

Per-Instance Privacy Accounting for Differentially Private Stochastic Gradient Descent

Da Yu[†] Gautam Kamath^{‡*} Janardhan Kulkarni^{§*}
 Tie-Yan Liu^{§*} Jian Yin^{†*} Huishuai Zhang^{§*}

June 8, 2022

Abstract

Differentially private stochastic gradient descent (DP-SGD) is the workhorse algorithm for recent advances in private deep learning. It provides a single privacy guarantee to all datapoints in the dataset. We propose an efficient algorithm to compute per-instance privacy guarantees for individual examples when running DP-SGD. We use our algorithm to investigate per-instance privacy losses across a number of datasets. We find that most examples enjoy stronger privacy guarantees than the worst-case bounds. We further discover that the loss and the privacy loss on an example are well-correlated. This implies groups that are underserved in terms of model utility are simultaneously underserved in terms of privacy loss. For example, on CIFAR-10, the average ϵ of the class with the highest loss (Cat) is 32% higher than that of the class with the lowest loss (Ship). We also run membership inference attacks to show this reflects disparate empirical privacy risks.

1 Introduction

Differential privacy is a strong notion of data privacy, enabling rich forms of privacy-preserving data analysis [DMNS06, DR14]. Informally speaking, it quantitatively bounds the maximum influence of any datapoint using a privacy parameter ϵ , where a small value of ϵ corresponds to stronger privacy guarantees. Training deep models with differential privacy is an active research area [ACG⁺16, PAE⁺17, BDLS20, YNB⁺22, AGG⁺21, LTLH22, GAW⁺22, MTKC22, DBH⁺22]. Models trained with differential privacy not only provide theoretical privacy guarantee to their data but also are more robust against empirical attacks [BGRK19, CLE⁺19, JUO20, NST⁺21].

Differentially private stochastic gradient descent (DP-SGD) is the de-facto choice for differentially private deep learning [SCS13, BST14, ACG⁺16]. DP-SGD first clips per-instance gradients and then adds Gaussian noise to the aggregated gradients. Standard privacy accounting takes a worst-case approach, and provides all examples with the same privacy parameter ϵ . However, from the perspective of machine learning, different examples can have very different impacts on a learning algorithm [Koh17, FZ20]. For example, consider support vector machines: removing a non-support vector has no effect on the resulting model, and hence that example would have perfect privacy.

In this paper, we give an efficient algorithm to approximately compute per-instance privacy parameters for DP-SGD. Inspecting these per-instance privacy parameters allow us to better understand instance-wise impacts. It turns out that, for a particular dataset, many instances experience much lower privacy loss than the worst-case guarantees. To illustrate this, we plot the per-instance privacy parameters for CIFAR-10 and MNIST in Figure 1. Experimental details, as well as more results, can be found in Section 4. These differences in per-instance privacy parameters naturally arise when running DP-SGD. To the best of our knowledge, our investigation is the first to explicitly reveal this difference.

[†]Sun Yat-sen University. {yuda3@mail2, issjyin@mail}.sysu.edu.cn

[‡]Cheriton School of Computer Science, University of Waterloo. Supported by an NSERC Discovery Grant, an unrestricted gift from Google, and a University of Waterloo startup grant. g@csail.mit.edu

[§]Microsoft Research. {jakul, tyliu, huzhang}@microsoft.com

* Authors are listed in alphabetical order.

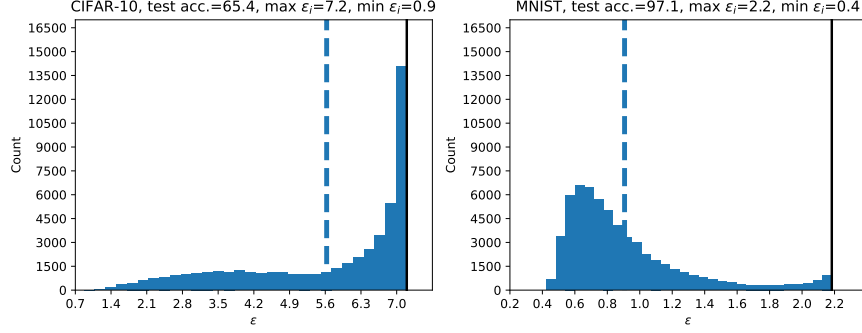


Figure 1: Distribution of per-instance privacy parameters on CIFAR-10 and MNIST. The value of δ is 1×10^{-5} . The dashed line indicates the average of ϵ values. The black solid line indicates the original privacy parameter of DP-SGD for all instances.

We propose two techniques to make per-instance privacy accounting viable for DP-SGD. First, we maintain estimates of the gradient norms for all examples so the per-instance privacy loss can be computed accurately at every update. Second, we round the gradient norms with a small precision r to control the number of unique privacy losses, which need to be computed numerically. We explain why these two techniques are necessary in Section 2. More details of the proposed algorithm, as well as methods to release per-instance parameters without additional privacy loss, are in Section 3.

We further demonstrate a strong correlation between per-instance privacy parameters and per-instance losses. That is, we find that datapoints with large per-instance privacy parameters usually also experience high losses over the training process. Stated differently: the same examples suffer a simultaneous unfairness in terms of worse privacy and worse utility. While prior works have shown that underrepresented groups experience worse utility [BG18], and that these disparities are amplified when models are trained privately [BPS19, SPGG21, PBHTNS⁺22, NHS22], we are the first to show that the privacy loss *and* utility are both negatively impacted concurrently. This is in comparison to prior work in the differentially private setting which took a worst-case perspective for privacy accounting, resulting in a uniform privacy loss for all training examples. Empirical evaluation on MNIST [LBBH98], CIFAR-10 [Kri09], and UTKFace [ZSQ17] datasets are in Section 5. For instance, when running gender classification on the UTKFace dataset, the average ϵ of the subgroup with the highest loss (Asian) is 28% higher than that of the subgroup with the lowest loss (Indian). We also run membership inference attacks on those datasets and show the privacy parameters correlate well with the success rates.

1.1 Related Work

There are several works exploring instance-wise privacy in differentially private learning. [RW21] study how to privately publish the per-instance guarantees of objective perturbation. [FZ21] design a Rényi filter to make use of the per-instance privacy budget of DP-GD. [MB22] provide per-instance privacy guarantees for the PATE framework. In this work, we give an algorithm to compute the per-instance privacy guarantees for DP-SGD, which is the most widely used algorithm in differentially private deep learning.

A recent line of work has found that some examples are more vulnerable to empirical attacks [LBG17, KYC⁺19, CCTCP21, CCN⁺21]. They show membership inference attacks have significantly higher success rates on some examples, e.g., on some specific classes [SSSS17]. In this work, we show the disparity of pre-example privacy risks also exists theoretically when learning with differential privacy. Moreover, we also show the disparity in privacy risks correlates well with the disparity in utility.

2 Preliminaries

We first give some background on differentially private learning. We then describe the privacy analysis of DP-SGD and highlight the challenges in computing per-instance privacy parameters. Finally, we argue that providing the same

privacy bound to all samples is not ideal because different examples naturally have different privacy losses due to variation in gradient norms.

