INDIVIDUAL PRIVACY ACCOUNTING FOR DIFFERENTIALLY PRIVATE STOCHASTIC GRADIENT DESCENT

Anonymous authors
Paper under double-blind review

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 privacy guarantees for individual examples when releasing models trained by DP-SGD. We use our algorithm to investigate individual privacy parameters across a number of datasets. We find that most examples enjoy stronger privacy guarantees than the worst-case bound. We further discover that the training loss and the privacy parameter of an example are well-correlated. This implies groups that are underserved in terms of model utility are simultaneously underserved in terms of privacy guarantee. For example, on CIFAR-10, the average ε of the class with the lowest test accuracy is 43.6% higher than that of the class with the highest accuracy.

1 INTRODUCTION

Differential privacy is a strong notion of data privacy, enabling rich forms of privacy-preserving data analysis (Dwork & Roth, 2014). 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 (Papernot et al., 2017; Bu et al., 2020; Yu et al., 2022; Anil et al., 2021; Li et al., 2022; Golatkar et al., 2022; Mehta et al., 2022; De et al., 2022; Bu et al., 2022). Models trained with differential privacy not only provide theoretical privacy guarantee to their data but also are more robust against empirical attacks (Bernau et al., 2019; Carlini et al., 2019; Jagielski et al., 2020; Nasr et al., 2021).

Differentially private stochastic gradient descent (DP-SGD) is the de-facto choice for differentially private deep learning (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016). DP-SGD first clips individual gradients and then adds Gaussian noise to the average of clipped 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 (Koh & Liang, 2017; Feldman & Zhang, 2020). 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 accurately estimate individual privacy parameters of models trained by DP-SGD. Our privacy guarantee adapts to the training trajectory of one execution of DP-SGD to provide a precise characterization of privacy cost (see Section 2.1 for more details). Inspecting individual privacy parameters allows us to better understand example-wise impacts. It turns out that, for common benchmarks, many examples experience much stronger privacy guarantee than the worst-case bound. To illustrate this, we plot the individual privacy parameters of MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky, 2009), and UTKFace (Zhang et al., 2017) in Figure 1. Experimental details, as well as more results, can be found in Section 4 and 5. The disparity in individual privacy guarantees naturally arises when running DP-SGD. To the best of our knowledge, our investigation is the first to explicitly reveal such disparity.

We propose two techniques to make individual privacy accounting viable for DP-SGD. First, we maintain estimates of the gradient norms for all examples so the individual privacy costs can be computed accurately at every update. Second, we round the gradient norms with a small precision r to control the number of different privacy costs, which need to be computed numerically. We explain

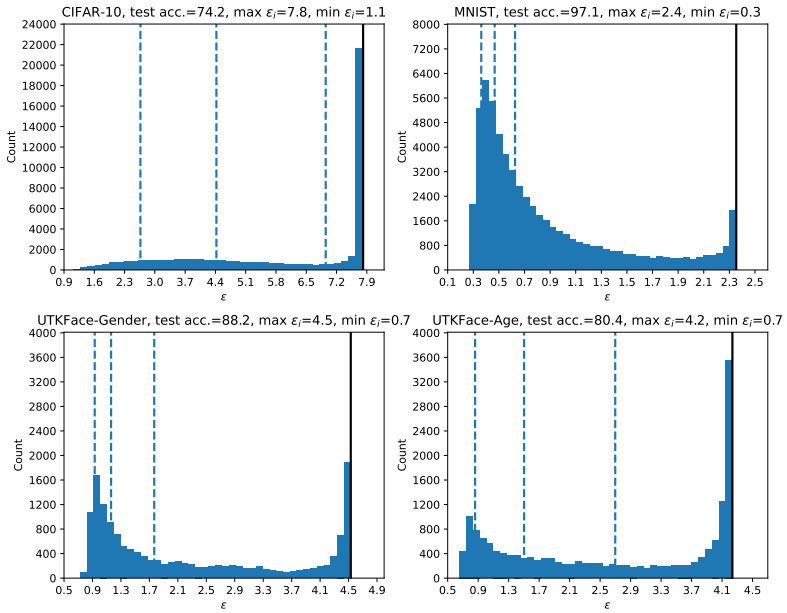

Figure 1: Individual privacy parameters of models trained by DP-SGD. The value of δ is 1×10^{-5}. The dashed lines indicate 10%, 30%, and 50% of datapoints. The black solid line shows the privacy parameter of the original analysis.

why these two techniques are necessary in Section 2. More details of the proposed algorithm, as well as methods to release individual privacy parameters, are in Section 3.

We further demonstrate a strong correlation between the privacy parameter of an example and its final training loss. We find that examples with higher training loss also have higher privacy parameters in general. This suggests that 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 (Buolamwini & Gebru, 2018), and that these disparities are amplified when models are trained privately Bagdasaryan et al. (2019); Suriyakumar et al. (2021); Hansen et al. (2022); Noe et al. (2022), we are the first to show that the privacy guarantee *and* utility are 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 guarantee for all training examples. For instance, when running gender classification on UTKFace, the average ε of the race with the lowest test accuracy is 25% higher than that of the race with the highest accuracy. We also study the disparity in privacy when models are trained without differential privacy, which may be of independent interest to the community. We use the success rates of membership inference attacks to measure privacy in this case and show groups with worse accuracy suffer from higher privacy risks.

1.1 Related Work

Several works have explored example-wise privacy guarantees in differentially private learning. Jorgensen et al. (2015) propose *personalized differential privacy* that provides pre-specified individual privacy parameters which are independent of the learning algorithm, e.g., users can choose different levels of privacy guarantees based on their sensitivities to privacy leakage (Mühl & Boenisch, 2022). A recent line of works also uses the variation in example-wise sensitivities that naturally arise in learning to study example-wise privacy. *Per-instance differential privacy* captures the privacy parameter of a target example with respect to a fixed training set (Wang, 2019; Redberg & Wang, 2021; Golatkar et al., 2022). Feldman & Zrnic (2021) design an individual *Rényi differential privacy filter*. The filter stops when the accumulated cost reaches a target budget that is defined before training. It allows examples with smaller per-step privacy costs to run for more steps. The final privacy guarantee offered by the filter is still the worst-case over all possible outputs as the predefined budget has to be independent of the algorithm outcomes. In this work, we propose output-specific differential privacy and give an efficient algorithm to compute individual guarantees of DP-SGD. We

further discover that the disparity in individual privacy parameters correlates well with the disparity in utility.

2 PRELIMINARIES

We first give some background on differential privacy. Then we briefly introduce the privacy analysis of DP-SGD and highlight the challenges in computing individual privacy guarantees. Finally, we argue that providing the same privacy bound to all examples is not ideal because of the variation in individual gradient norms.

2.1 BACKGROUND ON DIFFERENTIALLY PRIVATE LEARNING

Differential privacy builds on the notion of neighboring datasets. A dataset $\mathbb{D} = \{\boldsymbol{d}_i\}_{i=1}^n$ 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 $\mathbb{D}$. The indiviudal privacy guarantees in this work take the form of (ε, δ)-differential privacy.

Definition 1. *[Individual (ε, δ)-DP] For a datapoint $\boldsymbol{d}$, let $\mathbb{D}$ be an arbitrary dataset and $\mathbb{D}' = \mathbb{D} \cup \{\boldsymbol{d}\}$ be its neighboring dataset. An algorithm $\mathcal{A}$ satisfies $(\varepsilon(\boldsymbol{d}), \delta)$-individual DP if for any subset of outputs $\mathbb{S} \subset Range(\mathcal{A})$ it holds that*

$$\Pr[\mathcal{A}(\mathbb{D}) \in \mathbb{S}] \leq e^{\varepsilon(\boldsymbol{d})} \Pr[\mathcal{A}(\mathbb{D}') \in \mathbb{S}] + \delta \text{ and } \Pr[\mathcal{A}(\mathbb{D}') \in \mathbb{S}] \leq e^{\varepsilon(\boldsymbol{d})} \Pr[\mathcal{A}(\mathbb{D}) \in \mathbb{S}] + \delta.$$

We further allow the privacy parameter ε to be a function of a subset of outcomes to provide a sharper characterization of privacy. We term this variant as *output-specific* (ε, δ)-DP. Our output-specific DP shares similar insights as the *ex-post* DP in Ligett et al. (2017); Redberg & Wang (2021) as both definitions tailor the privacy guarantee to algorithm outcomes. One important difference is that output-specific DP is ex-ante as it gives the guarantee before the final outcome is sampled.

