 for CIFAR100 and 0.03 for CINIC10. In contrast, other pipelines which rely on RandRes and PATE-FM have worse performance compared to NoiseCluster as too much noise added when ε is much lower.

PATE-FM+SSL benefits from the choice of flexible number of queries instead of the whole training set to add less noise. After training the teacher models, querying each instance to be used for training the student model will increase the privacy cost. Therefore, the fewer queries, the less noise. The performance of PATE-FM+SSL is slightly better than PATE-FM on whole CIFAR10 dataset (i.e., query the whole CIFAR10 training set to PATE-FM). Also, PATE-FM+SSL is better than PATE-FM+DenoiseSSL at $\varepsilon = 2, 4, 6$ on CIFAR100. For $\varepsilon = 1$, we analyze that $\varepsilon = 1$ might be too low, there the limited number of labels and limited accuracy of the labels makes it hard for FixMatch to perform better than PATE-FM [35].

DenoiseSSL solves the limitation of NoiseCluster as ε increases. One limitation of NoiseCluster for label-DP is that it will not utilize the private label set during the clustering process so that the utility is limited to the clustering performance, even when the private label set has little noise. For example, on CIFAR100, the label set accuracy by RandRes is 80.4% at $\varepsilon = 6$ and 2.7% at $\varepsilon = 1$, while NoiseCluster is limited to 34.5% accuracy. DenoiseSSL overcomes this limitation by leveraging both the feature information and the private label information. For all three datasets, the classification accuracy of DenoiseSSL increases as the private label set has less noise. Besides, DenoiseSSL is better than LDP-SSL when ε is low, while worse when ε is higher, e.g., $\varepsilon = 6$ by RandRes on CIFAR100, as we have discussed in Section 4.3.

For most cases, DenoiseSSL performs better than NoiseCluster as DenoiseSSL benefits both from unsupervised learning and semi-supervised learning while NoiseCluster solely relies on unsupervised learning. However, when the accuracy of the noisy label set is extremely low, DenoiseSSL fails to achieve good accuracy and is even worse than NoiseCluster, this is because DenoiseSSL derives the purified label set from the noisy label set generated either from RandRes or PATE-FM, which adds more noise to the label set compared to NoiseCluster. For example, for $\varepsilon = 1$ by RandRes on CIFAR100, the private label accuracy (2.7%) is worse than the clustering accuracy (34.5%).

Aug-Descent outperforms other approaches as a benefit of dynamic division of labeled set and unlabeled set. DenoiseSSL divides the labeled set and unlabeled set before SSL, and the wrong label information in labeled set will keep misleading the training

Table 3. CIFAR10 (non-private baseline: 97.2%). For each ε column, the bold number is the highest accuracy.

Private Label Method	Learning Method with Private Labels	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$
NoiseCluster (SCAN)	N.A.	88.3%	88.3%	88.3%	88.3%
RandRes	DenoiseSSL (SCAN+FixMatch)	90.7%	90.8%	91.2%	93.6%
	LDP-SSL (FixMatch)	36.2%	61.9%	84.6%	92.7%
	LDP-Aug-Descent	37.7%	94.1%	**95.2%**	**96.5%**
PATE-FM	SSL (FixMatch)	94.1%	94.5%	95.1%	95.1%
	DenoiseSSL (SCAN+FixMatch)	93.3%	93.6%	93.7%	93.7%
	LDP-SSL (FixMatch)	93.4%	94.2%	94.5%	94.5%
	LDP-Aug-Descent	**94.6%**	**95.0%**	95.1%	95.1%

Table 4. CIFAR100 (non-private baseline: 83.3%). For each ε column, the bold number is the highest accuracy.

