Have it your way: Individualized Privacy Assignment for DP-SGD

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

When training a machine learning model with differential privacy, one sets a privacy budget. This uniform budget represents an overall maximal privacy violation that any user is willing to face by contributing their data to the training set. We argue that this approach is limited because different users may have different privacy expectations. Thus, setting a uniform privacy budget across all points may be overly conservative for some users or, conversely, not sufficiently protective for others. In this paper, we capture these preferences through individualized privacy budgets. To demonstrate their practicality, we introduce a variant of Differentially Private Stochastic Gradient Descent (DP-SGD) which supports such individualized budgets. DP-SGD is the canonical approach to training models with differential privacy. We modify its data sampling and gradient noising mechanisms to arrive at our approach, which we call Individualized DP-SGD (IDP-SGD). Because IDP-SGD provides privacy guarantees tailored to the preferences of individual users and their data points, we empirically find it to improve privacy-utility trade-offs.

1 Introduction

Machine learning (ML) models are known to leak information about their training data. Such leakage can result in attacks that determine whether a specific data point was used to train a given ML model (membership inference) [29, 33, 6], infer sensitive attributes from the model's training data [11, 35], or even (partially) reconstruct that training data [12, 28]. Therefore, the need to provide privacy guarantees to the individuals who contribute their sensitive data to train ML models is pressing.

Approaches to train ML models with guarantees of differential privacy [8] have established themselves as the canonical answer to this need. In the context of ML, differential privacy bounds how much can be learned about any of the individual data point from the model's training set. Take the canonical example of differentially private stochastic gradient descent (DP-SGD): it updates the model using noisy averages of clipped per-example gradients. Unlike vanilla SGD, this ensures that any single training point has a limited influence on model updates (due to the clipping to a pre-defined maximum clip norm) and that the training is likely to output a similar model should a single point be added or removed to the training set (due to the noisy average). As is the case for any differentially private algorithm, the privacy guarantees of DP-SGD are derived analytically. They are captured by a privacy budget ε. This budget represents the maximal privacy violation that any user contributing data to the training set is willing to tolerate. Lower budgets correspond to stronger privacy protection since they impose that the outputs of the training algorithm have to be closer for similar training sets.

In this paper, we argue that there is a key limitation when training ML models with DP guarantees: the privacy budget ε is set uniformly across all training points. This implicitly assumes that all users who contribute their data to the training set have the same expectations of privacy. However, that is not true; individuals have diverse values and privacy preferences [13, 4]. Yet, because DP-SGD assumes a uniform privacy budget, this budget must match the individual whose privacy expectations are the

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

strongest. That is, ε has to correspond to the lowest privacy violation tolerance expressed amongst individuals who contributed their data. This limits the ability of DP-SGD to learn because the privacy budget ε together with the sensitivity of each step of the algorithm (*i.e.*, the clip norm) define the scale of noise that needs to be added: the stronger the required privacy protection, the higher the noise scale. Thus, implementing privacy protection according to the lowest privacy violation tolerance in the training dataset comes at a significant cost in model accuracy.

We are the first to propose two variants of the DP-SGD algorithm to enforce different privacy budgets for each training point.[1] First, our modified sampling mechanism (Sample) samples data points proportionally to their privacy budgets: points with higher privacy budgets are sampled more frequently during training. It is intuitive that the more a training point is analyzed during training, the more private information can leak about this point. Second, with our individualized scaling mechanism (Scale), we rescale the noise added to gradients based on the budget specified for each training point. Naively, one would implement Scale by changing the scale of added noise per individual data point. However, for efficiency, current implementations of DP-SGD typically noise the *sum* of the per-example clipped gradients over an entire mini-batch, hence, adding the same amount of noise to all gradients. Yet, within the context of this implementation, we note that we can *effectively* adapt the scale of noise being added on an individual basis by adjusting the sensitivity (*i.e.* the clip norm) of each example by a multiplier. Because clipping is performed on a per-example basis, this approach enables the efficient implementation of our proposed Scale method.

To summarize our contributions:

- We introduce two novel individualized variants of the DP-SGD algorithm, where each data point can be assigned its desired privacy budget. This allows individuals who contribute data to the training set to specify different privacy expectations. We refer to our individualized DP-SGD algorithm as *IDP-SGD*.

- We provide a theoretical analysis of our two mechanisms to derive how the privacy parameters (*e.g.*, noise multipliers and sample rates) should be set to meet individual data points' privacy guarantees and provide a thorough privacy analysis for Sample and Scale.

- We carry out an extensive empirical evaluation on MNIST, SVHN, and CIFAR10. Our results are summarized in Table 7 for the reader's convenience. They highlight the utility gain of our methods for different distributions of privacy expectations among users.

Ethical considerations. We need to ensure that our individualized privacy assignment does not harm individuals by disclosing too much of their sensitive information. We, therefore, suggest implementing our methods in a controlled environment where privacy risks are openly communicated [32, 7, 10], and where a regulatory entity is in charge of ensuring that even the individuals with lowest expectations regarding their privacy obtain a sufficient degree of protection. We further discuss the ethical deployment of our Individualized DP-SGD (IDP-SGD) in Appendix A.

2 Background and Related Work

2.1 Differential Privacy and DP-SGD

Algorithm M satisfies (ε, δ)**-DP**, if for any two datasets $D, D' \subseteq \mathcal{D}$ that differ in any one record and any set of outputs R

$$\mathbb{P}\left[M(D) \in R\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta. \tag{1}$$

Since the possible difference in outputs is bounded for any pair of datasets, DP bounds privacy leakage for any individual. The value $\varepsilon \in \mathbb{R}_+$ specifies the privacy level with lower values corresponding to stronger privacy guarantees. The $\delta \in [0, 1]$ offers a relaxation, *i.e.*, a small probability of violating the guarantees. See Appendix B.1 for a more thorough introduction of (ε, δ)-DP.