2.1 Background on Differentially Private Learning

Differential privacy is built on the notion of neighboring datasets. A dataset $\mathbb{D} = \{\mathbf{d}_i\}_{i=1}^n$ is a neighboring dataset of \mathbb{D}' (denoted as $\mathbb{D} \sim \mathbb{D}'$) if \mathbb{D}' can be obtained by adding/removing one example from \mathbb{D} . We use the following *per-instance* form of (ε, δ) -differential privacy [Wan19, RW21].

Definition 1. [Per-instance DP] Fix a dataset \mathbb{D} and $\mathbb{D}' = \mathbb{D} \setminus \{\mathbf{d}\}$. An algorithm \mathcal{A} satisfies (ε, δ) -differential privacy for (\mathbb{D}, \mathbf{d}) if for any subset of outputs S it holds that $\Pr[\mathcal{A}(\mathbb{D}) \in S] \leq e^\varepsilon \Pr[\mathcal{A}(\mathbb{D}') \in S] + \delta$ and $\Pr[\mathcal{A}(\mathbb{D}') \in S] \leq e^\varepsilon \Pr[\mathcal{A}(\mathbb{D}) \in S] + \delta$.

The privacy guarantee of Definition 1 is for a pair of a dataset \mathbb{D} and a single datapoint \mathbf{d} . It is slightly different with the individual privacy notions in [JYC15] and [FZ21], where the guarantee is for a datapoint \mathbf{d} while the dataset \mathbb{D}' is arbitrarily chosen. Definition 2 gives the notion of individual differential privacy.

Definition 2. [Individual DP] Fix a datapoint \mathbf{d} , for arbitrary \mathbb{D} and $\mathbb{D}' = \mathbb{D} \setminus \{\mathbf{d}\}$, an algorithm \mathcal{A} satisfies (ε, δ) -differential privacy for \mathbf{d} if for any subset of outputs S it holds that $\Pr[\mathcal{A}(\mathbb{D}) \in S] \leq e^\varepsilon \Pr[\mathcal{A}(\mathbb{D}') \in S] + \delta$ and $\Pr[\mathcal{A}(\mathbb{D}') \in S] \leq e^\varepsilon \Pr[\mathcal{A}(\mathbb{D}) \in S] + \delta$.

The privacy guarantee in Definition 2 is for a datapoint \mathbf{d} which means the guarantee holds when \mathbb{D}' is arbitrarily chosen. Although individual DP is a stronger privacy notion, it may hide the disparate privacy parameters of an individual datapoint because the disparity is inherent with respect to a specific dataset. A datapoint may be an inlier for one dataset but an outlier for another, and would thus suffer disparate privacy losses. For example, one can modify the dataset \mathbb{D} maliciously to maximize the influence of a given example \mathbf{d} [TSJ⁺22]. If using individual DP, the final guarantee for \mathbf{d} would be the worst-case privacy loss for the worst possible \mathbb{D} . However, the privacy risk of a trained model is always bound to a given training set and many examples would have a much smaller influence than the worst case. Therefore, we use Definition 1 to evade the worst-case analysis.

DP-SGD is the most common approach for doing deep learning with differential privacy. Instead of protecting the trained model directly, DP-SGD makes each SGD update differentially private. The composition property of differential privacy allows us to reason about the overall privacy of running several such steps. In this work, privacy of different steps is composited through *Rényi Differential Privacy* [Mir17]. More details on per-instance Rényi DP and how it composes are in Appendix A. The overall Rényi DP of an instance is converted into (ε, δ) -DP after training.

We give a simple example to illustrate how to privatize each update. Suppose we take the sum of all gradients $\mathbf{v} = \sum_{i=1}^n \mathbf{g}_i$ from dataset \mathbb{D} . Without loss of generality, further assume we add an arbitrary example \mathbf{d}' to obtain a neighboring dataset \mathbb{D}' . The summed gradient becomes $\mathbf{v}' = \mathbf{v} + \mathbf{g}'$, where \mathbf{g}' is the gradient of \mathbf{d}' . If we add independent Gaussian noise with variance σ^2 to each coordinate, then the output distributions of two neighboring datasets are

$$\mathcal{A}(\mathbb{D}) \sim \mathcal{N}(\mathbf{v}, \sigma^2 \mathbf{I}) \text{ and } \mathcal{A}(\mathbb{D}') \sim \mathcal{N}(\mathbf{v}', \sigma^2 \mathbf{I}).$$

We then can bound the Rényi divergence between $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ to provide (ε, δ) -DP. Their expectations only differ by \mathbf{g}' and hence a large gradient leads to a large divergence (privacy loss).

2.2 Challenges of Computing Per-Instance Privacy Parameters for DP-SGD

Privacy accounting in DP-SGD is more complex than the simple example in Section 2.1 because the analysis involves *privacy amplification by subsampling* [ACG⁺16, BBG18, MTZ19, ZW19, WBK19]. Roughly speaking, randomly sampling a minibatch in DP-SGD strengthens the privacy guarantees since most points in the dataset are not involved in a single step. This complication makes direct computation of per-instance privacy parameters impractical.

Before we expand on these difficulties, we first describe the output distributions of neighboring datasets in DP-SGD [ACG⁺16]. Poisson sampling is assumed, i.e., each example is sampled independently with probability p . Let $\mathbf{v} = \sum_{i \in \mathbb{M}} \mathbf{g}_i$ be the sum of the minibatch of gradients of \mathbb{D} , where \mathbb{M} is the set of sampled indices. Consider also a neighboring dataset \mathbb{D}' that has one datapoint with gradient \mathbf{g}' added. Because of Poisson sampling, the output is

exactly \mathbf{v} with probability $1 - p$ (\mathbf{g}' is not sampled) and is $\mathbf{v}' = \mathbf{v} + \mathbf{g}'$ with probability p (\mathbf{g}' is sampled). Suppose we still add isotropic Gaussian noise, the output distributions of two neighboring datasets are

$$\mathcal{A}(\mathbb{D}) \sim \mathcal{N}(\mathbf{v}, \sigma^2 \mathbf{I}), \quad (1)$$

$$\mathcal{A}(\mathbb{D}') \sim \mathcal{N}(\mathbf{v}, \sigma^2 \mathbf{I}) \text{ with prob. } 1 - p, \quad \mathcal{A}(\mathbb{D}') \sim \mathcal{N}(\mathbf{v}', \sigma^2 \mathbf{I}) \text{ with prob. } p. \quad (2)$$

With Equation (1) and (2), we explain the challenge in computing per-instance privacy parameters.

2.2.1 Full Batch Gradient Norms Are Required at Every Iteration

There is some privacy loss for \mathbf{d}' even if it is not sampled in the current iteration because the analysis makes use of the subsampling process. For a given sampling probability and noise variance, the amount of privacy loss is determined by $\|\mathbf{g}'\|$. Therefore, we need accurate gradient norms of all examples to compute accurate privacy losses at every iteration. However, when running SGD, we only have minibatch gradients. Previous analysis of DP-SGD evades this problem by simply assuming all examples have the maximum possible norm, i.e., the clipping threshold.

2.2.2 Computational Cost of Per-Instance Privacy Parameters is Huge

The density function of $\mathcal{A}(\mathbb{D}')$ is a mixture of two Gaussian distributions. This makes computing the Rényi divergence between $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ harder as there are no closed form solutions. Although there are some asymptotic bounds, those bounds are looser than computing the divergence numerically [ACG⁺16, WBK19, MTZ19, GLW21], and thus such numerical computations are necessary to achieve strong privacy guarantees. In the classic analysis, there is only one numerical computation as all examples have the same privacy loss over all iterations. However, naive computation of per-instance privacy losses would require up to $n \times T$ computations, where n is the dataset size and T is the number of iterations.