Private Label Method	Learning Method with Private Labels	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 6$
NoiseCluster (SCAN)	N.A.	34.5%	34.5%	34.5%	34.5%
RandRes	DenoiseSSL (SCAN+FixMatch)	8.3%	25.5%	43.3%	50.2%
	LDP-SSL (FixMatch)	1.8%	3.5%	51.4%	70.5%
	LDP-Aug-Descent	2.6%	10.8%	**75.3%**	**79.5%**
PATE-FM	SSL(FixMatch)	31.9%	**55.1%**	59.0%	68.1%
	DenoiseSSL (SCAN+FixMatch)	30.7%	34.7%	40.3%	42.8%
	LDP-SSL (FixMatch)	41.2%	47.4%	54.0%	58.1%
	LDP-Aug-Descent	**46.0%**	55.0%	60.1%	65.3%

Table 5. CINIC10 (non-private baseline: 90.0%). For each ε column, the bold number is the highest accuracy.

Private Label Method	Learning Method with Private Labels	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$
NoiseCluster (SCAN)	N.A.	62.8%	62.8%	62.8%	62.8%
RandRes	DenoiseSSL (SCAN+FixMatch)	**65.6%**	66.3%	68.2%	72.1%
	LDP-SSL (FixMatch)	20.8%	53.9%	73.6%	83.8%
	LDP-Aug-Descent	42.6%	80.7%	85.7%	**88.7%**
PATE-FM	SSL (FixMatch)	25.5%	76.0%	81.1%	85.4%
	DenoiseSSL (SCAN+FixMatch)	14.8%	70.5%	71.1%	71.8%
	LDP-SSL (FixMatch)	18.2%	81.7%	82.0%	83.5%
	LDP-Aug-Descent	18.9%	**87.6%**	**87.9%**	88.1%

Table 6. Comparison with the existing works.

Method	CIFAR10					CIFAR100				CINIC10				
	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$
Our work	94.6%	95.0%	95.2%	96.2%	96.5%	46.0%	55.1%	64.6%	75.3%	65.6%	87.6%	87.9%	88.1%	88.7%
ALIBI [35]	33.8%	70.0%	81.9%	87.2%	89.6%	4.7%	20.4%	51.2%	60.8%	30.6%	58.2%	69.8%	76.1%	79.5%
LP1ST [23]	38.4%	61.4%	83.2%	89.5%	92.0%	2.6%	7.6%	24.0%	52.6%	19.7%	52.0%	71.8%	80.1%	84.6%
Non-private baseline	97.2%					83.3%				90.0%				

process. The Aug-Descent algorithm [39] dynamically divides the labeled set at each round of the training process to better utilize the label information of samples which is correct with high confidence. In most cases, the highest accuracy of our framework (i.e., in bold), are from Aug-Descent Algorithms either on label set from RandRes or PATE-FM. When ε is low (e.g., CIFAR10 at $\varepsilon = 0.5$), the highest accuracy is usually from PATE-FM+Aug-Descent while as ε increase (e.g.,CIFAR100 at $\varepsilon = 6$), Aug-Descent+RandRes is better because RandRes provides the label set with higher accuracy.

One special case is CIFAR100 at $\varepsilon = 4$. Our explanation for why PATE-FM generates the private label set at $\varepsilon = 4$ with higher accuracy than RandRes while Aug-Descent on PATE-FM performs worse includes two folds. Firstly, the private label set by PATE-FM can be biased, which worsen utility. Secondly, PATE-FM generates the private label set mainly from the feature space and model, while RandRes is from the ground truth label alone, therefore, though the private label set by PATE-FM is more accurate than that by RandRes, it does not provide enough label information.

We have already discussed the performance and analysis of our framework in detail. Next we compare the performance of our framework with existing works.

5.4 Comparison with Existing Works

Table 6 compares our work and other label-DP mechanisms [23, 35] on CIFAR10 at $\varepsilon = 0.5, 1, 2, 3, 4$, CIFAR100 at $\varepsilon = 1, 2, 3, 4$ and CINIC10 at $\varepsilon = 0.5, 1, 2, 3, 4$. For our framework at each ε , we will report the highest accuracy in our framework.