Definition 2. *[Output-specific individual (ε, δ)-DP] Fix a datapoint $\boldsymbol{d}$ and a set of outcomes $\mathbb{A} \subset Range(\mathcal{A})$, let $\mathbb{D}$ be an arbitrary dataset and $\mathbb{D}' = \mathbb{D} \cup \{\boldsymbol{d}\}$. An algorithm $\mathcal{A}$ satisfies $(\varepsilon(\mathbb{A}, \boldsymbol{d}), \delta)$-output-specific individual DP for $\boldsymbol{d}$ at $\mathbb{A}$ if for any $\mathbb{S} \subset \mathbb{A}$ it holds that*

$$\Pr[\mathcal{A}(\mathbb{D}) \in \mathbb{S}] \leq e^{\varepsilon(\mathbb{A}, \boldsymbol{d})} \Pr[\mathcal{A}(\mathbb{D}') \in \mathbb{S}] + \delta \text{ and } \Pr[\mathcal{A}(\mathbb{D}') \in \mathbb{S}] \leq e^{\varepsilon(\mathbb{A}, \boldsymbol{d})} \Pr[\mathcal{A}(\mathbb{D}) \in \mathbb{S}] + \delta.$$

Definition 2 has the same semantics as Definition 1 once the algorithm's outcome belongs to $\mathbb{A}$ is known. It is a strict generalization of (ε, δ)-DP as one can recover (ε, δ)-DP by maximizing $\varepsilon(\mathbb{A}, \boldsymbol{d})$ over $\mathbb{A}$ and $\boldsymbol{d}$. Making this generalization is crucial for us to precisely compute the privacy parameters of models trained with DP-SGD. The output of DP-SGD is a sequence of models $\{\theta_t\}_{t=1}^T$, where T is number of iterations. The privacy risk at step t highly depends on the training trajectory (the first $t-1$ models). For example, one can adversarially modify the training data and hence change the training trajectory to maximize the privacy risk of a target example (Tramèr et al., 2022). With Definition 2, we can analyze the privacy guarantee of DP-SGD with regards to a fixed training trajectory, which specifies a subset of DP-SGD's outcomes. In comparison, canonical DP analyzes the worst-case privacy over all possible outcomes which could give loose privacy parameters.

A common approach for doing deep learning with differential privacy is to make each update differentially private instead of protecting the trained model directly. The composition property of differential privacy allows us to reason about the overall privacy of running several such steps. We give a simple example to illustrate how to privatize each update. Suppose we take the sum of all gradients $\boldsymbol{v} = \sum_{i=1}^n \boldsymbol{g}_i$ from dataset $\mathbb{D}$. Without loss of generality, further assume we add an arbitrary example $\boldsymbol{d}'$ to obtain a neighboring dataset $\mathbb{D}'$. The summed gradient becomes $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{g}'$, where $\boldsymbol{g}'$ is the gradient of $\boldsymbol{d}'$. If we add isotropic Gaussian noise with variance σ^2, then the output distributions of two neighboring datasets are

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

We then can bound the divergence between two Gaussian distributions to prove differential privacy, e.g., through *Rényi differential privacy (RDP)* (Mironov, 2017). We give the definition of individual RDP in Appendix A. The expectations of $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ differ by $\boldsymbol{g}'$. A larger gradient leads to a larger difference and hence a worse privacy parameter.

2.2 Challenges of Computing Individual 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*. 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. How subsampling complicates the theoretical privacy analysis has been studied extensively (Abadi et al., 2016; Balle et al., 2018; Mironov et al., 2019; Zhu & Wang, 2019; Wang et al., 2019). In this section, we focus on how subsampling complicates the empirical computation of individual privacy parameters.

Before we expand on these difficulties, we first describe the output distributions of neighboring datasets in DP-SGD (Abadi et al., 2016). Poisson sampling is assumed, i.e., each example is sampled independently with probability p. Let $\boldsymbol{v} = \sum_{i \in \mathbb{M}} \boldsymbol{g}_i$ 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 $\boldsymbol{g}'$ added. Because of Poisson sampling, the output is exactly $\boldsymbol{v}$ with probability $1-p$ ($\boldsymbol{g}'$ is not sampled) and is $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{g}'$ with probability p ($\boldsymbol{g}'$ is sampled). Suppose we still add isotropic Gaussian noise, the output distributions of two neighboring datasets are

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

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

With Equation (1) and (2), we explain the challenges in computing individual privacy parameters.

2.2.1 Full Batch Gradient Norms Are Required at Every Iteration

There is some privacy cost for $\boldsymbol{d}'$ 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 cost is determined by $\|\boldsymbol{g}'\|$. Therefore, we need accurate gradient norms of all examples to compute accurate privacy costs at every iteration. However, when running SGD, we only compute 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 Individual Privacy Parameters is Huge

The density function of $\mathcal{A}(\mathbb{D}')$ is a mixture of two Gaussian distributions. Abadi et al. (2016) compute the Rényi divergence between $\mathcal{A}(\mathbb{D})$ and $\mathcal{A}(\mathbb{D}')$ numerically to get tight privacy parameters. Although there are some asymptotic bounds, those bounds are looser than numerical computation, and thus such numerical computations are necessary in practice (Abadi et al., 2016; Wang et al., 2019; Mironov et al., 2019; Gopi et al., 2021). In the classic analysis, there is only one numerical computation as all examples have the same privacy cost over all iterations. However, naive computation of individual privacy parameters would require up to $n \times T$ numerical computations, where n is the dataset size and T is the number of iterations.

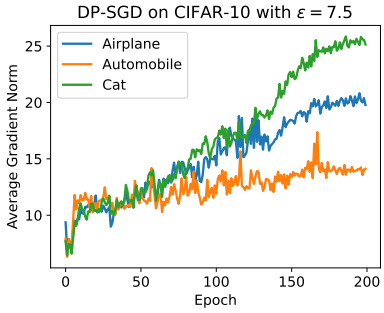

Figure 2: Gradient norms on CIFAR-10.

2.3 An Observation: Gradient Norms in Deep Learning Vary Significantly

We show gradient norms vary significantly to demonstrate that different examples experience very difference privacy costs when training with DP-SGD. We train the standard ResNet-20 model in He et al. (2016) on CIFAR-10. The maximum clipping threshold is the median of gradient norms at initialization. We plot the average norms of three different classes in Figure 2. The gradient norms of different classes show significant stratification. Such stratification naturally leads to different individual privacy costs. This suggests that quantifying individual privacy parameters may be valuable despite the aforementioned challenges.

3 Deep Learning with Individual Privacy

We give an efficient algorithm (Algorithm 1) to estimate individual privacy parameters for DP-SGD. The privacy analysis of Algorithm 1 is in Section 3.1. We perform two modifications to make individual privacy accounting feasible with small computational overhead. First, we compute full

batch gradient norms once in a while, e.g., at the beginning of every epoch, and use the results to estimate the gradient norms for subsequent iterations. We show the estimations of gradient norms are accurate in Section 3.2. Additionally, we round the gradient norms to a given precision so the number of numerical computations is independent of the dataset size. More details of this modification are in Section 3.3. Finally, we discuss how to make use of the individual privacy parameters in Section 3.4.

In Algorithm 1, we clip individual gradients with estimations of gradient norms (referred to as *individual clipping*). This is different from the vanilla DP-SGD that uses the maximum threshold to clip all examples (Abadi et al., 2016). Individual clipping gives us exact upper bounds on gradient norms and hence we have exact bounds on privacy costs. Our analysis is also applicable to vanilla DP-SGD though the privacy parameters in this case become estimates of exact guarantees, which is inevitable because one can not have exact bounds on gradient norms at every iteration when running SGD. We report the results for vanilla DP-SGD in Appendix C.2. Our observations include: (1) the estimates of privacy guarantees for vanilla DP-SGD are very close to the exact ones, (2) the privacy parameters of vanilla DP-SGD are very close to the privacy parameters of Algorithm 1.