2.3 An Observation: Gradient Norms in Deep Learning Vary Significantly

We show the gradient norms vary significantly across datapoints in the dataset to demonstrate that different examples experience very different privacy losses when running DP-SGD. We train the standard ResNet-20 model in [HZRS16] on CIFAR-10. The maximum clipping threshold is the median of gradient norms at initialization. More details are in Section 4. We first sort all examples based on their average gradient norms across training. Then we divide them into five equally sized groups based on their quintile. We plot the average norms across training in Figure 2. The norms of different groups show significant stratification. Such stratification naturally leads to different per-instance privacy losses and hence different privacy parameters for each example. This suggests that quantifying per-instance privacy parameters may be valuable despite the aforementioned challenges.

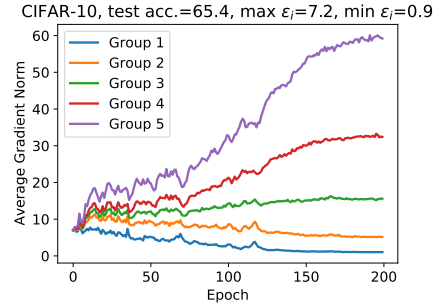


Figure 2: Average gradient norms.

3 Deep Learning with Per-Instance Privacy Parameters

We give an efficient algorithm to compute per-instance privacy parameters for DP-SGD (Algorithm 1). We perform two modifications to make per-instance privacy loss accounting feasible with small computational overhead. We first use past gradient norms to estimate the gradients norms at the current iteration, and update the estimates when points are sampled into the current minibatch. We also introduce the option to compute full batch gradient norms deliberately to trade off between running time and estimation accuracy. Additionally, we round the gradient norms to a given precision so the number of numerical computations is independent of the dataset size and number of iterations.

3.1 Estimated Privacy Parameters Are Accurate

Although the gradient norms used for privacy accounting are updated only occasionally, we show that the computed per-instance privacy parameters are very close to the actual ones (Figure 3). This indicates that, in general, the gradient

Algorithm 1: Deep Learning with Per-Instance Privacy Accounting

Input : Maximum clipping threshold C , rounding precision r , noise variance σ^2 , sampling probability p , frequency of updating full gradient norms at every epoch K .

- 1 Let $\{C_i\}_{i=1}^n$ be the estimated gradient norms of all examples and initialize $C_i = C$.
- 2 Let $\mathbb{C} = [r, 2r, 3r, \dots, C]$ be all possible norms under rounding.
- 3 **for** $c \in \mathbb{C}$ **do**
- 4 //Formulations of $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ are in Equation (1) and (2).
- 5 Compute the Rényi divergences between $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ numerically with c , p , and σ^2 .
- 6 **end**
- 7 **for** $t = 1$ to $T \times E$ **do**
- 8 // T is the number of iterations per epoch, E is the number of epochs.
- 9 Sample a minibatch of indices $\{I_j\}_{j=1}^m$ and compute gradients $\{g_{I_j}\}_{j=1}^m$.
- 10 Compute $\|g_{I_j}\|$ and $\bar{g}_{I_j} = \text{clip}(g_{I_j}, C_{I_j})$.
- 11 Update model with $\sum \bar{g}_{I_j} + z$, where $z \sim \mathcal{N}(0, \sigma^2 I)$.
- 12 Update $C_{I_j} = \min(\|g_{I_j}\|, C)$.
- 13 Round C_{I_j} with precision r .
- 14 //Update privacy loss for the whole dataset.
- 15 **for** $i = 1$ to n **do**
- 16 Find corresponding $c \in \mathbb{C}$ for the i_{th} example.
- 17 Add the privacy loss of c to the accumulated loss of the i_{th} example.
- 18 **end**
- 19 **if** $K > 0$ and $t \bmod \lfloor T/K \rfloor = 0$ **then**
- 20 Compute full batch gradient norms and update $\{C_{I_j}\}_{i=1}^n$ with rounded norms.
- 21 **end**
- 22 **end**

norms do not change rapidly during training. Before we examine this phenomenon, we note that the estimated privacy parameters themselves are strict differential privacy guarantees because we use the estimated norms to clip per-instance gradients.¹

To compute the actual privacy parameters, we randomly sample 1000 examples and compute the exact gradient norms at every iteration. We compute the Pearson correlation coefficient between the estimated and actual privacy parameters as well as the average and the worst absolute errors. In addition to CIFAR-10 and MNIST, we also include the UTKFace dataset and run age/gender classification tasks on it [ZSQ17]. Details about the experiments are in Section 4.

We plot the results in Figure 3. The estimated ϵ values are very close to the actual ones (Pearson’s $r > 0.99$) even we only update the gradient norms when points are sampled into a minibatch, which incurs almost no computational overhead as those gradients are already calculated by DP-SGD. Updating full batch gradient norms once or twice per epoch further improves the estimation, though doing so would double or triple the running time.

It is worth noting that C affects the computed privacy parameters. Large C increases the variation of gradient norms but leads to large worst-case privacy (or large gradient variance if keeping the worst-case privacy unchanged) while small C suppresses the variation and leads to large gradient bias [CWH20, SSTT21]. In this work, we set the maximum clipping threshold as the median of gradient norms at initialization unless otherwise mentioned, which is a common choice in practice and has been observed to achieve good accuracy [ACG⁺16, ATMR21]. In Appendix D, we show the influence of using different different values of C on both accuracy and privacy.

3.2 Rounding Per-Instance Gradient Norms

The rounding operation in Algorithm 1 is essential to make the computation of per-instance privacy loss feasible. For given σ^2 and p , one needs to run the numerical method in [MTZ19] once for every unique privacy loss. Consequently,

¹Using per-instance clipping thresholds could lose more gradient signal if the estimates are inaccurate. In Appendix C, we show the per-instance clipping in Algorithm 1 does not affect the utility.