Another relaxation of DP is based on the Rényi divergence [20], namely (α, ε)**-RDP**, which for the parameters as described above and an order $\alpha \in (1, \infty)$ is defined as follows: $\mathbb{D}_{\alpha}\left(M(D) \parallel M(D')\right) \leq \varepsilon$. Due to its smoother composition properties, it is employed for privacy accounting in ML with

---

[1]In the rest of this paper, we assume without loss of generality that each training point is contributed by a different individual.

2

DP-SGD. Yet, it is possible to convert between the two notions: If M is an (α, ε)-RDP mechanism, it also satisfies $(\varepsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP for any $0 < \delta < 1$ [20].

DP-SGD extends standard SGD with two additional steps, namely clipping and noise addition. Clipping each data point's gradients to a pre-defined *clip norm* c bounds their *sensitivity* to ensure that no data point causes too large model updates. Adding noise from a Gaussian distribution, $\mathcal{N}(0, (\sigma c)^2 \mathbf{I})$, with zero mean and standard deviation according to the sensitivity c and a pre-defined *noise multiplier* σ introduces privacy. We formally depict DP-SGD as Algorithm 3 in Appendix B.1. To yield tighter privacy bounds, DP-SGD implements privacy amplification through subsampling [3] (line 3 in Algorithm 3) where training data points are sampled into mini-batches with a Poisson sampling,[2] rather than assigning each data point from the training dataset to a mini-batch before an epoch starts. DP-SGD hence implements a Sampled Gaussian Mechanism (SGM) [21] where each training data point is sampled independently at random without replacement with probability $q \in (0, 1]$ over I training iterations. As shown by Mironov *et al.* [21], an SGM with sensitivity of $c = 1$, satisfies (α, ε)-RDP where

$$\varepsilon \leq I \cdot 2q^2 \frac{\alpha}{\sigma^2}. \tag{2}$$

Equation (2) highlights that the privacy guarantee ε depends on the noise multiplier σ, the sample rate q, the number of training iterations I, and the RDP order α.

2.2 Individualized Privacy

Individualized privacy is a topic of high importance given that society consists at least of three different groups of individuals, demanding either strong, average, or weak privacy protection for their data, respectively [13, 4]. Furthermore, some individuals inherently require higher privacy protection, given their personal circumstances and the degree of uniqueness of their data. However, without individualization, when applying DP to datasets that hold data from individuals with different privacy requirements, the privacy level ε has to be chosen according to the lowest ε encountered among all individuals, which favors poor privacy-utility trade-offs. Prior work on individualized privacy guarantees focused mainly on standard data analyses without considering ML [2, 15, 19, 24]. However, some of the underlying ideas inspired the design of our IDP-SGD variants.

Our **Sample** method relies on a similar idea as the *sample mechanism* proposed by Jorgensen et al. [15] where data points with stronger privacy requirements are sampled with a lower probability than data points that agree to contribute more of their information. The most significant difference between the *sample mechanism* and our approach is that the former performs sampling as a pre-processing step independent of the subsequent algorithm. In contrast, we perform subsampling *within* every iteration of our IDP-SGD training and show how to leverage the sampling probability of DP-SGD to obtain individual privacy guarantees. Based on this difference, we could not build on how Jorgensen et al. [15] compute the sampling probabilities and had to propose an original mechanism to derive sampling probabilities for individualized privacy in IDP-SGD.

In a similar vein to our **Scale** method, Alaggan et al. [2] scale up data points individually before the noise addition to increase their sensitivity in their *stretching mechanism*. As a consequence, the signal-to-noise ratios of data points that are scaled with large factors will be higher than the ones of data points that are scaled with small factors. In contrast to Alaggan et al. [2], we do not scale the data points themselves, but the magnitude of noise added to them relative to their individual sensitivity. Our new interesting observation is that data points' individual clip norms can be used to indirectly scale the noise added to a mini-batch of data in DP-SGD per data point.

Closest work to ours [5] proposes two mechanisms for an individualized privacy assignment within the framework of Private Aggregation of Teacher Ensembles (PATE) [26]. Their *upsampling* duplicates training data points and assigns the duplicates to different teacher models according to the data points' privacy requirements, while their *weighting* changes the aggregation of the teachers' predicted labels to reduce or increase the contribution of teachers according to their respective training data points' privacy requirements. PATE's approach for implementing ML with DP differs significantly from DP-SGD: PATE relies on privately training multiple non-DP models on the sensitive data and introducing

---

[2]In DP-SGD, the concept of an epoch does not exist and training duration is measured in *iterations*. Due to the Poisson sampling, training mini-batches can contain varying numbers of data points.

3

DP guarantees during a knowledge transfer to a separately released model. The approaches for individualized privacy in PATE, therefore, are non-applicable to DP-SGD. See Appendix B.4 for details on the PATE algorithm and its individualized extensions.