Comparison with LP1ST (+mixup). LP1ST (+mixup) imposes a significant accuracy drop compared to training with non-private label (for evaluated ε, at most 58.8% on CIFAR10, at most 80.7% on CIFAR100 and at most 70.3% on CINIC10), while our method achieves comparable utility as non-private label training process on CIFAR10 and CINIC10 and significant utility advantage compared to LP1ST (+mixup) on CIFAR100: among three evaluated datasets, our method improves classification accuracy by $45.9\% \sim 56.2\%$ at $\varepsilon = 0.5$, $33.6\% \sim 43.4\%$ at $\varepsilon = 1$, $12.0\% \sim 47.5\%$ at $\varepsilon = 2$, by $6.7\% \sim 40.6\%$ at $\varepsilon = 3$, and by $4.1\% \sim 22.7\%$ at $\varepsilon = 4$. As ε increases, the utility advantage of our framework over previous algorithms decreases but is still significant.

Comparison with ALIBI. Our method achieves a better utility than ALIBI on all three datasets. ALIBI achieves a slightly worse performance than LP1ST (+mixup) on CIFAR10 and CINIC10 which have 10 classes for most cases, while performs much better on CIFAR100 which has 100 classes as ALIBI benefits adding much less noise for high label cardinality. However, ALIBI still incurs additional classification accuracy drop compared with our work: $35.0\% \sim 60.8\%$ for $\varepsilon = 0.5$, $25.0\% \sim 41.3\%$ for $\varepsilon = 1$, $13.3\% \sim 34.7\%$ for $\varepsilon = 2$, $9.0\% \sim 13.4\%$ for $\varepsilon = 3$ and $6.9\% \sim 14.5\%$ for $\varepsilon = 4$. Similarly as LP1ST, as ε increases, the utility advantage of our framework over previous algorithms decreases but is still significant.

6 Discussion

In this section we first discuss the impact of label cardinality on utility advantage of our framework. We then discuss two limitations of our framework. Finally we provide some insights on how to best maximize our framework performance under different scenarios.

Discussion of label cardinality Our experiment results include the label cardinality $N = 10$ and $N = 100$, which show that the noisy label accuracy by RandRes for 10 classes is higher than that at the same ε by RandRes for 100 classes. Therefore, the advantage of leveraging our framework on CIFAR10 is smaller than that on CIFAR100. For smaller label cardinality, e.g., binary classes, this is also true. We expect the utility advantage of our framework to exist for various label cardinality, though the advantage may be smaller for lower N.

Limitations: While our framework significantly improves the utility on CIFAR10, CIFAR100 and CINIC10 datasets, here we discuss two limitations of our work.

1. One limitation of our framework is that it requires more computing resources. Table 8 and Table 7 summarizes the computation time of generating the differentially private labels and training the models under label-DP respectively on CIFAR10. In Table 8, we can see that RandRes and ALIBI only incur negligible computation time, while NoiseCluster requires 21.3 h as it needs to train unsupervised learning model. PATE-FM requires much more time: it needs to train hundreds of models in the teacher model and therefore needs around 6000 GPU·h (though this process can be parallelized). For model training process, our framework also incurs much more computational resources. For example, DenoiseSSL needs sequential steps, i.e., we first need to train an unsupervised model f_0 (though f_0

Table 7. The computation time of training models on a single NVIDIA Tesla-P100 GPU under label-DP on CIFAR10.

Algorithm Name	LP-1ST(+mixup)	ALIBI	DenoiseSSL	LDP-SSL	LDP-AugDescent
Time	2.3 h	2.9 h	47.6 h	26.9 h	10.0 h

Table 8. The computation time of DP label on CIFAR10.

AlgName	NoiseCluster	RandRes	PATE-FM	ALIBI
Time	21.3 h	0.5 s	∼6000 GPU·h	0.03 s

can be used for both NoiseCluster and DenoiseSSL), we then use semi-supervised learning.

2. All three datasets evaluated are image datasets. For non-vision datasets, we expect that in our framework, NoiseCluster will still add less noise compared to RandRes and PATE-FM under the same ε privacy level, DenoiseSSL can generate a smaller label set with higher accuracy and LDP-LNL can improve utility, though the utility advantage may not be as significant as that on vision datasets.