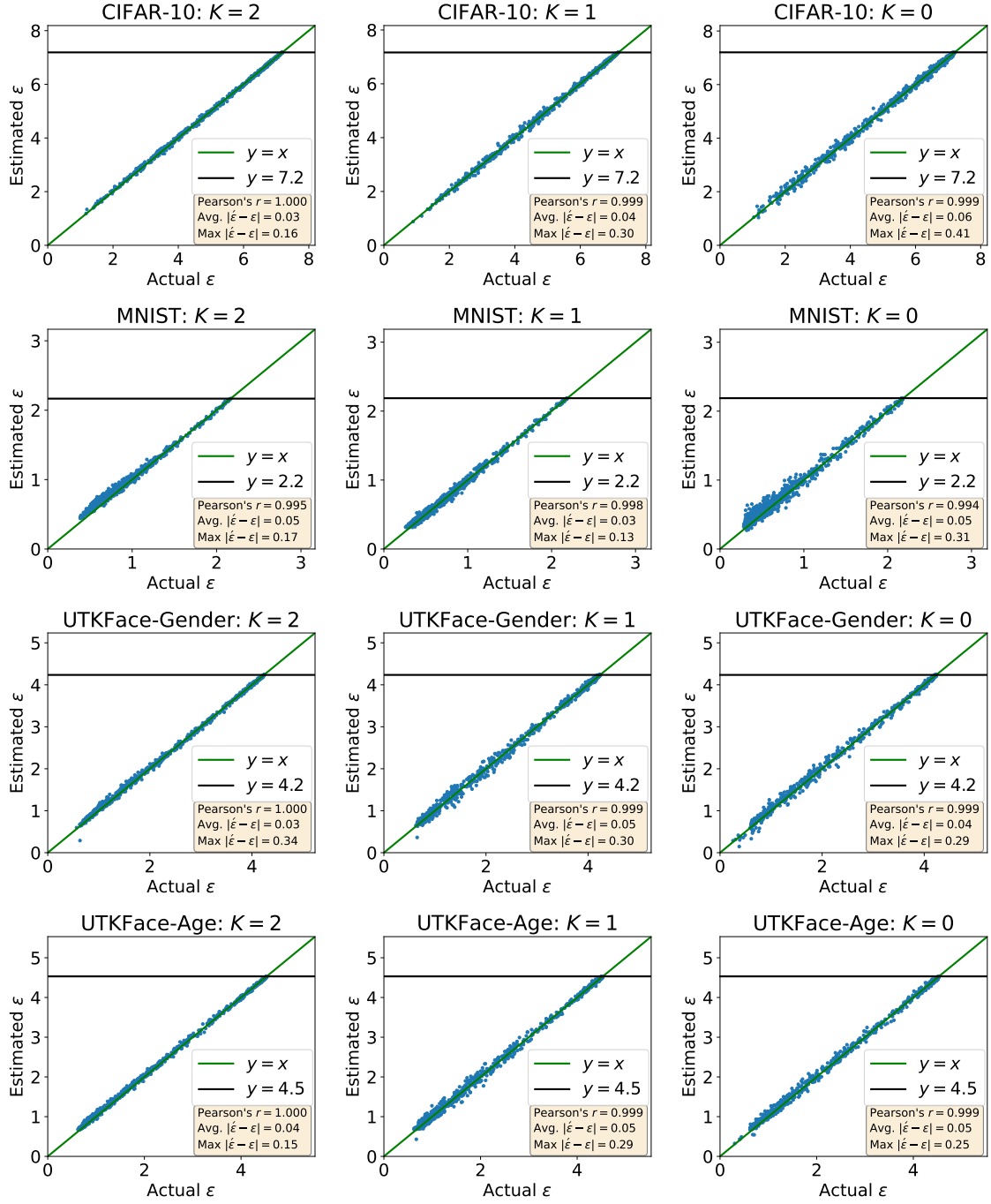


Figure 3: Estimated ϵ values versus actual ϵ values. The value of K is the times of full batch norms update at every epoch. The solid black line indicates ϵ of the original analysis for every example.

there are at most $n \times T$ unique privacy losses because gradient norms vary across different examples and iterations. In order to make the number of unique privacy losses tractable, we round the gradient norms with a prespecified precision r . Because the maximum clipping threshold is usually a small constant, then, by the pigeonhole principle, there are at most $\lceil C/r \rceil$ unique values. Throughout this paper we set $r = 0.1$, which has almost no impact on the precision of privacy accounting.

To give a concrete running time comparison, running the numerical method of [MTZ19] one hundred times using a single CPU core takes around five seconds. This is the total computation overhead of Algorithm 1 on CIFAR-10 ($n = 50000$) with $K = 0$ in Figure 3. We only need to run the computations before training and can reuse the results directly throughout training. However, without rounding, we need to compute 50000 unique privacy losses at every epoch on CIFAR-10. This will incur an additional cost of approximately 10 hours to compute the per-instance privacy parameters.

3.3 What Can We Do with Per-Instance Privacy Parameters?

Note that per-instance privacy parameters are dependent on the private data and thus sensitive, and consequently may not be released publicly without care. We describe some approaches to safely make use of per-instance privacy parameters. The first is to only release the privacy parameter to the rightful data owner. The second is to release some statistics of the per-instance privacy parameters to the public. Both approaches offer more granular and tighter privacy guarantees than the single worst-case guarantee offered by the conventional analysis. Another approach is for a trusted data curator to improve the model quality based on the per-instance parameters.

The first approach is to release ε_i to the owner of the i th example. Although we use gradient norms without adding noise, this approach does not incur additional privacy loss for two reasons. First, it is safe for the i th example because only the rightful owner sees ε_i . Second, releasing ε_i does not increase the privacy loss of any other examples. This is because the *post-processing* property of differential privacy and the fact that computing ε_i only involves a differentially private model and the i th example itself. The second reason is important and it may not hold under other private learning algorithms. For example, the per-instance privacy parameters for objective perturbation are interdependent and require additional delicate analysis before publication [RW21].

The second approach is to privately release aggregate statistics of the population, e.g., the average or quantiles of the ε values. Recent works have demonstrated such statistics can be published accurately with minor increase in the privacy loss [ATMR21]. In Appendix E, we show the statistics can be released accurately with very small privacy cost ($\varepsilon \leq 0.1$).

Finally, per-instance privacy parameters can also serve as a powerful tool for a trusted data curator to improve the model quality. By analysing the per-instance privacy parameters of a dataset, a trusted curator can focus on collecting more data representative of the subgroups that have higher privacy risks (and worse performance) to mitigate the disparity in privacy.

4 Per-Instance Privacy Parameters on Different Datasets

We investigate the distribution of per-instance privacy parameters of running DP-SGD on four classification tasks. All experiments are run on a Tesla V100 GPU. Experimental setup is as follows.

Datasets. We use two benchmark datasets MNIST ($n = 60000$) and CIFAR-10 ($n = 50000$) [LBBH98, Kri09] as well as the UTKFace dataset ($n \simeq 15000$) [ZSQ17] that contains the face images of four different races (White, $n \simeq 7000$; Black, $n \simeq 3500$; Asian, $n \simeq 2000$; Indian, $n \simeq 2800$). We construct two tasks on UTKFace: predicting gender, and predicting whether the age is under 30.² We slightly modify the dataset between these two tasks by randomly removing a few examples to ensure each race has balanced positive and negative labels.

Models and hyperparameters. We train ResNet-20 models on all datasets. For CIFAR-10 and MNIST, we train the models from scratch. For UTKFace, we fine-tune a model from the PyTorch library³ that is pre-trained on ImageNet.

²We acknowledge that predicting gender and age from images may be problematic. Nonetheless, as facial images have previously been highlighted as a setting where machine learning has disparate accuracy on different subgroups, we revisit this domain through a related lens. The labels are provided by the dataset curators.