Relationship to Individualized Privacy Accounting. An orthogonal line of research that comes closest to providing individualized guarantees for DP-SGD focuses on individualized *privacy accounting*; the idea is to learn more from points that consume less of the privacy budget over training [9, 34]. Jordon et al. [14] propose a personalized moments' accountant to compute the privacy loss on a per-sample basis. Yu et al. [34] perform individualized privacy accounting within DP-SGD based on the gradient norms of the individual data points. Feldman and Zrnic [9] introduce *RDP filters* to account for privacy consumption per data point and train as long as there remain data points that do not exceed the global privacy level. Yet, it has been shown that the data points that consume little privacy budget and therefore remain in the training are the ones that already incur a small training loss [34]. Training more on them will not significantly boost the model accuracy. While privacy assignment and accounting are concerned with improving privacy-utility trade-offs within DP, they operate under different setups. Privacy *accounting* (in the above three methods) assumes a single fixed privacy budget assigned over the whole dataset. In contrast, our privacy *assignment* is concerned with enabling different individuals to specify their respective privacy preferences. Since individual accounting and assignment are two independent methods, we experimentally show their synergy by applying individual assignment to individual accounting [9], which yields higher model utility than individual accounting alone.

3 Our Individualized Privacy Assignment

3.1 Formalizing Individualizing Privacy

Setup and Notation. Given a training dataset D with points $\{x_1, \ldots, x_N\}$ which each have their own privacy preference (or *budget*), we consider data points with the same privacy budget ε_p together as a *privacy group* $\mathcal{G}_p$. This notation is aligned with [13, 4] that identified different *groups* of individual privacy preferences within society. We denote with ε_1 the smallest privacy budget encountered in the dataset. Standard DP-SGD needs to set $\varepsilon = \varepsilon_1$ to comply with the strongest privacy requirement encountered within the data. For all $\varepsilon_p, p \in [2, P]$, it follows that $\varepsilon_p > \varepsilon$ and we arrange the groups such that $\varepsilon_p > \varepsilon_{p-1}$. Note that following [2], we argue that the privacy preferences themselves should be kept private to prevent leakage of sensitive information which might be correlated with them. If there is a necessity to release individual privacy preferences, this should be done under the addition of noise to obtain DP guarantees for the privacy preferences. This can for instance be done through a smooth sensitivity analysis, as done in prior work, *e.g.* [27].[3]

Individualized Privacy. Following [15], our methods aim at providing data points $x_i \in \mathcal{G}_p$ with individual DP guarantees according to their privacy budget ε_p as follows: The learning algorithm M satisfies (ε_p, δ)-DP if for all datasets $D \overset{x_i}{\sim} D'$ which differ only in x_i and for all outputs $R \subseteq \mathcal{R}$

$$\mathbb{P}\left[M(D) \in R\right] \leq e^{\varepsilon_p} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta. \tag{3}$$

Without loss of generality, we assume that all individual data points have privacy preferences with the same δ. Note that other notions of DP (*e.g.*, RDP) can analogously be generalized to enable per-data point privacy guarantees. The key difference between Equation (3) and the standard definition of DP (Equation (1)) does not lie in the replacement of ε with ε_p. Instead, the key difference is about the definition of neighboring datasets. For standard DP, neighboring datasets are defined as $D \sim D'$ where D and D' differ in any random data point (given that it is standard DP, every data point has privacy ε). In contrast, in our work, following [15] the neighboring datasets are defined as $D \overset{x_i}{\sim} D'$ where D and D' differ in any arbitrary data point x_i which has a privacy budget of (ε_p, δ). This yields to the individualized notion of DP.

---

[3]In general, we find that it is not necessary to release the privacy budgets: To implement IDP-SGD, only the party who trains the ML model on the individuals' sensitive data needs to know their privacy budgets to set the respective privacy parameters accordingly during training. Given that this party holds access to the data itself, this does not incur any additional privacy disclosure for the individuals. Other parties then only interact with the final private model (but not with the individuals' sensitive data or their associated privacy budgets).

4

Algorithm 1: Finding Sample Parameters.	**Algorithm 2: Finding Scale Parameters.**
The subroutine *getSampleRate* is equivalent to Opacus' function get_noise_multiplier [25]. We sketch its implementation in Algorithm 4 in Appendix C.3.	The subroutine *getNoise* corresponds to Opacus' function get_noise_multiplier [25]. We sketch its implementation in Algorithm 5 in Appendix C.3.

Require: Per-group target privacy budgets $\{\varepsilon_1, \ldots, \varepsilon_P\}$, target δ, iterations I, number of total data points N and per-privacy group data points $\{\|\mathcal{G}_1\|, \ldots, \|\mathcal{G}_P\|\}$.	**Require:** Per-group target privacy budgets $\{\varepsilon_1, \ldots, \varepsilon_P\}$, target δ, iterations I, number of total data points N and per-privacy group data points $\{\|\mathcal{G}_1\|, \ldots, \|\mathcal{G}_P\|\}$, default clip norm c, sample rate q
1: **init** σ_{sample}: $\sigma_{\text{sample}} \leftarrow \sigma$	
2: **init** $\{q_1, \ldots, q_P\}$ where for $p \in [P]$:	1: **init** $\{\sigma_1, \ldots, \sigma_P\}$ where for $p \in [P]$:
3: $\quad q_p \leftarrow$ getSampleRate$(\varepsilon_p, \delta, \sigma_{\text{sample}}, I)$	2: $\quad \sigma_p \leftarrow$ getNoise$(\varepsilon_p, \delta, q, I)$
4: **while** $q \not\approx \frac{1}{N}\sum_{p=1}^{P}\|G_p\|q_p$:	3: **init** σ_{scale}: $\sigma_{\text{scale}} \leftarrow (\frac{1}{N}\sum_{p=1}^{P}\frac{\|\mathcal{G}_p\|}{\sigma_p})^{-1}$ {see derivation of σ_{scale} in Appendix C.2}
5: $\quad \sigma_{\text{sample}} \leftarrow s_i\sigma_{\text{sample}}$ {scaling factor: $s_i < 1$}	4: **for** $p \in [P]$:
6: $\quad$ **for** $p \in [P]$:	5: $\quad c_p \leftarrow \frac{\sigma_{\text{scale}}c}{\sigma_p}$
7: $\quad\quad q_p \leftarrow$ getSampleRate$(\varepsilon_p, \delta, \sigma_{\text{sample}}, I)$	
8: **Output** $\sigma_{\text{sample}}, \{q_1, \ldots, q_P\}$	6: **Output** $\sigma_{\text{scale}}, \{c_1, \ldots, c_P\}$