Recommendations on how to make a choice of our framework: Here we provide some insights on how to best maximize our framework performance when offering label-DP guarantee under different scenarios:

1. When ε is extremely low and therefore much noise is added to the label set, for example, $\varepsilon = 0.007$ on CIFAR10, NoiseCluster provides a strong performance, i.e., 88.3% accuracy. In contrast, RandRes provides private label set of 10.06% accuracy.
2. If abundant computing resource is available and with a high label cardinality N and a low ε, for example CIFAR100 at $\varepsilon = 1$, PATE-FM is competitive: it adds less noise compared to RandRes (and therefore FixMatch and Aug-Descent on PATE-FM can performs better and are usually preferred), but requires much more computing resources to train hundreds of models. The improvement in GPU technology are making the computing resources cheap while the privacy threat remains severe, therefore we recommended PATE-FM if abundant computing resource is available. We consider the constraint *a high* N because at the same level of privacy, a higher N will have lower accuracy by RandRes, i.e., the accuracy decrease is different between RandRes and PATE-FM as N increases. Compare CIFAR10 and CIFAR100 at $\varepsilon = 2$. For CIFAR10, RandRes gives 45.2% while PATE-FM gives 94.9%. For CIFAR100, RandRes gives 6.9%, while PATE-FM gives 48.1%, i.e., the relative decreased accuracy is much lower than RandRes. We consider the constraint *a low* ε because for high ε, RandRes can be more accurate on PATE-FM as the PATE-FM will be limited to the single model's performance. In fact, for high ε, even with a high N, RandRes is less affected compared to PATE-FM when samples are significantly smaller as PATE-FM needs enough data to train the teacher model to get good accuracy, e.g., CIFAR100 at $\varepsilon = 6$.
3. Without above specific conditions, choosing RandRes makes the things simpler. Specifically, RandRes generally provides private label set with high accuracy for high ε. We can estimate the value of p by setting the label cardinality and epsilon. When ε is relatively low, it's advisable to consider our DenoiseSSL and algorithms like Aug-Descent to improve utility. For example, for CIFAR10 at $\varepsilon = 1$, the noisy label accuracy is 23.2%, and we advise the readers to consider our DenoiseSSL (90.8%) Aug-Descent (94.1%), which both are significantly higher than 23.2%, while at $\varepsilon = 8$, the noisy label accuracy is 99.7%, reader can simply consider LP1ST.

7 Related Work

7.1 Deep Learning with Differential Privacy

To achieve differential privacy in modern ML models, Abadi et al. [1] design DP-SGD training algorithm. Compared to the conventional SGD training algorithm, DP-SGD involves two additional steps: (1) gradient clipping to bound sensitivity, where the gradient per each training sample is clipped to make sure its norm is no larger than the clipping bound; (2) noise addition to achieve DP guarantees, where a random Gaussian noise is added to per-sample gradient before aggregating for gradient descent. For the added Gaussian noise, its mean is 0 and its standard deviation is proportional to the clipping bound in the gradient clipping step.

Furthermore, deep learning models are usually trained with a large number of iterations. Abadi et al. [1] propose the moments accountant to compute accumu-

lated privacy loss during training by tracking higher moments of the privacy loss random variable, which is shown to achieve tighter privacy analysis than standard strong composition theorem in DP [19]. The research community further proposes tight privacy estimation methods by using concentrated differential privacy [18] for privacy accounting [57], analyzing Rényi differential privacy amplification [38] with sub-sampling [54], tracking the privacy loss distribution with central limit theorem [49]. Meanwhile, researchers also design methods to improve the utility of differentially private deep learning models including using bounded activation functions [44], smoothing training loss functions [53]. Besides image classification datasets evaluated in the original paper [1], DP-SGD can also be applied into language models [36] and federated learning settings [22].

When public data is available, Papernot et al. [42, 43] propose another DP training algorithm by leveraging the semi-supervised knowledge transfer architecture. They first train an ensemble of independent teacher models on disjoint subsets of sensitive private training data. The predictions by these teacher models are then aggregated and added a Laplacian/Gaussian noise to achieve differential privacy guarantees. Finally, they train a student model as the final output on public data labeled by the private aggregation of teacher ensembles.