³<https://pytorch.org/vision/stable/models.html>

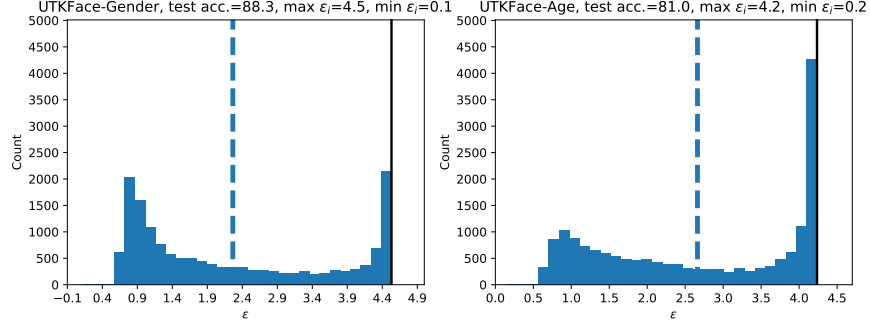


Figure 4: Distribution of per-instance privacy parameters on UTKFace. The value of δ is 1×10^{-5} . The dashed line indicates the average of ϵ values. The black solid line indicates the original privacy parameter of DP-SGD for all instances.

For all datasets, the maximum clipping threshold is the median of gradient norms at initialization. We update the full gradient norms twice per epoch. More details about the models and hyperparameters are in Appendix B.

4.1 Per-Instance Privacy Parameters Vary Significantly

Figure 1 shows the per-instance privacy parameters on CIFAR-10 and MNIST. Figure 4 shows the per-instance privacy parameters on UTKFace. The privacy parameters vary across a large range on all datasets. For example, when running gender classification on the UTKFace dataset, the maximum ϵ_i value is 4.5 while the minimum ϵ_i value is only 0.1. We also observe that, for easier tasks, more examples enjoy stronger privacy guarantees. For example, $\sim 30\%$ of examples reach the worst-case ϵ on CIFAR-10 while only $\sim 3\%$ do so on MNIST. This may because the loss decreases quickly when the task is easy, resulting in gradient norms also decreasing and thus a reduced privacy loss.

5 Privacy Loss is Unequal Across Different Subgroups

We investigate the difference of privacy loss among different subgroups. We first empirically show the privacy parameter of one example correlates well with its loss. Then we show example groups with higher loss (i.e., groups underserved by the model) also have worse privacy parameters. Finally, we run membership inference attacks to show the computed privacy parameters reflect empirical privacy risks.

5.1 Privacy Parameters Correlate with Loss

We show a strong correlation between the privacy parameters and loss values. Based on the analysis in Section 2, the privacy parameter of an example directly correlates with its gradient norms across training. The gradient norms further depend on the loss values. To verify this correlation, we plot the average loss and ϵ of different groups on CIFAR-10 and MNIST in Figure 5. We use two ways to divide examples into different groups. The first takes the average loss over training and the second takes the loss during the last epoch. We then sort the examples based on loss values and divide the sorted examples into ten even groups. The correlation between loss values and privacy parameters is apparent on both datasets in Figure 5.

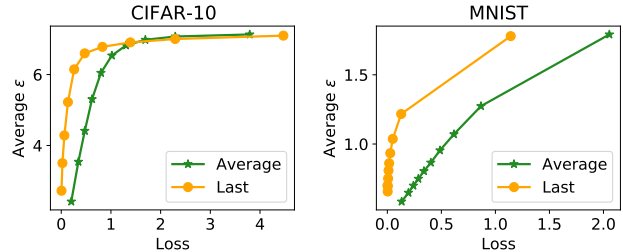


Figure 5: Correlation between ϵ and loss.

5.2 Groups Are Simultaneously Underserved in Both Accuracy and Privacy

It is well-documented that machine learning models may have large differences in accuracy on different subgroups [BG18, BPS19]. Our finding demonstrates that this disparity is simultaneous in terms of both loss values *and* privacy risks. We empirically verify this by plotting the average ϵ and loss values of different subgroups. For loss values, we use both the average loss across training and loss during the last epoch. For CIFAR-10 and MNIST, the subgroups are the data from different classes, while for UTKFace, the subgroups are the data from different races.

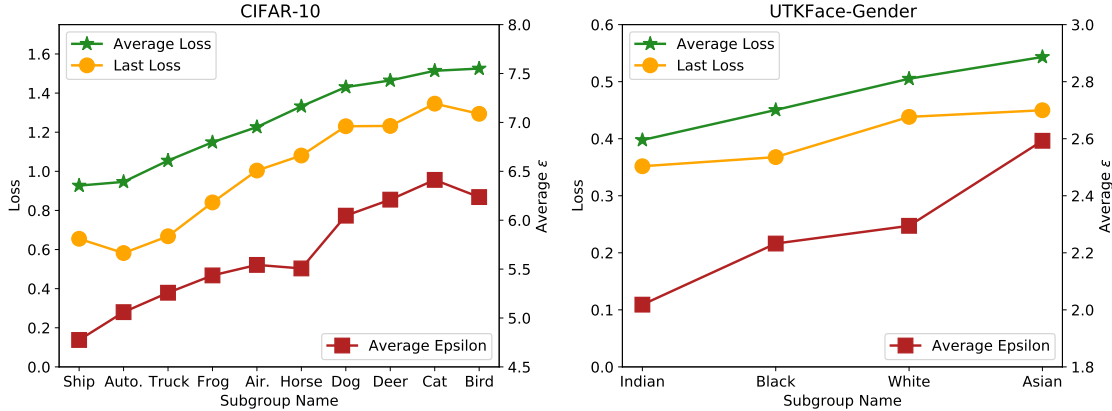


Figure 6: Loss and ϵ of different groups. Groups with higher loss have worse privacy in general.

We plot the results on CIFAR-10 and UTKFace-Gender in Figure 6. The results on MNIST and UTKFace-Age are in Appendix F. The subgroups are sorted based on their average loss values. The loss still correlates well with ϵ in general and subgroups with higher loss do tend to have higher privacy parameters. On CIFAR-10, the average ϵ of the ‘Cat’ class (which has the highest average loss at the last epoch) is 6.43 while the average ϵ of the class with the lowest loss (‘Ship’) is only 4.87. The observation is similar on UTKFace-Gender, in which the average ϵ of the subgroup with the highest loss is 2.62 while the average ϵ of the subgroup with the lowest loss is 2.04. To the best of our knowledge, our work is the first to reveal this simultaneous disparity.

5.3 Privacy Parameters Reflect Empirical Privacy Risks

We run membership inference (MI) attacks to verify whether examples with larger privacy parameters have higher privacy risks in practice. We use a simple loss-threshold attack that predicts an example is a member if its loss value is smaller than a prespecified threshold [SDS⁺19]. Previous works show that even large privacy parameters are sufficient to defend against such attacks [CLE⁺19, YZC⁺21]. In order to better observe the difference in privacy risks, we also include models trained without differential privacy as target models. For each data subgroup, we use its whole test set and a random subset of training set so the numbers of training and test loss values are balanced. We further split the data into two subsets evenly to find the optimal threshold on one and report the success rate on another.

The results on CIFAR-10 and UTKFace-Gender are in Figure 6. The results on MNIST and UTKFace-Age are in Appendix F. The subgroups are sorted based on their average ϵ . When the models are trained with DP, all attack success rates are close to random guessing (50%), as anticipated. Although the attack we use can not show the disparity in this case, we note that there are more powerful attacks whose success rates are closer to the lower bound that DP offers [JUO20, NST⁺21]. On the other hand, the difference in privacy risks is clear when models are trained without DP. On CIFAR-10, the MI success rate is 79.7% on the Cat class (which has the worst average ϵ) while is only 61.4% on the Ship class (which has the best average ϵ). These results suggest that the ϵ values reflect empirical privacy risks which could vary significantly on different subgroups.