We formalize the relationship between the individualized notion of DP (which we denote by $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP) and standard (ε, δ)-DP in the following two theorems over the learning algorithm M. The proofs are included in Appendix G.

Theorem 3.1. *An algorithm M that satisfies (ε_1, δ)-DP also satisfies $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP.*

Theorem 3.2. *An algorithm M that satisfies $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP also satisfies (ε_P, δ)-DP.*

3.2 From Individualized Privacy Preferences to Privacy Parameters

From the individual privacy preferences and the total given privacy distribution over the data, *i.e.*, the number of different privacy budgets P, their values ε_p, and the sizes of the respective privacy groups $\|\mathcal{G}_p\|$, we derive individual privacy parameters for IDP-SGD such that all data points within one privacy group (same privacy budget) obtain the same privacy parameters, and such that all privacy groups are expected to exhaust their privacy budget at the same time, after I training iterations.

The individualized parameters for our Sample are the individual sample rates $\{q_1, \ldots, q_P\}$, and a respective noise multiplier σ_{sample}—common to all privacy groups—which we derive from the privacy budget distribution as described in Sec-

Table 1: **(Individualized) Privacy Bounds.**

DP-SGD	SAMPLE	SCALE
$\varepsilon \leq I \cdot 2q^2\frac{\alpha}{\sigma^2}$	$\varepsilon_p \leq I \cdot 2q_p{}^2\frac{\alpha}{\sigma_{\text{SAMPLE}}{}^2}$	$\varepsilon_p \leq I \cdot 2q^2\frac{\alpha}{\sigma_p{}^2}$

tion 3.3. In Scale, the individualized parameters are noise multipliers $\{\sigma_1, \ldots, \sigma_P\}$ with their respective clip norms $\{c_1, \ldots, c_P\}$, as we explain in Section 3.4. Our individual bounds per privacy group are depicted in Table 1. We omit α from consideration since its optimal value is selected as an optimization parameter when converting from RDP to (ε, δ)-DP.

3.3 Our Sample Method

Our Sample method relies on sampling data points with different sample rates $\{q_1, \ldots, q_P\}$ depending on their individual privacy budgets. In this case, the noise multiplier σ_{sample} is fixed. Data points with higher privacy budgets (weaker privacy requirements) are assigned higher sampling rates than those with lower privacy budgets. This modifies the Poisson sampling for DP-SGD (line 3 in Algorithm 3) to sample data points with higher privacy budgets within more training iterations.

Deriving Parameters. We aim at deriving the privacy parameters that yield the specified individual privacy preferences. The learning hyperparameters, such as mini-batch size B or learning rate η can be found through hyperparameter tuning.[4] For Sample, given a tuned mini-batch size B, we have to

---

[4]We discuss in Appendix C.1 other alternatives to implement Sample if we allow changing the *training* hyperparameters.

5

find $\{q_1, \ldots, q_P\}$, such that their weighted average equals q: $\frac{1}{N} \sum_{p=1}^{P} |\mathcal{G}_p| q_p \stackrel{!}{=} q = \frac{B}{N}$. This asserts that the Poisson sampling (line 3 in Algorithm 3) will yield mini-batches of size B in expectation. We also need to ensure that the privacy budgets of all groups will exhaust after I training iterations, which we do by deriving the adequate σ_{sample}, shared across all privacy groups. Equation (2) highlights that the final privacy budget ε depends on both sample rate q and noise multiplier σ. Our Sample method enables higher individualized privacy budgets for the different groups than the standard DP-SGD ε. Yet, we still require the expected average sampling rate to remain q to obtain constant mini-batch size B. Hence, the value of σ_{sample} has to be decreased in comparison to the initial default σ. As an effect of this noise reduction, Sample improves utility of the final model.

Our concrete derivation of values for σ_{sample} and $\{q_1, \ldots, q_P\}$ is presented in Algorithm 1. We start from initializing σ_{sample} with σ from standard DP-SGD, the noise multiplier required for the privacy group $\mathcal{G}_1$ (strongest privacy requirement of all groups) with the smallest privacy budget ε_1. This corresponds to instantiating σ_{sample} with the upper bound noise over all privacy groups. Then, we use a *getSampleRate* function that derives the sampling rate for the given privacy parameters based on approximating Equation (2). Finally, we iteratively decrease σ_{sample} by a scaling factor s_i slightly smaller than one and recompute $\{q_1, \ldots, q_P\}$ until their weighted average is (approximately) equal to q. In Appendix G, we present the formal proof that Sample satisfies $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP). Our proof relies on Mironov et al. [21] and considers training of each privacy group as a separate SGM—with an individual sampling probability—that all simultaneously update the same private model.