Algorithm 1: Deep Learning with Individual 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, number of epochs E.

Let $\{C^{(i)}\}_{i=1}^n$ be estimated gradient norms of all examples and initialize $C^{(i)} = C$.
Let $\mathbb{C} = \{r, 2r, 3r, \ldots, C\}$ be all possible norms under rounding.
for $c \in \mathbb{C}$ **do**
  Compute Rényi divergences at different orders between Equation (1), (2) and store the results in $\boldsymbol{\rho}_c$.
end
Let $\{\boldsymbol{o}^{(i)} = \mathbf{0}\}_{i=1}^n$ be the accumulated Rényi divergences at different orders.
for $e = 1$ *to* E **do**
  for $t' = 1$ *to* $\lfloor 1/p \rfloor$ **do**
    $t = t' + \lfloor 1/p \rfloor \times (e-1)$ *//Global step count.*
    if $t \bmod \lfloor 1/pK \rfloor = 0$ **then**
      Compute full batch gradient norms $\{S^{(i)}\}_{i=1}^n$.
      Update gradient estimates with rounded norms $\{C^{(i)} = \arg\min_{c \in \mathbb{C}}(|c - S^{(i)}|)\}_{i=1}^n$.
    end
    Sample a minibatch of gradients $\{\boldsymbol{g}^{(I_j)}\}_{j=1}^{|I|}$, where I is the sampled indices.
    Clip gradients $\bar{\boldsymbol{g}}^{(I_j)} = clip(\boldsymbol{g}^{(I_j)}, C^{(I_j)})$.
    Update model $\theta_t = \theta_{t-1} - \eta(\sum \bar{\boldsymbol{g}}^{(I_j)} + z)$, where $z \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.
    for $i = 1$ *to* n **do**
      Set $\boldsymbol{\rho}^{(i)} = \boldsymbol{\rho}_c$ where $c = C^{(i)}$.
      $\boldsymbol{o}^{(i)} = \boldsymbol{o}^{(i)} + \boldsymbol{\rho}^{(i)}$. *//Update individual privacy costs.*
    end
  end
end

3.1 Privacy Analysis of Algorithm 1

We state the privacy guarantee of Algorithm 1 in Theorem 3.1. We use output-specific DP to precisely account the privacy costs of a realized run of Algorithm 1.

Theorem 3.1. *[Output-specific privacy guarantee] Algorithm 1 at step t satisfies $(o_\alpha^{(i)} + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-output-specific individual DP for the i_{th} example at $\mathbb{A} = (\theta_1, \ldots, \theta_{t-1}, \Theta_t)$, where $o_\alpha^{(i)}$ is the accumulated RDP at order α and Θ_t is the domain of θ_t.*

To prove Theorem 3.1, we first define a t-step non-adaptive composition with $\theta_1, \ldots, \theta_{t-1}$. We then show RDP of the non-adaptive composition gives an output-specific privacy bound on the adaptive composition in Algorithm 1. We relegate the proof to Appendix A.

3.2 Estimations of Gradient Norms Are Accurate

Although the gradient norms used for privacy accounting are updated only occasionally, we show that the individual privacy parameters based on estimations are very close to the those based on exact

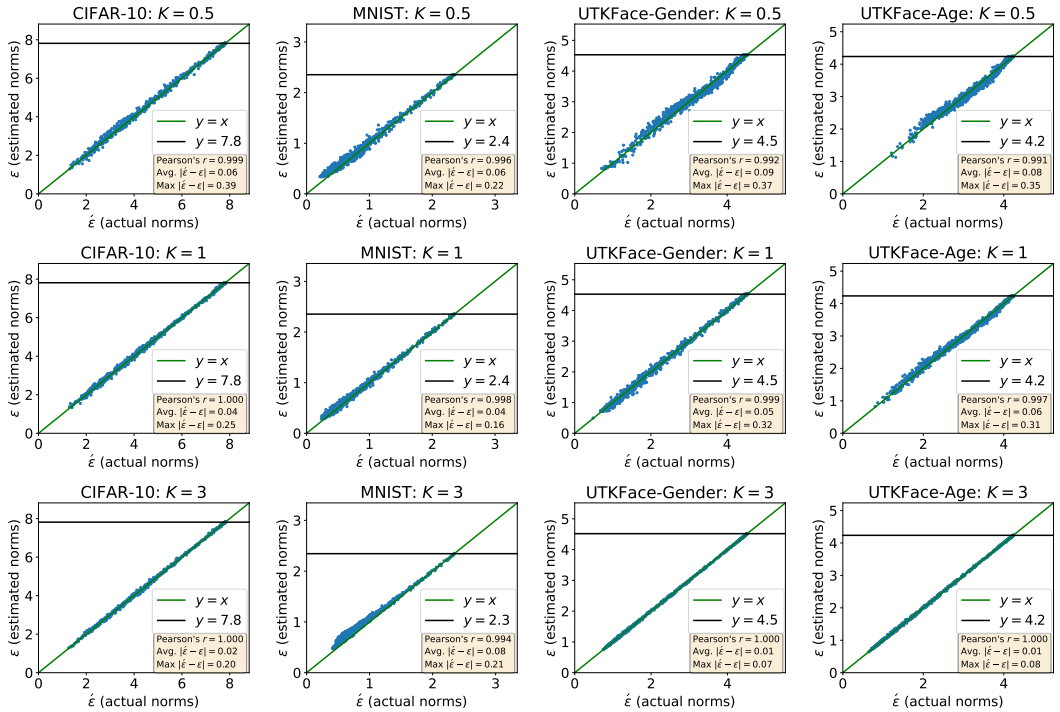

Figure 3: Privacy parameters based on estimations of gradient norms (ε) versus those based on exact norms ($\acute{\varepsilon}$). The results suggest that the estimations of gradient norms are close to the exact norms.

norms (Figure 3). This indicates that the estimations of gradient norms are close to the exact ones. We comment that the privacy parameters computed with exact norms are not exactly the same as those of vanilla DP-SGD because individual clipping in Algorithm 1 slightly modifies the training trajectory. We show the privacy parameters of Algorithm 1 are very close to those of the vanilla DP-SGD in Appendix C.2.

To compute the exact gradient norms, we randomly sample 1000 examples and compute the exact gradient norms at every iteration. We compute the Pearson correlation coefficient between the privacy parameters computed with estimated norms and those computed with exact norms. We also compute the average and the worst absolute errors. We report results on MNIST, CIFAR-10, and UTKFace. Details about the experiments are in Section 4. We plot the results in Figure 3. The ε values based on estimations are close to those based on exact norms (Pearson's $r > 0.99$) even we only update the gradient norms every two epochs ($K = 0.5$). Updating full batch gradient norms more frequently further improves the estimation, though doing so would increase the computational overhead.

It is worth noting that the maximum clipping threshold C affects the computed privacy parameters. Large C increases the variation of gradient norms (and hence the variation of privacy parameters) but leads to large noise variance while small C suppresses the variation and leads to large gradient bias. Large noise variance and gradient bias are both harmful for learning (Chen et al., 2020; Song et al., 2021). In Appendix D, we show the influence of using different C on both accuracy and privacy.

3.3 Rounding Individual Gradient Norms

The rounding operation in Algorithm 1 is essential to make the computation of individual privacy parameters feasible. The privacy cost of one example at each step is the Rényi divergences between Equation (1) and (2). For a fixed sampling probability and noise variance, we use $\boldsymbol{\rho}_c$ to denote the privacy cost of an example with gradient norm c. Note that $\boldsymbol{\rho}_c$ is different for every value of c. Consequently, there are at most $n \times E$ different privacy costs because individual gradient norms vary across different examples and epochs (n is the number of datapoints and E is the number of training epochs). In order to make the number of different $\boldsymbol{\rho}_c$ tractable, we round individual gradient norms with a prespecified precision r (see Line 12 in Algorithm 1). Because the maximum clipping

threshold C is usually a small constant, then, by the pigeonhole principle, there are at most $\lceil C/r \rceil$ different values of $\boldsymbol{\rho}_c$. Throughout this paper we set $r = 0.01C$ that has almost no impact on the precision of privacy accounting.