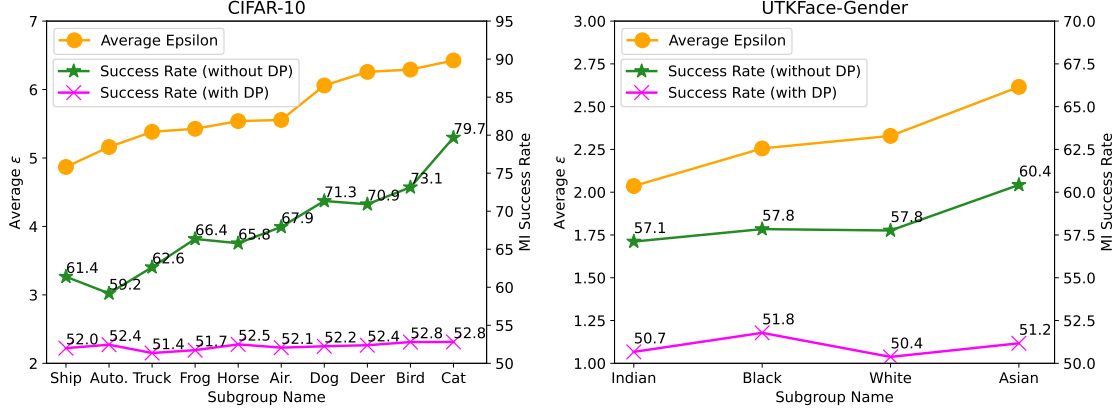


Figure 7: Average ϵ and membership inference success rates on different subgroups.

6 Conclusion

We propose an algorithm to compute per-instance privacy parameters for DP-SGD. The algorithm can give accurate per-instance privacy parameters even when extra computation is very small (for the case $K = 0$). We use this new algorithm to examine per-instance privacy risks for examples in several datasets. Significantly, we find that groups with worse utility also suffer from worse privacy. Our paper reveals the complex while interesting relation among utility, fairness and privacy, which may inspire new studies of jointly considering these factors to build trustworthy systems.

References

- [ACG⁺16] 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 Conference on Computer and Communications Security, CCS '16*, pages 308–318, New York, NY, USA, 2016. ACM.
- [AGG⁺21] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.
- [ATMR21] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.
- [BBG18] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pages 6277–6287. Curran Associates, Inc., 2018.
- [BDLS20] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J. Su. Deep learning with Gaussian differential privacy. *Harvard Data Science Review*, 2(3), 2020.
- [BG18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability, and Transparency, FAT* '18*, pages 77–91. JMLR, Inc., 2018.
- [BGRK19] Daniel Bernau, Philip-William Grassal, Jonas Robl, and Florian Kerschbaum. Assessing differentially private deep learning with membership inference. *arXiv preprint arXiv:1912.11328*, 2019.
- [BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 15479–15488. Curran Associates, Inc., 2019.

- [BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science, FOCS '14*, pages 464–473, Washington, DC, USA, 2014. IEEE Computer Society.
- [CCN⁺21] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021.
- [CCTCP21] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning, ICML '21*, pages 1964–1974. JMLR, Inc., 2021.
- [CLE⁺19] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, USENIX Security '19, pages 267–284. USENIX Association, 2019.
- [CWH20] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 13773–13782. Curran Associates, Inc., 2020.
- [DBH⁺22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.
- [DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography, TCC '06*, pages 265–284, Berlin, Heidelberg, 2006. Springer.
- [DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- [FZ20] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 2881–2891. Curran Associates, Inc., 2020.
- [FZ21] Vitaly Feldman and Tijana Zrnica. Individual privacy accounting via a Renyi filter. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.
- [GAW⁺22] Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. In *Proceedings of the 2022 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '22*, Washington, DC, USA, 2022. IEEE Computer Society.
- [GLW21] Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '16*, pages 770–778, Washington, DC, USA, 2016. IEEE Computer Society.
- [JUO20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private SGD? In *Advances in Neural Information Processing Systems 33*, NeurIPS '20. Curran Associates, Inc., 2020.
- [JYC15] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *International Conference on Data Engineering (ICDE)*, 2015.
- [Koh17] Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML '17*, pages 1885–1894. JMLR, Inc., 2017.
- [Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

- [KYC⁺19] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. Disparate vulnerability to membership inference attacks. *arXiv preprint arXiv:1906.00389*, 2019.
- [LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [LBG17] Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. Towards measuring membership privacy. *arXiv preprint arXiv:1712.09136*, 2017.
- [LTLH22] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR ’22, 2022.
- [MB22] Christopher Mühl and Franziska Boenisch. Personalized PATE: Differential privacy for machine learning with individual privacy guarantees. *arXiv preprint arXiv:2202.10517*, 2022.
- [Mir17] Ilya Mironov. Rényi differential privacy. In *Proceedings of the 30th IEEE Computer Security Foundations Symposium*, CSF ’17, pages 263–275, Washington, DC, USA, 2017. IEEE Computer Society.
- [MTKC22] Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2202.10530*, 2022.
- [MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.
- [NHS22] Frederik Noe, Rasmus Herskind, and Anders Søgaard. Exploring the unfairness of dp-sgd across settings. *arXiv preprint arXiv:2202.12058*, 2022.
- [NST⁺21] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. *arXiv preprint arXiv:2101.04535*, 2021.
- [PAE⁺17] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR ’17, 2017.
- [PBHTNS⁺22] Victor Petró Bach Hansen, Atula Tejaswi Neerkaje, Ramit Sawhney, Lucie Flek, and Anders Søgaard. The impact of differential privacy on group disparity mitigation. *arXiv e-prints*, 2022.
- [RW21] Rachel Redberg and Yu-Xiang Wang. Privately publishable per-instance privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS ’21. Curran Associates, Inc., 2021.
- [SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP ’13, pages 245–248, Washington, DC, USA, 2013. IEEE Computer Society.
- [SDS⁺19] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, ICML ’19, pages 5558–5567. JMLR, Inc., 2019.
- [SPGG21] Vinith M Suriyakumar, Nicolas Papernot, Anna Goldenberg, and Marzyeh Ghassemi. Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’21, pages 723–734. JMLR, Inc., 2021.
- [SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, SP ’17, pages 3–18, Washington, DC, USA, 2017. IEEE Computer Society.

- [SSTT21] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private GLMs. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, AISTATS '21, pages 2638–2646. JMLR, Inc., 2021.
- [TSJ⁺22] Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. *arXiv preprint arXiv:2204.00032*, 2022.
- [Wan19] Yu-Xiang Wang. Per-instance differential privacy. *The Journal of Privacy and Confidentiality*, 2019.
- [WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, AISTATS '19, pages 1226–1235. JMLR, Inc., 2019.
- [YNB⁺22] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.
- [YZC⁺21] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale privacy learning via low-rank reparametrization. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21, pages 12208–12218. JMLR, Inc., 2021.
- [ZSQ17] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '17, pages 5810–5818, Washington, DC, USA, 2017. IEEE Computer Society.
- [ZW19] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled Rényi differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, ICML '19, pages 7634–7642. JMLR, Inc., 2019.