3.4 Scale

Our Scale method adjusts the individual noise multipliers $\{\sigma_1, \ldots, \sigma_P\}$ according to data points' individual privacy budgets and performs clipping with individualized clip norms $\{c_1, \ldots, c_P\}$. This changes lines 6 and 8 in the DPSGD Algorithm (3). Data points with higher privacy budgets (weaker privacy requirements) obtain lower noise and higher clip norms. Since Equation (2) highlights that the clip norm c has no direct impact on the obtained ε, individualized privacy in Scale results from the individual noise multipliers $\{\sigma_1, \ldots, \sigma_P\}$. Utility gains come from the overall increase in the signal-to-noise ratio during training.

Deriving Parameters. While the ultimate goal of our Scale method is to adapt individual noise multipliers per privacy group, we cannot implement this directly without degrading training performance. The reason is that whereas in DP-SGD sampling and gradient clipping are performed on a per-data point basis, noise is added per mini-batch, see lines 3, 6 and 8 in Algorithm 3, respectively. However, if we restrict mini-batches to contain only data points from the same privacy group (which share the same noise multiplier), we lose the gains in privacy-utility trade-offs which result from the subsampling (see Appendix F.1 for more details). Hence, while we rely on mini-batches containing data points with different privacy requirements (*i.e.*, different noise multipliers) we can only specify one fixed noise multiplier σ_{scale}.

To overcome this limitation, we do not set noise multipliers $\{\sigma_1, \ldots, \sigma_P\}$ directly, but indirectly obtain them through individualized clip norms $\{c_1, \ldots, c_P\}$ as follows: In standard DP-SGD, Algorithm 3, a gradient clipped to c (line 6) obtains noise with standard deviation σc (line 8). For Scale, we clip gradients to $c_p = s_p c$ with a per-privacy group scaling factor s_p and they obtain noise $\sigma_p c_p$. But in practice, we add noise according to $\sigma_{\text{scale}} c$ to all mini-batches. Hence, the effective scale σ_p of added noise is $\sigma_{\text{scale}} c = \sigma_{\text{scale}} \frac{c_p}{s_p} \stackrel{!}{=} \sigma_p c_p \Rightarrow \sigma_p = \frac{1}{s_p} \sigma_{\text{scale}}$. For data points with higher privacy budgets $s_p > 1$, so their gradients are clipped to larger norms $c_p = s_p c$ and a smaller noise multiplier $\sigma_p = \frac{1}{s_p} \sigma_{\text{scale}}$ is assigned to them. The opposite is true for data points with lower privacy budgets.

We find the values of σ_p required to obtain each privacy groups' desired ε_p using the *getNoise* function (see Algorithm 2 line 1), which takes as inputs the privacy parameters $\varepsilon_p, \delta, q, I$. To optimize utility, we want to set the individual clip norms such that their average over the dataset corresponds to the best clip norm c obtained through initial hyperparameter tuning: $\frac{1}{N} \sum_{p=1}^{P} |\mathcal{G}_p| c_p \stackrel{!}{=} c$. Given the interdependence of c_p, σ_p, and σ_{scale} ($c_p = c \frac{\sigma_{\text{scale}}}{\sigma_p}$), this can be achieved by setting σ_{scale} as the inverse of the weighted average over all $1/\sigma_p$ as: $\sigma_{\text{scale}} = (\frac{1}{N} \sum_{p=1}^{P} \frac{|\mathcal{G}_p|}{\sigma_p})^{-1}$ (please see Appendix C.2 for the derivation of σ_{scale}). Given σ_{scale}, the required σ_p-s, and c found through hyperparamter tuning,

6

the individual clip norms are calculated as $c_p = \frac{\sigma_{\text{scale}} c}{\sigma_p}$. We detail the derivation of the parameters in Algorithm 2. In Appendix G, we formally show that Scale satisfies $(\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_P\}, \delta)$-DP). Similar to Sample, the proof is based on considering the training for all privacy groups as simultaneously executed SGMs with differing sensitivities.

4 Empirical Evaluation

For our empirical evaluation, we implement our methods in Python 3.9 and extend standard Opacus with our individualized privacy parameters and a per-privacy group accounting. We perform evaluation on the MNIST [18], SVHN [22], and CIFAR10 [17] dataset, using the convolutional architectures from Tramer and Boneh [31]. To evaluate the utility of our methods, we use the datasets' standard train-test splits and report test accuracies. The training and standard DP-SGD and IDP-SGD hyperparameters are specified in Table 4 and Table 5 in Appendix D, respectively where the noise multiplier σ is derived with the function `get_noise_multiplier` provided in Opacus [25]. It takes in as arguments the specified parameters δ, q, I, and target ε. For experimentation on individualized privacy, we assigned privacy budgets randomly to the training data points if not indicated otherwise.

4.1 Utility Improvement and General Applicability of Individualization

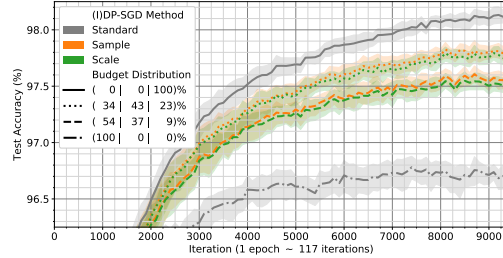
Assigning heterogeneous individual privacy budgets over the training dataset yields significant improvements in terms of the resulting model's utility, as showcased in Table 2. For example, by following the privacy budget distribution of Alaggan et al. [2] with privacy budgets $\varepsilon_p = \{1, 2, 3\}$ for strong, medium, and weak privacy requirements, our Sample method yields accuracy improvements of 1.06%, 2.63%, and 5.09% on MNIST, SVHN, and CIFAR10, respectively. On the CIFAR10 dataset, our Scale even outperforms improvement with an accuracy increase of 5.26% in comparison to the non-individualized baseline, which would have to assign $\varepsilon = 1$ to the whole training set in order to respect each individual's privacy preferences.