Table 1: Computational costs of computing individual privacy parameters for CIFAR-10 with $K = 1$.

	With Rounding	Without Rounding
# of numerical computations	1×10^2	1×10^7
Time (in seconds)	< 3	$\sim 2.6 \times 10^4$

We give a concrete comparison on the computational costs in Table 1. We run the numerical method in Mironov et al. (2019) once for every different privacy cost (with the default setup in the Opacus library (Yousefpour et al., 2021)). We run DP-SGD on CIFAR-10 for 200 epochs with $K = 1$. All results in Table 1 use multiprocessing with 5 cores of an AMD EPYC™ 7V13 CPU. With rounding, the overhead of computing individual privacy parameters is negligible. In contrast, the computational cost without rounding is more than 7 hours.

3.4 What Can We Do with Individual Privacy Parameters?

Note that individual 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 individual privacy parameters.

The first approach is to release ε_i to the owner of the ith 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 ith 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 computation only uses $(\theta_1, \ldots, \theta_{t-1})$, which is reported in a privacy-preserving manner. We proof the claim in Appendix E.1.

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 privacy cost Andrew et al. (2021). We show the statistics can be released accurately with a very small privacy parameter ($\varepsilon = 0.1$) in Appendix E.2.

Finally, individual privacy parameters can also serve as a powerful tool for a trusted data curator to improve the model quality. By analysing the individual privacy parameters of a dataset, a trusted curator can focus on collecting more data representative of the groups that have higher privacy risks to mitigate the disparity in privacy.

4 Individual Privacy Parameters on Different Datasets

We first show the distribution of individual privacy parameters of running DP-SGD on four classification tasks in Section 4.1. Then we investigate how individual privacy parameters correlate with training loss in Section 4.2. Experimental setup is as follows.

Datasets. We use two benchmark datasets MNIST ($n = 60000$) and CIFAR-10 ($n = 50000$) (LeCun et al., 1998; Krizhevsky, 2009) as well as the UTKFace dataset ($n \simeq 15000$) (Zhang et al., 2017) 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.[1] 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. For CIFAR-10, we use the WRN16-4 model in De et al. (2022), which achieves advanced performance in private setting. We follow the implementation details in De et al. (2022) expect their data augmentation method to reduce computational cost. For MNIST and UTKFace, we use ResNet20 models with batch normalization layers replaced by group normalization layers. For UTKFace, we initialize the model with weights pre-trained on ImageNet. We set $C = 1$ on CIFAR-10, following De et al. (2022). For MNIST and UTKFace, we set C as the median of

[1]We 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 groups, we revisit this domain through a related lens. The labels are provided by the dataset curators.

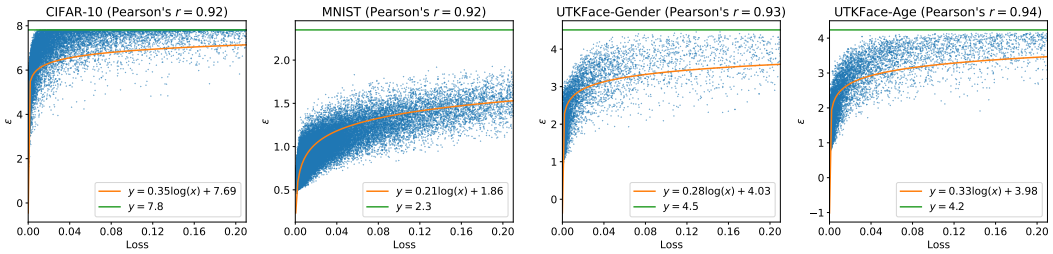

Figure 4: Privacy parameters and final training losses. Each point shows the loss and privacy parameter of one example. Pearson's r is computed between privacy parameters and log losses.

gradient norms at initialization, following the suggestion in Abadi et al. (2016). We set $K = 3$ for all experiments in this section. More details about the hyperparameters are in Appendix B.

4.1 Individual Privacy Parameters Vary Significantly

Figure 1 shows the individual privacy parameters on all datasets. The privacy parameters vary across a large range on all tasks. On the CIFAR-10 dataset, the maximum ε_i value is 7.8 while the minimum ε_i value is only 1.1. We also observe that, for easier tasks, more examples enjoy stronger privacy guarantees. For example, $\sim$40% 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.

4.2 Privacy Parameters And Loss Are Positively Correlated

We study how individual privacy parameters correlate with individual training loss. The analysis in Section 2 suggests that the privacy parameter of one example depends on its gradient norms across training. Intuitively, an example would have high loss after training if its gradient norms are large. We visualize individual privacy parameters and the final training loss values in Figure 4. The individual privacy parameters on all datasets increase with loss until they reach the maximum ε. To quantify the order of correlation, we further fit the points with one-dimensional logarithmic functions and compute the Pearson correlation coefficients with the privacy parameters predicted by the fitted curves. The Pearson correlation coefficients are larger than 0.9 on all datasets, showing an logarithmic correlation between the privacy parameter of a datapoint and its training loss.

5 Groups Are Simultaneously Underserved in Both Accuracy and Privacy

5.1 Low-Accuracy Groups Have Worse Privacy Parameters

It is well-documented that machine learning models may have large differences in accuracy on different groups (Buolamwini & Gebru, 2018; Bagdasaryan et al., 2019). Our finding demonstrates that this disparity may be simultaneous in terms of both accuracy *and* privacy. We empirically verify this by plotting the average ε and training/test accuracy of different groups. For CIFAR-10 and MNIST, the groups are the data from different classes, while for UTKFace, the groups are the data from different races.

We plot the results in Figure 5. The groups are sorted based on the average ε. Both training and test accuracy correlate well with ε. Groups with worse accuracy do have worse privacy guarantee in general. On CIFAR-10, the average ε of the 'Cat' class (which has the worst test accuracy) is 43.6% higher than the average ε of the 'Automobile' class (which has the highest test accuracy). On UTKFace-Gender, the average ε of the group with the lowest test accuracy ('Asian') is 25.0% higher than the average ε of the group with the highest accuracy ('Indian'). Similar observation also holds on other tasks. To the best of our knowledge, our work is the first to reveal this simultaneous disparity.

5.2 Low-Accuracy Groups Suffer From Higher Empirical Privacy Risks

We run membership inference (MI) attacks against models trained **without** differential privacy to study whether underserved groups are more vulnerable to empirical privacy attacks. Several recent works show that MI attacks have higher success rates on some examples than other examples (Song

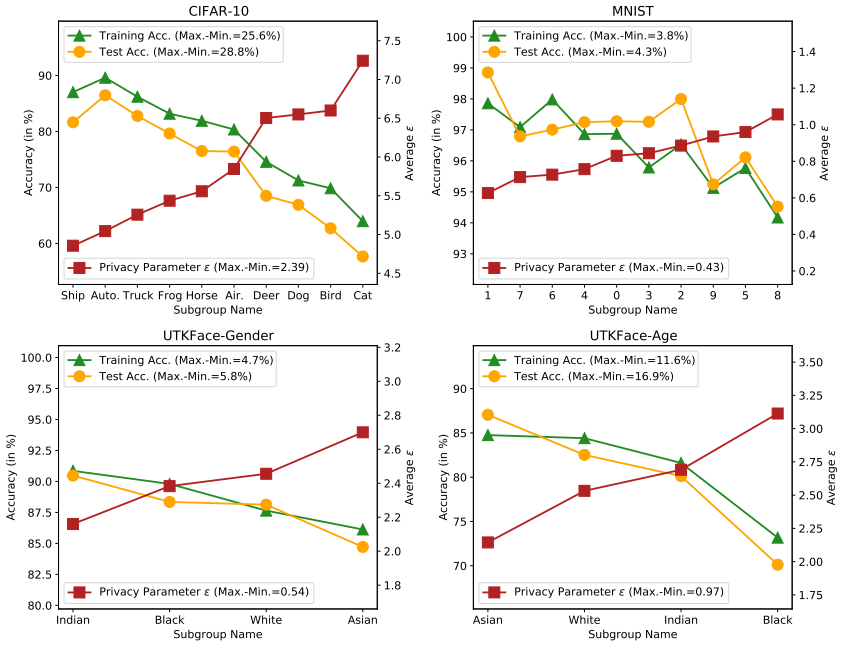

Figure 5: Accuracy and average ε of different groups. Groups with worse accuracy also have worse privacy in general.