7.2 Deep Learning with Label Differential Privacy

Ghazi et al. [23] propose the first label DP training algorithm for deep learning. Similar as Section 4.2.1, they first obtain differentially private training labels by directly adding noises to labels with randomized response mechanism. To further improve standard supervised learning accuracy, they further employ a multi-stage training algorithm: the training set is divided into multiple non-overlapping subsets, on each stage, they use the last trained model to obtain predictions on a new training subset and add randomized response noises using these predictions as priors.

Malek et al. [35] further propose two label DP approaches. Their first approach is PATE-FM (see our Section 4.2.2). The second approach ALIBI proposes a soft randomized response mechanism by directly adding Laplace noises to one-hot encodings of training labels. Based on the post-processing property of DP, they further leverage Bayes formula to compute the valid soft training labels from the above noisy labels. After that, they perform standard optimization with soft labels.

7.3 Other Privacy Frameworks

Besides the standard differential privacy definition, the research community also seeks other quantification of privacy loss such as Rényi-DP [38], correlated-DP [61] and KL-DP [55] and capacity bounded DP [7]. Section 3 in Desfontaines et al. [13] gives a comprehensive survey on how to quantify the privacy loss.

Another line of research focuses on identifying features/attributes to be preserved and designing frameworks accordingly. Section 3 in Desfontaines et al. [13] gives a comprehensive summary on how to identify the neighboring dataset, e.g., changing the sensitive property [29] (record or attribute), Pufferfish [30], privacy axioms [28], Blowfish [27] and extension of DP using metrics [6]. These frameworks generalize DP by restricting the secrets about individuals that should not be inferred by the attacker, as well as explicitly enumerating the side information available to the adversary, and therefore allow to design novel, application specific privacy definitions that can achieve better privacy-utility trade-offs than the original DP (e.g., achieved using DP-SGD). Label-DP can be modeled as an instantiation of AttributeDP, Pufferfish or Blowfish, where the secrets to protect are the labels of training data (instead of both features and labels as in original DP).

8 Conclusion

In this work, we consider unsupervised learning and semi-supervised learning (SSL) and investigate three pipelines for label-DP. Pipeline 1 NoiseCluster is based on unsupervised learning and does not include private labels in the learning phase, therefore much less noise added. Pipeline 2 focuses on how to leverage the private labels in the learning phase which includes our proposed denoising algorithm DenoiseSSL based on unsupervised learning and semi-supervised learning. Pipeline 3 which is based on PATE-FM benefits from only querying a partial set of training set to reduce added noise. For evaluation, we first investigated the quality of private label generation mechanisms, RandRes and PATE-FM. Next we evaluate our three pipelines and compare with previous works. We also state our discussion on label cardinality, limitations and how to make the right choice in our proposed framework. We hope this work can provide insights on how to design label-DP systems.

Acknowledgements

We are grateful to anonymous reviewers at PoPETs for valuable feedback. This work was supported in part by the National Science Foundation under grants CNS-2131910, CNS-1553437, CNS-1553301, CNS-1704105, and CNS-1953786, the ARL's Army Artificial Intelligence Innovation Institute (A2I2), the Office of Naval Research Young Investigator Award, the Army Research Office Young Investigator Prize, Schmidt DataX award, and Princeton E-ffiliates Award.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[2] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[3] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Advances in Neural Information Processing Systems*, 2014.

[4] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 394–403.

[5] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 2014, pp. 464–473.

[6] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Privacy Enhancing Technologies*, 2013, pp. 82–102.

[7] K. Chaudhuri, J. Imola, and A. Machanavajjhala, "Capacity bounded differential privacy," in *Advances in Neural Information Processing Systems*, 2019.

[8] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization." *Journal of Machine Learning Research*, vol. 12, 2011.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 1597–1607.

[10] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems*, 2020.

[11] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," *arXiv preprint arXiv:1810.03505*, 2018.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[13] D. Desfontaines and B. Pejó, "Sok: differential privacies," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 2, pp. 288–313, 2020.

[14] T. G. Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*, 2000, pp. 1–15.

[15] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, 2008, pp. 1–19.

[16] ——, "Differential privacy," in *Encyclopedia of Cryptography and Security, 2nd Ed*, 2011, pp. 338–340.

[17] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[18] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.

[19] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 51–60.

[20] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta, "Privacy amplification by iteration," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 521–532.