The benefits of our individualized privacy assignment also become clear in Figure 1, which depicts the test accuracy of our Sample and Scale vs. standard DP-SGD over the course of training. Both our methods continuously outperform the non-individualized DP-SGD baseline with $\varepsilon = 1$. Additionally, the test accuracy with privacy budget distribution according to Alaggan et al. [2] $(34\%, 43\%, 23\%)$ is higher than the one of Niu et al. [23] $(54\%, 37\%, 9\%)$. This can be explained by the fact that in this latter distribution, more individuals exhibit a higher

Figure 1: **IDP-SGD vs DP-SGD.**

privacy preference. With more data points choosing lower privacy protection as in Alaggan et al. [2], our individualization boosts the performance of the final trained model more significantly. For comparison, we also report results for the standard DP-SGD with $\varepsilon = 3$. This corresponds to the upper bound on utility that could have been achieved by assigning the highest privacy budget to all data points—which violates the privacy requirements of individuals with strong and medium privacy preferences. We observe that our individualized methods' performance is close to this upper bound without violating data points' privacy requirements. In Appendix E, we also discuss other possible baselines (apart from *standard* DP-SGD) and empirically evaluate our individualized methods against them. We report our hyperparameters and the respective individualization parameters (noise multipliers, clip norms, and sampling rates) obtained through our methods in Appendix D. Figure 4 highlights additionally that this privacy parameter derivation is well calibrated: all different groups exhaust their privacy budget simultaneously when the targeted number of iterations is reached.

We also experimentally show that our methods are flexible and generally applicable and can extend to substantially more privacy groups which all have different budgets.[5] The results are shown in Appendix D.2.

---

[5]In principle, with our methods, each data point could have its own privacy budget.

7

Table 2: **Model Test Accuracy after training with Standard DP-SGD vs our Individualized DP-SGD** using Sample or Scale. The percentages of the three privacy groups are chosen according to [2] (first setup) and [23] (second setup). Additional hyperparameters are found in Table 4 and Table 5. We report the standard deviation over 10 trials.

DATASET	PRIVACY GROUPS	PRIVACY BUDGETS	DP-SGD	SAMPLE	SCALE
MNIST	*34%-43%-23%*	*1.0-2.0-3.0*	96.75±0.15	**97.81±0.09**	97.78±0.08
	54%-37%-9%	*1.0-2.0-3.0*	96.75±0.15	**97.6±0.11**	97.54±0.09
SVHN	*34%-43%-23%*	*1.0-2.0-3.0*	83.26±0.31	**85.89±0.14**	85.57±0.24
	54%-37%-9%	*1.0-2.0-3.0*	83.26±0.31	**85.14±0.30**	85.08±0.12
CIFAR10	*34%-43%-23%*	*1.0-2.0-3.0*	52.77±0.65	57.86±0.56	**58.03±0.36**
	54%-37%-9%	*1.0-2.0-3.0*	52.77±0.65	**56.83±0.39**	56.65±0.49

4.2 Privacy Assessment via Membership Inference

To evaluate the impact of our IDP-SGD on the privacy of individual data points, we perform membership inference based on the Lira attack [6]. For the reader's convenience, we include a description of the attack in Appendix B.5. We experiment with CIFAR10 and train 512 shadow models and the target model using Sample on different subsets of 25,000 training data points each. Results for Scale can be found in Appendix D.3. The privacy budgets are set to $\varepsilon = 10$ and $\varepsilon = 20$ and evenly assigned to the shadow models' training data, resulting in 12,500 training data points per privacy budget. Our target model achieves a train accuracy of 68.46% on its 25,000 member data points, and a test accuracy of 64.89% on its 25,000 non-member data points. The lower accuracy in comparison to Carlini et al. [6], who achieved 100% and 92% train and test accuracy respectively, results from us introducing DP to the training of the shadow models. Learning with DP is known to reduce model performance, particularly so on small datasets.

We depict the membership inference risk of the target model's training data per privacy budget (privacy group) in Figure 2. The dotted blue line represents the ROC curve for Lira over all CIFAR10 data points and shows that overall, as expected by its definition, DP protects the training data well against membership inference attacks. However, when inspecting the ROC curve separately for the two privacy groups ($\varepsilon = 10$ and $\varepsilon = 20$), we observe a significant difference between them. The privacy group with stronger privacy guarantees ($\varepsilon = 10$) is protected better (AUC= 0.537) than the group with the higher privacy budget ($\varepsilon = 20$, AUC= 0.581). To evaluate whether the difference in the Lira-likelihood scores between the two privacy groups is statistically significant, we perform a

Figure 2: **Per-Privacy Group Lira [6] Membership Inference Success.**

Student t-test on the likelihood score distributions over the data points with privacy budget $\varepsilon = 10$ vs. the data points with privacy budget $\varepsilon = 20$. We report $\Delta = 2.54$ with $p = 0.01 < 0.05$, and hence a statistically significant difference between the two groups exists.

Experimental results for additional target models and detailed per-target model statistics are presented in Figure 9 and Table 10 in Appendix D.3. These results highlight that the individual privacy assignment of our IDP-SGD approach has a practical impact: it indeed protects data points with different levels of privacy to different degrees.