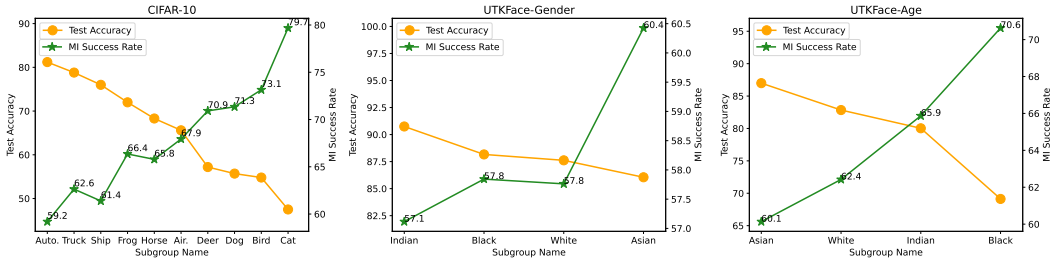

Figure 6: Success rates on different groups of MI attacks against models trained without DP.

& Mittal, 2020; Choquette-Choo et al., 2021; Carlini et al., 2021). In this section, we directly connect such disparity in privacy risks with the unfairness in utility. We use a simple loss-threshold attack that predicts an example is a member if its loss value is smaller than a prespecified threshold (Sablayrolles et al., 2019). For each group, we use its whole test set and a random subset of training set so the numbers of training and test losses 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 are in Figure 6. The groups are sorted based on their average ε. The disparity in privacy risks is clear. On UTKFace-Age, the MI success rate is 70.6% on the 'Black' group while is only 60.1% on the 'Asian' group. This suggests that empirical privacy risks also correlate well with the disparity in utility.

6 Conclusion

We propose an efficient algorithm to accurately estimate the individual privacy parameters for DP-SGD. We use this new algorithm to examine individual privacy guarantees on several datasets. Significantly, we find that groups with worse utility also suffer from worse privacy. This new finding reveals the complex while interesting relation among utility, privacy, and fairness. It has two immediate implications. Firstly, it shows that the learning objective aligns with privacy protection for a given privacy budget, i.e., the better the model learns about an example, the better privacy that example would get. Secondly, it suggests that mitigating the utility fairness under differential

privacy is more tricky than doing so in the non-private case. This is because classic methods such as upweighting underserved examples would exacerbate the disparity in privacy.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of ACM Conference on Computer and Communications Security*, 2016.

Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, 2021.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, 2019.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, 2018.

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*, 2014.

Daniel Bernau, Philip-William Grassal, Jonas Robl, and Florian Kerschbaum. Assessing differentially private deep learning with membership inference. *arXiv preprint arXiv:1912.11328*, 2019.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J. Su. Deep learning with Gaussian differential privacy. *Harvard Data Science Review*, 2020.

Zhiqi Bu, Jialin Mao, and Shiyun Xu. Scalable and efficient training of large convolutional neural networks with differential privacy. *arXiv preprint arXiv:2205.10683*, 2022.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability, and Transparency*, 2018.

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*, 2019.

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.

Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 2020.

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, pp. 1964–1974. JMLR, Inc., 2021.

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.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

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*, 2020.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a Renyi filter. In *Advances in Neural Information Processing Systems*, 2021.

Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems*, 2021.

Victor Petrén Bach Hansen, Atula Tejaswi Neerkaje, Ramit Sawhney, Lucie Flek, and Anders Søgaard. The impact of differential privacy on group disparity mitigation. *arXiv preprint arXiv:2203.02745*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In *Advances in Neural Information Processing Systems*, 2020.

Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *International Conference on Data Engineering (ICDE)*, 2015.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Mathias Lécuyer. Practical privacy filters and odometers with rényi differential privacy and applications to differentially private deep learning. *arXiv preprint arXiv:2103.01379*, 2021.

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*, 2022.

Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Z Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. *Advances in Neural Information Processing Systems*, 2017.

Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:*, 2022.

Ilya Mironov. Rényi differential privacy. In *Proceedings of the 30th IEEE Computer Security Foundations Symposium*, 2017.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

Christopher Mühl and Franziska Boenisch. Personalized PATE: Differential privacy for machine learning with individual privacy guarantees. *arXiv preprint arXiv:2202.10517*, 2022.

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.

Frederik Noe, Rasmus Herskind, and Anders Søgaard. Exploring the unfairness of dp-sgd across settings. *arXiv preprint arXiv:2202.12058*, 2022.

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*, 2017.

Rachel Redberg and Yu-Xiang Wang. Privately publishable per-instance privacy. In *Advances in Neural Information Processing Systems*, 2021.

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*, 2019.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. *arXiv preprint arXiv:2003.10595*, 2020.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of IEEE Global Conference on Signal and Information Processing*, 2013.

Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Vinith M Suriyakumar, Nicolas Papernot, Anna Goldenberg, and Marzyeh Ghassemi. Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of ACM Conference on Fairness, Accountability, and Transparency*, 2021.

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.

Yu-Xiang Wang. Per-instance differential privacy. *The Journal of Privacy and Confidentiality*, 2019.

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*, 2019.

Justin Whitehouse, Aaditya Ramdas, Ryan Rogers, and Zhiwei Steven Wu. Fully adaptive composition in differential privacy. *arXiv preprint arXiv:2203.05481*, 2022.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

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*, 2022.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017.

Yuqing Zhu and Yu-Xiang Wang. Poission subsampled Rényi differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

A PROOF OF THEOREM 3.1

Theorem 3.1. *[Output-specific privacy guarantee] Algorithm 1 at step t satisfies* $(o_\alpha^{(i)} + \frac{\log(1/\delta)}{\alpha-1}, \delta)$*-output-specific individual DP for the* i_{th} *example at* $\mathbb{A} = (\theta_1, \ldots, \theta_{t-1}, \Theta_t)$*, where* $o_\alpha^{(i)}$ *is the accumulated RDP at order* α *and* Θ_t *is the domain of* θ_t*.*

Here we give the the proof of Theorem 3.1. We compose the privacy costs at different steps through Rényi differential privacy (RDP) (Mironov, 2017). RDP uses the Rényi divergence at different orders to measure privacy. We use $D_\alpha(\mu||\nu) = \frac{1}{\alpha-1}\log\int(\frac{d\mu}{d\nu})^\alpha d\nu$ to denote the Rényi divergence at order α between μ and ν and $D_\alpha^\leftrightarrow(\mu||\nu) = \max(D_\alpha(\mu||\nu), D_\alpha(\nu||\mu))$ to denote the maximum divergence of two directions. The definition of individual RDP is as follows.

Definition 3. *[Individual RDP Feldman & Zrnic (2021)] Fix a datapoint* $\boldsymbol{d}$*, let* $\mathbb{D}$ *be an arbitrary dataset and* $\mathbb{D}' = \mathbb{D} \cup \{\boldsymbol{d}\}$*. A randomized algorithm* $\mathcal{A}$ *satisfies* $(\alpha, \rho(\boldsymbol{d}))$*-individual RDP for* $\boldsymbol{d}$ *if it holds that* $D_\alpha^\leftrightarrow(\mathcal{A}(\mathbb{D})||\mathcal{A}(\mathbb{D}')) \leq \rho(\boldsymbol{d})$*.*

Let $(\mathcal{A}_1, \ldots, \mathcal{A}_{t-1})$ be a sequence of randomized algorithms and $(\theta_1, \ldots, \theta_{t-1})$ be some arbitrary fixed outcomes from the domain, we define

$$\hat{\mathcal{A}}^{(t)}(\theta_1, \ldots, \theta_{t-1}, \mathbb{D}) = (\mathcal{A}_1(\mathbb{D}), \mathcal{A}_2(\theta_1, \mathbb{D}), \ldots, \mathcal{A}_t(\theta_1, \ldots, \theta_{t-1}, \mathbb{D})).$$

Noting that $\hat{\mathcal{A}}^{(t)}$ is not an adaptive composition as the input of each individual mechanism does not depend on the outputs of previous mechanisms. Further let

$$\mathcal{A}^{(t)}(\mathbb{D}) = (\mathcal{A}_1(\mathbb{D}), \mathcal{A}_2(\mathcal{A}_1(\mathbb{D}), \mathbb{D}), \ldots, \mathcal{A}_t(\mathcal{A}_1(\mathbb{D}), \ldots, \mathbb{D}))$$

be the adaptive composition. In Theorem A.1, we show a RDP bound on the non-adaptive composition $\hat{\mathcal{A}}^{(t)}$ gives an output-specific DP bound on the adaptive composition $\mathcal{A}^{(t)}$.