[21] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[22] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[23] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, and C. Zhang, "Deep learning with label differential privacy," in *Advances in Neural Information Processing Systems*, 2021.

[24] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, "Co-teaching: robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems*, 2018.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 2014, p. 1447–1458.

[28] D. Kifer and B.-R. Lin, "Towards an axiomatization of statistical privacy and utility," in *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2010, p. 147–158.

[29] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD Interna-*

tional Conference on Management of Data, 2011.

[30] ——, "Pufferfish: A framework for mathematical privacy definitions," *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 1, 2014.

[31] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[32] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[33] H. Li, L. Xiong, L. Ohno-Machado, and X. Jiang, "Privacy preserving rbf kernel support vector machine," *BioMed Research International*, 2014.

[34] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *International Conference on Learning Representations*, 2019.

[35] M. Malek, I. Mironov, K. Prasad, I. Shilov, and F. Tramèr, "Antipodes of label differential privacy: Pate and alibi," in *Advances in Neural Information Processing Systems*, 2021.

[36] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.

[37] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 691–706.

[38] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 263–275.

[39] K. Nishi, Y. Ding, A. Rich, and T. Höllerer, "Augmentation strategies for learning with noisy labels," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[40] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, 2007, p. 75–84.

[41] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *International Conference on Learning Representations*, 2017.

[43] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with pate," in *International Conference on Learning Representations*, 2018.

[44] N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson, "Tempered sigmoid activations for deep learning with differential privacy," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021.

[45] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin, "Genomic privacy and limits of individual detection in a pool," *Nature genetics*, vol. 41, no. 9, pp. 965–967, 2009.

[46] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[48] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Advances in Neural Information Processing Systems*, 2020.

[49] D. M. Sommer, S. Meiser, and E. Mohammadi, "Privacy loss classes: The central limit theorem in differential privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 245–269, 2019.

[50] C. Song and V. Shmatikov, "Overlearning reveals sensitive attributes," in *International Conference on Learning Representations*, 2019.

[51] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *USENIX Security*, 2016.

[52] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *European Conference on Computer Vision*, 2020, pp. 268–285.

[53] W. Wang, T. Wang, L. Wang, N. Luo, P. Zhou, D. Song, and R. Jia, "Dplis: Boosting utility of differentially private deep learning via randomized smoothing," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 4, pp. 163–183, 2021.

[54] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled rényi differential privacy and analytical moments accountant," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 1226–1235.

[55] Y.-X. Wang, J. Lei, and S. E. Fienberg, "On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms," in *International Conference on Privacy in Statistical Databases*, 2016, pp. 121–134.

[56] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *JASA*, vol. 60, no. 309, pp. 63–69, 1965.

[57] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 332–349.

[58] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference*, 2016, pp. 87.1–87.12.

[59] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[60] Z. Zhang, B. I. Rubinstein, and C. Dimitrakakis, "On the differential privacy of bayesian inference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[61] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-iid data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229–242, 2014.

Table 9. Comparison with the existing works of different model architectures on CIFAR10.

	ResNet-18			VGG-16			WRN-28-4		
Method	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$
Our work	95.2%	96.2%	96.5%	94.5%	95.0%	95.4%	93.1%	94.1%	94.7%
ALIBI [35]	81.9%	87.2%	89.6%	81.6%	86.0%	88.3%	73.0%	82.8%	86.9%
LP1ST [23]	83.2%	89.5%	92.0%	83.5%	87.0%	91.8%	75.7%	88.7%	92.7%
Non-private baseline	97.2%			95.8%			96.9%		

A Ablation Study on Different Model Architectures

In addition to ResNet-18 [26], we use two more architectures including VGG-16 [47] and WideResNet-28-4 (WRN-28-4) [58] and present the comparison of our framework and previous works on CIFAR10 dataset at $\varepsilon = 2, 3, 4$ in Table 9. On all three different model architectures, we can see that our framework is much better than previous works LP1ST (+mixup) and ALIBI. Specifically, at $\varepsilon = 4$, our framework is close to the non-private baseline, which shows the effectiveness of our framework across different model architectures.