5 Comparison to Other Methods

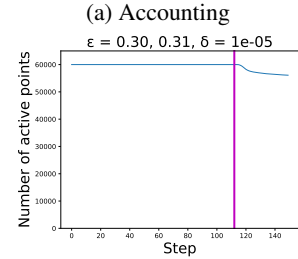
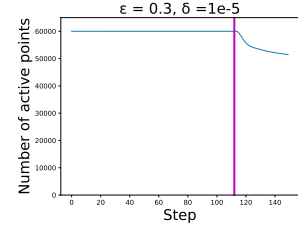
Comparison to Individualized PATE. We compare against Individualized PATE [5] (IPATE), the only other work aiming at enabling individual privacy assignment in ML. See details of their method and the general PATE framework in Appendix B.4. For the comparison, we report the student model accuracies of IPATE for the MNIST, SVHN, and CIFAR10 datasets. The results are presented in Table 11 in Appendix D.4. We observe that IDP-SGD constantly outperforms IPATE with the same privacy budget assignment.

8

Table 3: **Comparison between standard DP-SGD, individual accounting, and individual accounting+assignment.** We present the test accuracy (%) on MNIST for training with $\varepsilon = 0.3$ for standard DP-SGD and individual accounting [9], while setting $\varepsilon = 0.30$ and $\varepsilon = 0.31$ for half of the points each when using individual accounting and assignment. Combining individualized privacy accounting and assignment further increases utility.

Vanilla DP-SGD	Individual Accounting	Individual Accounting and Assignment
93.29 ± 0.49	93.64 ± 0.46	94.16 ± 0.23

Synergy between Individualized Privacy Accounting and Assignment. We show that we can leverage synergies between our individualized privacy assignment and the orthogonal line of work on privacy accounting (see Section 2.2). Therefore, we extend the approach by Feldman and Zrnic [9] with our method of assigning each data point its individual privacy budget through an individualized Renyi-filter. This filter causes the point to be excluded from training once its individually assigned privacy budget is exhausted. Note that in contrast, in the approach by Feldman and Zrnic [9], all data points obtain the same privacy budget and data points are excluded from training once they reach this budget. For more details, see the description in Appendix D.5.

To assess the performance of pure individual accounting, we assign the same privacy budget of 0.3 to all 60,000 training data points in MNIST, following the experimental setup by Feldman and Zrnic [9]. We then enable the individual privacy assignment and change the privacy values to $\varepsilon = 0.3$ for the first half of the points and $\varepsilon = 0.31$ for the second half of data points. We empirically observe that in this low-ε regime, the small change of ε for half the data points is enough to cause significant improvement in the accuracy of the final model, as we show in Table 3. Note that the differences in accuracy reported for standard DP-SGD on MNIST between Table 3 and Table 7 differ. This is because of the different privacy budgets ($\varepsilon = 0.3$ vs. $\varepsilon = 1.0$). Additionally, in Figure 3, we present the number of active data points, *i.e.*, data points that did not yet exhaust their individual privacy budget, over training. We find that many data points are used longer during training with individualized assignment compared to using only individualized accounting. This is because the data points with a higher privacy budget exhaust their budget later and, thus, can be used longer. We are only able to run this experiment on MNIST as done in Feldman and Zrnic [9] since the method is based on batch gradient descent: gradients are computed over the whole dataset at once, which limits its applicability to large and high-dimensional datasets.

(a) Accounting

(b) Accounting+Assignment

Figure 3: **Number of Active Data Points Over Training.**

6 Conclusion and Future Work

Prior work on ML with DP guarantees assigns a uniform privacy budget ε over the entire dataset. This approach fails to capture that different individuals have different expectations towards privacy and also decreases the model's ability to learn from the data. To overcome the limitations of a uniform privacy budget and to implement individual users' privacy preferences, we propose two modifications to the popular DP-SGD algorithm. Our Sample and Scale mechanisms adapt training to yield individual per-data point privacy guarantees and boost utility of the trained models. For future work, we believe that our individualized privacy assignment should be closely integrated with a form of practical individual privacy accounting. This could enable us to obtain a fine-grained notion of individualized privacy guarantees that tightly meet the users' expectations.

9

References

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

[2] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *arXiv preprint arXiv:1504.06998*, 2015.

[3] Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In *Theory of Cryptography Conference*, pages 437–454. Springer, 2010.

[4] Bettina Berendt, Oliver Günther, and Sarah Spiekermann. Privacy in e-commerce: Stated preferences vs. actual behavior. *Communications of the ACM*, 48(4):101–106, 2005.

[5] Franziska Boenisch, Christopher Mühl, Roy Rinberg, Jannis Ihrig, and Adam Dziedzic. Individualized PATE: Differentially Private Machine Learning with Individual Privacy Guarantees, 2022. URL https://arxiv.org/abs/2202.10517.

[6] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1519–1519. IEEE Computer Society, 2022.

[7] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. " i need a better description": An investigation into user expectations for differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3037–3052, 2021.

[8] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[9] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[10] Daniel Franzen, Saskia Nuñez von Voigt, Peter Sörries, Florian Tschorsch, and Claudia Müller-Birn. " am i private and if so, how many?"-communicating privacy guarantees of differential privacy with risk communication formats. *arXiv preprint arXiv:2208.10820*, 2022.

[11] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[12] Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. *arXiv preprint arXiv:2206.07758*, 2022.

[13] Carlos Jensen, Colin Potts, and Christian Jensen. Privacy practices of internet users: Self-reports versus observed behavior. *International Journal of Human-Computer Studies*, 63(1-2):203–227, 2005.