A More Background on Rényi Differential Privacy

In this work, privacy loss is composited through Rényi Differential Privacy (RDP) and the overall privacy loss is converted into (ϵ, δ) -DP after training [Mir17]. Here we first introduce the definition of per-instance Rényi Differential Privacy. Then we give the composition theorem for per-instance RDP and the conversion rule from RDP to (ϵ, δ) -DP.

A.1 Per-Instance Rényi Differential Privacy

RDP uses the Rényi divergences of different orders between two output distributions to measure privacy. Let $D_\lambda(\mu||v)$ be the Rényi divergence of order λ between μ and v and $D_\lambda^{\leftrightarrow}(\mu||v) = \max(D_\lambda(\mu||v), D_\lambda(v||\mu))$ be the maximum of the two directions. The definition of per-instance RDP is as follows.

Definition 3. [Per-instance RDP] Fix a dataset \mathbb{D} and $\mathbb{D}' = \mathbb{D} \setminus \{\mathbf{d}\}$. An algorithm \mathcal{A} satisfies (λ, γ) -RDP for (\mathbb{D}, \mathbf{d}) if for any subset of outputs S it holds that

$$D_\lambda^{\leftrightarrow}(\mathcal{A}(\mathbb{D})||\mathcal{A}(\mathbb{D}')) \leq \gamma.$$

We use the numerical tool in [MTZ19] to compute $D_\lambda^{\leftrightarrow}(\mathcal{A}(\mathbb{D})||\mathcal{A}(\mathbb{D}'))$ for every \mathbf{d} at every iteration. The results at different iterations are composed and then converted into (ϵ, δ) -DP.

A.2 Composition and Conversion Rules for Per-instance RDP

We use the composition theorem in [FZ21] that allows privacy parameters are chosen adaptively. Let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be a sequence of algorithms and $a_t = \mathcal{A}_t(a_1, \dots, a_{t-1}, \mathbb{D})$, where a_t is the output of the t th algorithm. Further let

Table 1: Comparison between the test accuracy of using per-example clipping thresholds and that of using a single maximum clipping threshold.

	CIFAR-10	MNIST
Per-example	65.62 (± 0.51)	97.32 (± 0.16)
Single	65.66 (± 0.68)	97.26 (± 0.11)

$\mathcal{A}^{(t)}(\cdot) = (\mathcal{A}_1(\cdot), \mathcal{A}_2(a_1, \cdot), \dots, \mathcal{A}_t(a_1, a_2, \dots, \cdot))$ that composes $\{\mathcal{A}_i\}_{i=1}^t$. For a fixed pair of neighboring datasets \mathbb{D} and $\mathbb{D}' = \mathbb{D} \setminus \{d\}$, the privacy parameter of order λ at the t th algorithm is

$$\rho_t = D_{\lambda}^{\leftrightarrow}(\mathcal{A}_t(a_1, \dots, a_{t-1}, \mathbb{D}) || \mathcal{A}_t(a_1, \dots, a_{t-1}, \mathbb{D}')). \quad (3)$$

Theorem A.1 (A special case of Theorem 3.1 in [FZ21]). *Fix any $B \geq 0, \lambda \geq 1$, and a pair of neighboring datasets $(\mathbb{D}, \mathbb{D}')$. For any sequence of algorithms $\mathcal{A}_1, \dots, \mathcal{A}_k$, if $\sum_{t=1}^k \rho_t \leq B$ holds almost surely, where the sequence ρ_1, \dots, ρ_k is defined in Equation (3), then the adaptive composition $\mathcal{A}^{(k)}$ satisfies*

$$D_{\lambda}^{\leftrightarrow}(\mathcal{A}^{(k)}(\mathbb{D}) || \mathcal{A}^{(k)}(\mathbb{D}')) \leq B.$$

Theorem A.1 states for adaptively chosen privacy parameters, we can still add up the privacy parameters at different steps to get the overall privacy guarantee. After training, we use Theorem A.2 to convert the overall RDP of an instance into (ε, δ) -DP [Mir17].

Theorem A.2 (Convert per-instance RDP into per-instance (ε, δ) -DP [Mir17]). *If \mathcal{A} obeys (λ, γ) -RDP for (\mathbb{D}, d) , then \mathcal{A} obeys $(\gamma + \log(1/\delta)/(\lambda - 1), \delta)$ -DP with respect to (\mathbb{D}, d) for all $0 < \delta < 1$.*

We compute $D_{\lambda}^{\leftrightarrow}(\mathcal{A}(\mathbb{D}) || \mathcal{A}(\mathbb{D}'))$ with different orders of λ in our experiments and choose the tightest (ε, δ) -DP bound from the conversion results.

B More Details on Hyperparameters

The noise multipliers are 3.2, 6.0, and 1.5 for CIFAR-10, MNIST, and UTKFace, respectively. The standard deviation of noise in Algorithm 1 is the noise multiplier times the maximum clipping threshold. The batchsize is 4000 for CIFAR-10/MNIST and 200 for UTKFace. The training epoch is 200 for CIFAR-10 and 100 for MNIST and UTKFace. For ResNet-20 models on CIFAR-10 and MNIST, we replace batch normalization with group normalization. For ResNet-20 models on UTKFace, we freeze the batch normalization layers of the pre-trained model. We compute Rényi divergence with integer orders up to 256.

C Per-example Clipping Does Not Affect Accuracy

Here we run experiments to check the influence of per-example clipping thresholds on utility. Algorithm 1 uses per-example clipping thresholds to ensure the computed privacy parameters are valid privacy guarantees. If the clipping thresholds are close to the actual gradient norms, then the clipped results are close to those of using a single maximum clipping threshold. However, if the estimations of gradient norms are not accurate, per-example thresholds would clip more signal than using a single maximum threshold.

We compare the accuracy of using the maximum clipping threshold and that of using per-example clipping thresholds. The results on CIFAR-10 and MNIST are in Table 1. All the per-example clipping thresholds are updated on-the-fly with $K = 0$. We repeat the experiment four times with different random seeds. Other setups are the same as those in Section 4. The results suggest that using per-example clipping thresholds in Algorithm 1 does not affect the accuracy.

D On the Influence of the Maximum Clipping Threshold on Privacy

As discussed in Section 3.1, the value of the maximum clipping threshold C would affect per-instance privacy parameters in Algorithm 1. Here we run experiments with different values of C on CIFAR-10. Let M be the median of gradient