Theorem A.1. *Let* $\mathbb{A} = (\theta_1, \ldots, \theta_{t-1}, Range(\mathcal{A}_t)) \subset Range(\mathcal{A}^{(t)}) = Range(\hat{\mathcal{A}}^{(t)})$ *where* $\theta_1, \ldots, \theta_{t-1}$ *are some arbitrary fixed outcomes. If* $\hat{\mathcal{A}}^{(t)}(\cdot)$ *satisfies* o_α *RDP at order* α*, then* $\mathcal{A}^{(t)}(\mathbb{D})$ *satisfies* $(o_\alpha + \frac{\log(1/\delta)}{\alpha-1}, \delta)$*-output-specific differential privacy at* $\mathbb{A}$*.*

Proof. For a given outcome $\theta^{(t)} = (\theta_1, \theta_2, \ldots, \theta_{t-1}, \theta_t) \in \mathbb{A}$, we have $\mathbb{P}\left[\mathcal{A}^{(t)}(\mathbb{D}) = \theta^{(t)}\right] =$

$$\mathbb{P}\left[\mathcal{A}^{(t-1)}(\mathbb{D}) = \theta^{(t-1)}\right]\mathbb{P}\left[\mathcal{A}_t(\mathcal{A}_1(\mathbb{D}), \ldots, \mathcal{A}_{t-1}(\mathbb{D}), \mathbb{D}) = \theta_t | \mathcal{A}^{(t-1)}(\mathbb{D}) = \theta^{(t-1)}\right], \tag{3}$$

$$= \mathbb{P}\left[\mathcal{A}^{(t-1)}(\mathbb{D}) = \theta^{(t-1)}\right]\mathbb{P}\left[\mathcal{A}_t(\theta_1, \ldots, \theta_{t-1}, \mathbb{D}) = \theta_t\right], \tag{4}$$

by the product rule of conditional probability. Apply the product rule recurrently on $\mathbb{P}\left[\mathcal{A}^{(t-1)}(\mathbb{D}) = \theta^{(t-1)}\right]$, we have $\mathbb{P}\left[\mathcal{A}^{(t)}(\mathbb{D}) = \theta^{(t)}\right] =$

$$\mathbb{P}\left[\mathcal{A}^{(t-2)}(\mathbb{D}) = \theta^{(t-2)}\right]\mathbb{P}\left[\mathcal{A}_{t-1}(\theta_1, \ldots, \theta_{t-2}, \mathbb{D}) = \theta_{t-1}\right]\mathbb{P}\left[\mathcal{A}_t(\theta_1, \ldots, \theta_{t-1}, \mathbb{D}) = \theta_t\right], \tag{5}$$

$$= \mathbb{P}\left[\mathcal{A}_1(\mathbb{D}) = \theta_1\right]\mathbb{P}\left[\mathcal{A}_2(\theta_1, \mathbb{D}) = \theta_2\right] \ldots \mathbb{P}\left[\mathcal{A}_t(\theta_1, \ldots, \theta_{t-1}, \mathbb{D}) = \theta_t\right], \tag{6}$$

$$= \mathbb{P}\left[\hat{\mathcal{A}}^{(t)}(\theta_1, \ldots, \theta_{t-1}, \mathbb{D}) = \theta^{(t)}\right]. \tag{7}$$

In words, $\mathcal{A}^{(t)}$ and $\hat{\mathcal{A}}^{(t)}$ are identical in $\mathbb{A}$. Therefore, $\mathcal{A}^{(t)}$ satisfies (ε, δ)-DP at any $\mathbb{S} \subset \mathbb{A}$ if $\hat{\mathcal{A}}^{(t)}$ satisfies (ε, δ)-DP. Converting the RDP bound on $\hat{\mathcal{A}}^{(t)}(\mathbb{D})$ into a (ε, δ)-DP bound with Lemma A.2 then completes the proof.

Lemma A.2 (Conversion from RDP to (ε, δ)-DP Mironov (2017))**.** *If* $\mathcal{A}$ *satisfies* (α, ρ)*-RDP, then* $\mathcal{A}$ *satisfies* $(\rho + \frac{\log(1/\delta)}{\alpha-1}, \delta)$*-DP for all* $0 < \delta < 1$*.*

□

We comment that the RDP parameters of the adaptive composition of Algorithm 1 are random variables because they depend on the randomness of the intermediate outcomes. This is different from the conventional analysis that chooses a constant privacy parameter before training. Composition of random RDP parameters requires additional care because one needs to upper bound the privacy parameter over its randomness (Feldman & Zrnic, 2021; Lécuyer, 2021; Whitehouse et al., 2022). In this paper, we focus on the realizations of those random RDP parameters and hence provide a precise output-specific privacy bound.

B More Details on Hyperparameters

The batchsize is 4K, 2K, and 200 for CIFAR-10, MNIST, and UTKFace, respectively. The training epoch is 300 for CIFAR-10 and 100 for MNIST and UTKFace. We use the package in Yousefpour et al. (2021) to compute the noise multiplier. The standard deviation of noise in Algorithm 1 is the noise multiplier times the maximum clipping threshold. All experiments are run on a single Tesla V100 GPU with 32G memory. Our source code will be released soon.

C More Discussion on Individual Clipping

C.1 Individual Clipping Does Not Affect Accuracy

Here we run experiments to check the influence of using individual clipping thresholds on utility. Algorithm 1 uses individual clipping thresholds to ensure the computed privacy parameters are strict 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, individual thresholds would clip more signal than using a single maximum threshold.

Table 2: Comparison between the test accuracy of using individual clipping thresholds and that of using a single maximum clipping threshold. The maximum ε is 7.8 for CIFAR-10 and 2.4 for MNIST.

	CIFAR-10	MNIST
Individual	74.0 ($\pm$0.19)	97.17 ($\pm$0.12)
Maximum	74.1 ($\pm$0.24)	97.26 ($\pm$0.11)

We compare the accuracy of two different clipping methods in Table 2. The individual clipping thresholds are updated once per epoch. 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 individual clipping thresholds in Algorithm 1 has a negligible effect on accuracy.

C.2 Experiments Without Individual Clipping

We run Algorithm 1 without individual clipping, i.e., vanilla DP-SGD in Abadi et al. (2016), to see whether our conclusions in the main text still hold. Specifically, we change the Line 15 of Algorithm 1 to clip all gradients with the maximum clipping threshold C. Other experimental setup is the same as that in Experiment 4 and Appendix B.

We get accurate estimations of actual guarantees. The privacy parameters are still computed with the estimations of gradient norms and hence are estimations of the actual guarantees. We compare the privacy parameters and actual guarantees in Figure 7. To compute the actual guarantees, we randomly sample 1000 examples and compute their exact gradient norms at every iteration. The results suggest that the privacy parameters are accurate estimations (Pearson's $r > 0.99$ and small maximum absolute errors).

Privacy parameters still have a strong correlation with training loss. In Figure 8, we show privacy parameters computed without individual clipping are still positively correlated with training losses. The Pearson correlation coefficient between privacy parameters and log losses is larger than 0.9 for all datasets. Moreover, we show our observation in Section 5, i.e., low-accuracy groups have worse privacy parameters, still holds in Figure 9. We also make a direct comparison with privacy parameters computed with individual clipping. We find that privacy parameters computed with individual clipping are very close to those computed without individual clipping. We also find that the order of groups, sorted by average ε, is exactly the same for both cases.