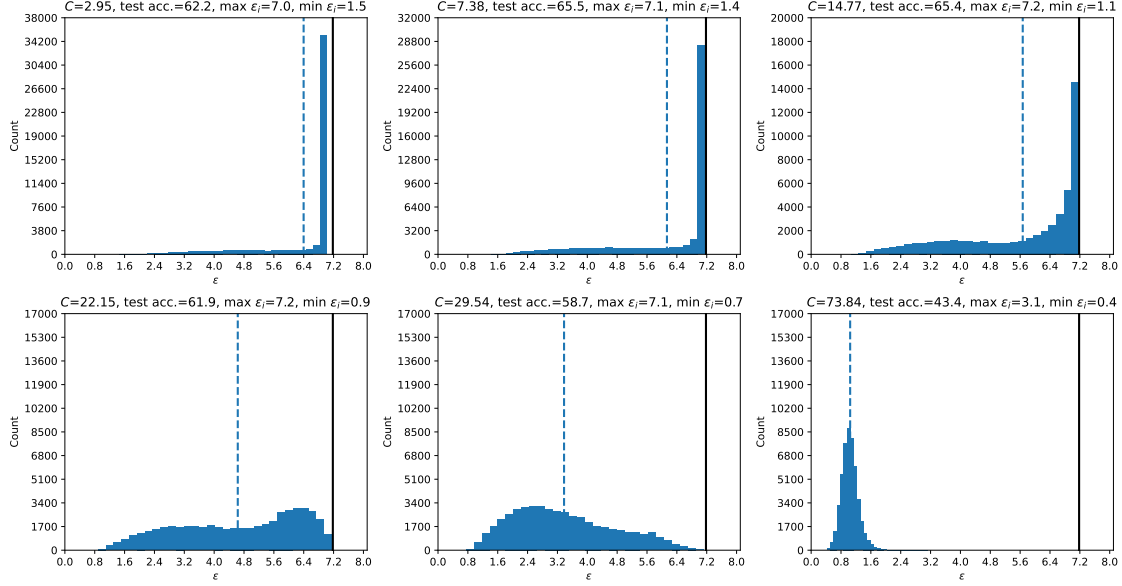


Figure 8: Distribution of privacy parameters on CIFAR-10 with different values of C . The median of gradient norms at initialization is 14.77. The black solid line indicates the original privacy parameter of DP-SGD for all instances. The maximum ϵ_i values in the first two plots do not match the bound of the original analysis because of the rounding operation in Algorithm 1.

Table 2: Populational statistics of per-instance privacy parameters on MNIST. The average estimation error rate is 0.77%. The value of δ is 1×10^{-5} .

MNIST	Average	0.1-quantile	0.3-quantile	Median	0.7-quantile	0.9-quantile
Non-private	0.906	0.563	0.672	0.790	0.967	1.422
$\epsilon = 0.1$	0.907	0.562	0.670	0.791	0.969	1.369

norms at initialization, we choose C from the list $[0.2M, 0.5M, M, 1.5M, 2M, 5M]$. Other experimental setup is the same as that in Section 4.

We plot the results in Figure 8. The variation in privacy parameters increases with the value of C . When $C = 0.2M$, nearly 70% datapoints reach the worst privacy parameter while only 3% datapoints reach the worst parameter when $C = 1.5M$. Moreover, when $C > 1.5M$, some privacy parameters do not reach the worst parameter after training. For instance, the maximum privacy parameter is only 3.1 when $C = 5M$ while the ϵ of the original analysis is 7.2.

E Release Populational Statistics of Per-instance Privacy Parameters

The per-instance privacy parameters computed by Algorithm 1 are sensitive and hence can not be directly released to the public. Here we show the populational statistics of per-instance parameters can be released with minor cost in privacy. Specifically, we compute the average and quantiles of the ϵ values with differential privacy. The sensitivity of ϵ is the value from the original analysis. For average, we release the noisy aggregation through the Gaussian mechanism in [DR14]. For quantiles, we solve the objective function in [ATMR21] with 20 steps of full batch gradient descent. The privacy loss of running gradient descents is composed under RDP and then converted into (ϵ, δ) -DP. The results on MNIST and CIFAR-10 are in Table 2 and Table 3 respectively. The released statistics are close to the actual values on both datasets.

Table 3: Populational statistics of per-instance privacy parameters on CIFAR-10. The average estimation error rate is 1.18%. The value of δ is 1×10^{-5} .

CIFAR-10	Average	0.1-quantile	0.3-quantile	Median	0.7-quantile	0.9-quantile
Non-private	5.670	2.942	4.822	6.398	6.959	7.132
$\varepsilon = 0.1$	5.669	3.125	4.818	6.390	6.977	7.159

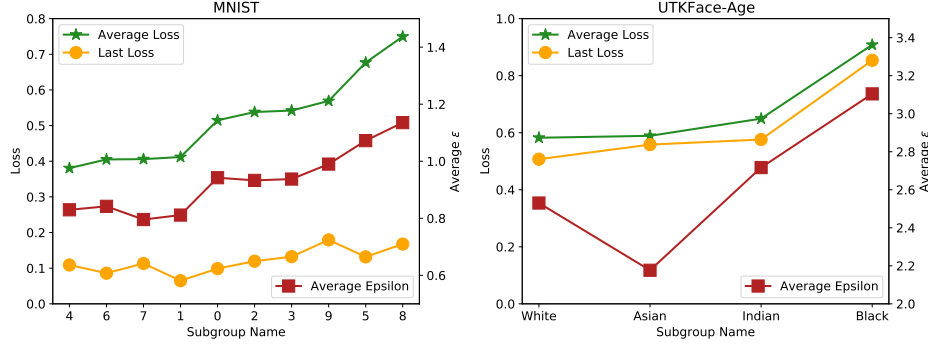


Figure 9: Loss and ε of different subgroups. Subgroups with higher loss have worse privacy in general. Experimental setup is the same as that in Section 5.

F More Plots Showing Privacy Risk is Unequal Across Different Subgroups

We plot the average loss and ε of different subgroups on MNIST and UTKFace-age in Figure 9. The subgroups are sorted based on their average loss values. The observation is similar to that in Section 5. Subgroups with higher loss also have higher privacy parameters in general. On UTKFace-Age, the average ε of the subgroup with the highest loss (Black) is 3.12 while the average ε of the subgroup with the lowest loss (Asian) is 2.54.

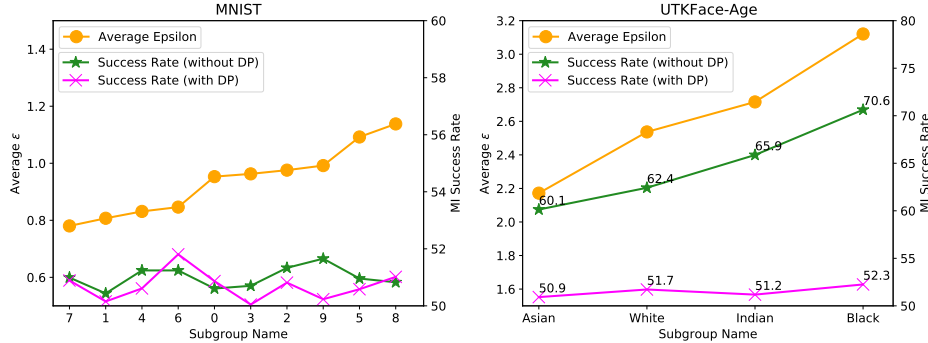


Figure 10: Average ε and membership inference success rates of different subgroups. Experimental setup is the same as that in Section 5.

The membership inference attack success rates and average ε on MNIST and UTKFace-Age are in Figure 10. Subgroups are sorted based on their average ε . On MNIST, the attack success rates are close to that of random guessing (50%) no matter the models are trained with DP or not. This may because the generalization gap is very small on MNIST, i.e., test accuracy $>99\%$ when trained without DP, so it is hard to distinguish training loss distribution and test loss distribution. On UTKface-Age, the difference in attack success rates is clear when the model is trained without DP. For example, the attack success rate on the Black subgroup (which has the highest average ε) is 70.6% while the attack success rate on the Asian subgroup (which has the lowest average ε) is only 60.1%.