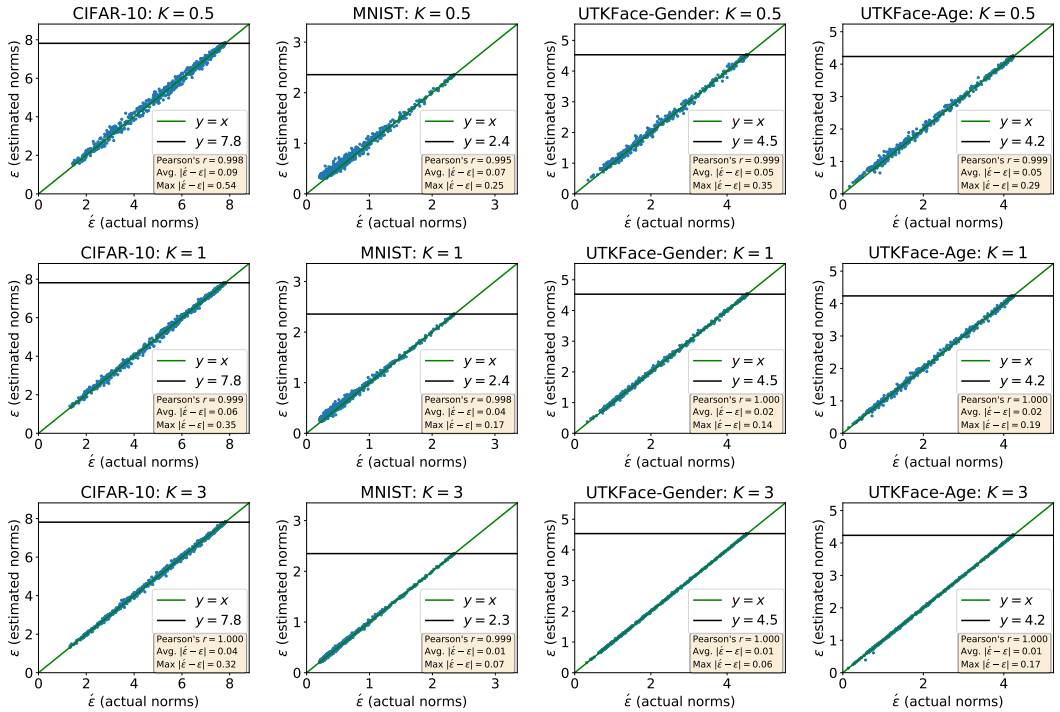

Figure 7: Privacy parameters based on estimations of gradient norms (ε) versus the actual privacy guarantees ($\acute{\varepsilon}$). We do not use individual clipping in this plot. The privacy parameters are very close to the actual guarantees.

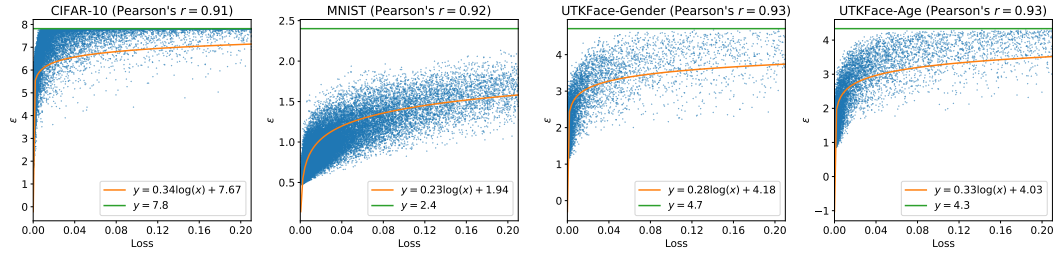

Figure 8: Privacy parameters computed without individual clipping and final training losses. The Pearson correlation coefficient is computed between privacy parameters and log losses.

D The Influence of Maximum Clipping on Individual Privacy

The value of the maximum clipping threshold C would affect individual privacy parameters. A large value of C would increase the stratification in gradient norms but also increase the noise variance for a fixed privacy budget. A small value of C would suppress the stratification but also increase the gradient bias. Here we run experiments with different values of C on CIFAR-10. We use the standard ResNet20 for CIFAR-10 in He et al. (2016), which only has $\sim$0.2M parameters, to reduce the computation cost. Let M be the median of gradient norms at initialization, we choose C from the list $[0.1M, 0.3M, 0.5M, 1.5M]$. Other experimental setup is the same as that in Section 4.

The histograms of individual privacy parameters are in Figure 10. In terms of accuracy, using clipping thresholds near the median gives better test accuracy. In terms of privacy, using smaller clipping thresholds increases privacy parameters in general. The number of datapoints that reaches the worst privacy decreases with the value of C. When $C = 0.1M$, nearly 70% datapoints reach the worst privacy parameter while only $\sim$2% datapoints reach the worst parameter when $C = 1.5M$.

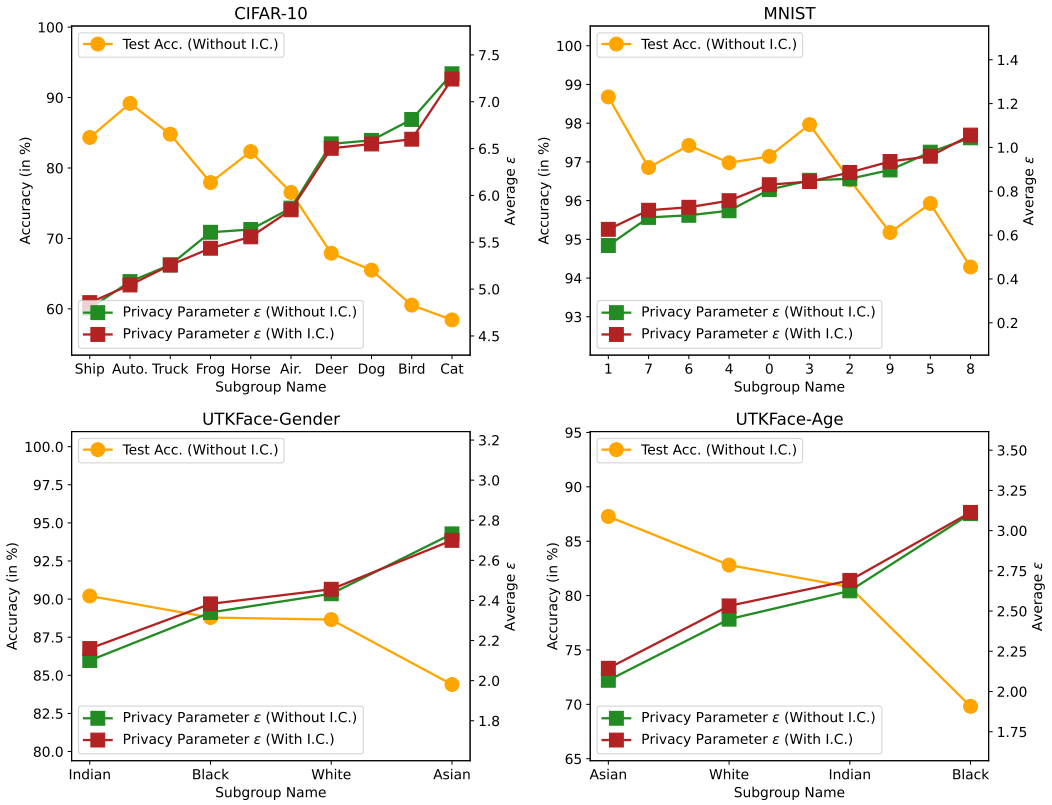

Figure 9: Test accuracy and privacy parameters computed with/without individual clipping. Groups with worse test accuracy also have worse privacy in general.

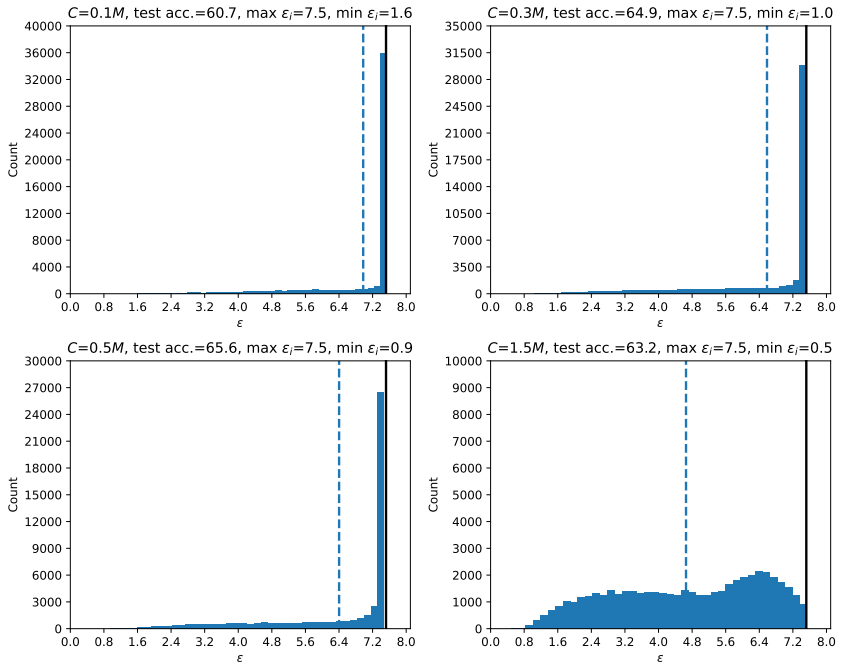

Figure 10: Distributions of individual privacy parameters on CIFAR-10 with different maximum clipping thresholds. The dashed line indicates the average of privacy parameters.

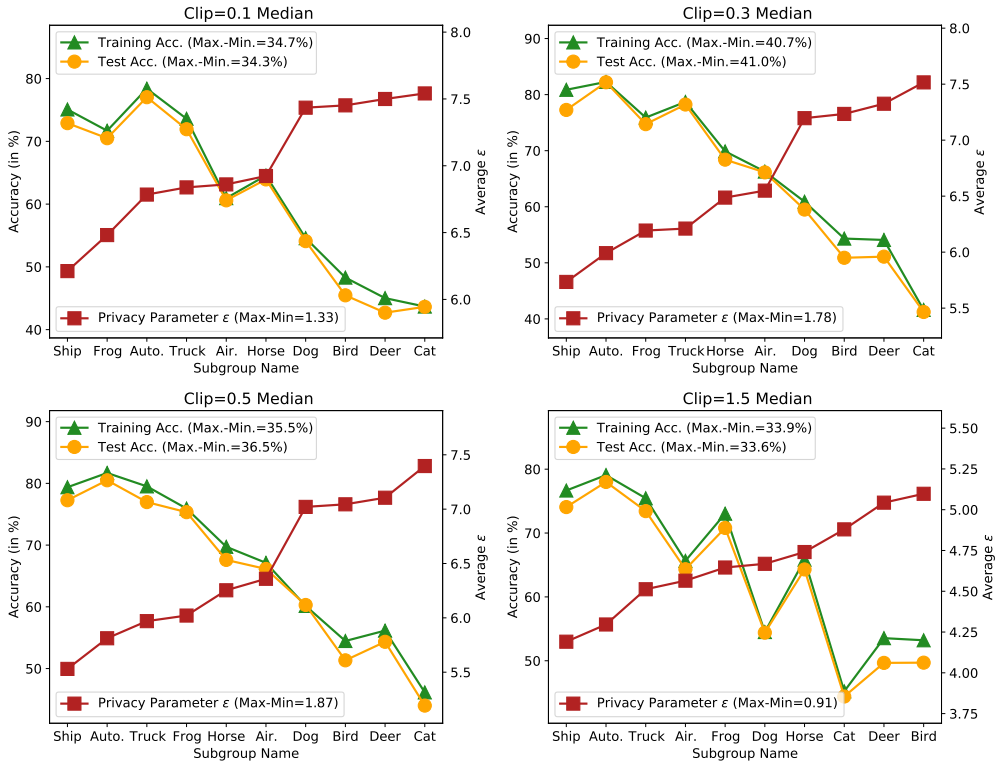

Figure 11: Accuracy and average ε of different groups on CIFAR-10 with different maximum clipping thresholds.

The correlation between accuracy and privacy is in Figure 11. The disparity in average ε is clear for all choices of C. Another important observation is that when decreasing C, the privacy parameters of underserved groups increase quicker than other groups. When changing $C = 1.5M$ to $0.5M$, the average ε of 'Cat' increases from 4.8 to 7.4, almost reaching the worst-case bound. In comparison, the increment in ε of the 'Ship' class is only 1.3 (from 4.2 to 5.5).

E Make Use of Individual Privacy Parameters

E.1 Releasing Individual Privacy Parameters to Rightful Owners

Let ε_i be the privacy parameter of the i_{th} user, we can compute ε_i with the training trajectory and $\boldsymbol{d}_i$ itself. Theorem E.1 shows that releasing ε_i does not increase the privacy cost of any other example $\boldsymbol{d}_j \neq \boldsymbol{d}_i$. The proof uses the fact that computing ε_i can be seen as a post-processing of $(\theta_1, \ldots, \theta_{t-1})$, which is reported in a privacy-preserving manner.

Theorem E.1. *Let $\mathcal{A} : \mathcal{D} \to \mathcal{O}$ be an algorithm that is (ε_j, δ)-output-specific DP for $\boldsymbol{d}_j$ at $\mathbb{A} \subset \mathcal{O}$. Let $f(\cdot, \boldsymbol{d}_i) : \mathcal{O} \to \mathcal{R} \times \mathcal{O}$ be a post-processing function that returns the privacy parameter of $\boldsymbol{d}_i$ ($\neq \boldsymbol{d}_j$) and the training trajectory. We have f is (ε_j, δ)-output-specific DP for $\boldsymbol{d}_j$ at $\mathbb{F} \subset \mathcal{R} \times \mathcal{O}$ where $\mathbb{F} = \{f(a, \boldsymbol{d}_i) : a \in \mathbb{A}\}$ is all possible post-processing results.*

Proof. We first note that the construction of f is independent of $\boldsymbol{d}_j$. Without loss of generality, let $\mathbb{D}, \mathbb{D}' \in \mathcal{D}$ be the neighboring datasets where $\mathbb{D}' = \mathbb{D} \cup \{\boldsymbol{d}_j\}$. Let $\mathbb{S} \subset \mathbb{F}$ be an arbitrary event and $\mathbb{T} = \{a \in \mathbb{A} : f(a, \boldsymbol{d}_i) \in \mathbb{S}\}$. Because f is a bijective function, we have

$$\Pr[f(\mathcal{A}(D), \boldsymbol{d}_i) \in \mathbb{S}] = \Pr[\mathcal{A}(D) \in \mathbb{T}] \tag{8}$$

$$\leq e^{\varepsilon_j} \Pr[\mathcal{A}(D') \in \mathbb{T}] + \delta \tag{9}$$

$$= e^{\varepsilon_j} \Pr[f(\mathcal{A}(D'), \boldsymbol{d}_i) \in \mathbb{S}] + \delta, \tag{10}$$

which completes the proof.

Table 3: Populational statistics of individual privacy parameters on MNIST. The average estimation error rate is 1.19%. 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	0.850	0.362	0.467	0.626	0.931	1.840
$\varepsilon = 0.1$	0.852	0.361	0.469	0.628	0.935	1.742

Table 4: Populational statistics of individual privacy parameters on CIFAR-10. The average estimation error rate is 1.51%. 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	5.813	2.870	4.957	6.602	7.290	7.488
$\varepsilon = 0.1$	5.811	2.960	4.975	6.609	7.357	7.828

□

E.2 Release Populational Statistics of Privacy Parameters

The individual 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 individual parameters can be released with minor privacy cost. Specifically, we compute the average and quantiles of the ε values with differential privacy. For average, we release the noisy aggregation through Gaussian Mechanism. For quantiles, we solve the objective function in Andrew et al. (2021) with 20 steps of full batch gradient descent. The results on MNIST and CIFAR-10 are in Table 3 and Table 4 respectively. The released statistics are close to the actual values on both datasets with $(0.1, 10^{-5})$-DP.