[14] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Differentially private bagging: Improved utility and cheaper privacy than subsample-and-aggregate. In *Advances in Neural Information Processing Systems*, pages pp. 4323–4332, 2019.

[15] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*, pages 1023–1034. IEEE, 2015.

[16] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pages 5213–5225. PMLR, 2021.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

10

[18] Yann LeCun, Corinna Cortes, and C. J. Burges. *MNIST handwritten digit database*. 2010. URL `https://scholar.google.com/citations?user=wln3qraaaaaj&hl=en&oi=sra`.

[19] Haoran Li, Li Xiong, Zhanglong Ji, and Xiaoqian Jiang. Partitioning-based mechanisms under personalized differential privacy. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 615–627. Springer, 2017.

[20] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

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

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[23] Ben Niu, Yahong Chen, Boyang Wang, Jin Cao, and Fenghua Li. Utility-aware exponential mechanism for personalized differential privacy. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2020.

[24] Ben Niu, Yahong Chen, Boyang Wang, Zhibo Wang, Fenghua Li, and Jin Cao. Adapdp: Adaptive personalized differential privacy. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[25] Opacus. Opacus get_noise_multiplier-function. `https://github.com/pytorch/opacus/blob/211acbab1734c865dcfee9edd106c954c58cb10c/opacus/accountants/utils.py#L23`.

[26] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[27] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

[28] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. {Updates-Leak}: Data set inference and reconstruction attacks in online learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1291–1308, 2020.

[29] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[30] Peter Sörries, Claudia Müller-Birn, Katrin Glinka, Franziska Boenisch, Marian Margraf, Sabine Sayegh-Jodehl, and Matthias Rose. Privacy needs reflection: Conceptional design rationales for privacy-preserving explanation user interfaces. *Mensch und Computer 2021-Workshopband*, 2021.

[31] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.

[32] Aiping Xiong, Tianhao Wang, Ninghui Li, and Somesh Jha. Towards effective differential privacy communication for users' data sharing decision and comprehension. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 392–410. IEEE, 2020.

[33] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[34] Da Yu, Gautam Kamath, Janardhan Kulkarni, Tie-Yan Liu, Jian Yin, and Huishuai Zhang. Individual privacy accounting for differentially private stochastic gradient descent. *arXiv preprint arXiv:2206.02617*, 2022.

[35] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

11

A Broader Impacts and Ethical Considerations of this Work

The specification of an adequate privacy level ε is challenging since it does not only depend on the domain but also on the dataset itself. This challenge persists when assigning individual privacy levels over a dataset. Especially in sensitive domains, one needs to make sure that the assignment of individual privacy guarantees does not pose an additional risk to the individuals whose data is processed. Therefore, one needs to first make sure that assigning individual privacy guarantees is not abused. This can occur when the entity training the model violates underlying individuals' privacy by nudging them into choosing inadequate privacy levels. Second, one needs to prevent individuals from giving up their privacy due to a lack of understanding or a wrong perception of their risk. To do so, IDP-SGD should not be used as a stand-alone method. Instead, following the example of [5], we suggest incorporating it into a whole process that involves communicating the functioning of DP and the risks associated with a decision to the individuals, analyzing their preferences, and supporting them in an informed decision-making process. There exist a growing body of work on the communication of DP methodology and associated risks to the public [32, 7, 10]. The most recent finding [10] suggests that medical risk communication formats, such as percentages or frequencies, are best applicable to inform about privacy risks. For the identification of privacy preferences, existing frameworks, such as [30] can be applied. Finally, we suggest giving individuals the choice between categorical privacy levels (*e.g.*, *high*, *medium*, *low*) while putting a regulatory entity, such as an ethics committee in charge of specifying concrete numeric values of ε. This mitigates the disadvantages of the general limited interpretability of ε that exist independently of individualization.

Limitations. In this work, we consider a setup where individuals indicate their privacy preferences. We acknowledge that not all individuals are aware of their own privacy preferences and might, therefore, provide inaccurate indications. Therefore, our framework should be always deployed in a protected setup as described above. While we provide theoretical privacy guarantees in the framework of differential privacy, we assess the practical impact of these guarantees on the individuals solely through membership inference attacks. Future work should investigate the practical (disparate per privacy-group) impact on other known privacy risks, such as data reconstruction. Due to the lack of sensitive-real world datasets that, in addition to individuals' data, also capture their privacy preferences, we evaluate our methods on standard benchmark datasets. We, therefore, have to *simulate* a privacy preference distribution on this data according to the distribution known from society.

B Extended Background

B.1 Differential Privacy

Differential Privacy (DP) [8] provides a theoretical upper bound on the influence that a single data point can have on the outcome of an analysis over the whole dataset.

The most commonly applied instantiation of DP is (ε, δ)-DP which is defined as follows:

Definition B.1. Let $D, D' \subseteq \mathcal{D}$ be two neighboring datasets, *i.e.*, datasets that differ in exactly one data point. Let further $M : \mathcal{D}^* \to \mathcal{R}$ be a mechanism that processes an arbitrary number of data points. M satisfies (ε, δ)-DP if for all datasets $D \sim D'$, and for all result events $R \subseteq \mathcal{R}$

$$\mathbb{P}\left[M(D) \in R\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta. \tag{4}$$

In the definition, the privacy level is specified by $\varepsilon \in \mathbb{R}_+$, while $\delta \in [0, 1]$ offers a relaxation, *i.e.*, a small probability of violating the guarantees.

In Algorithm 3, we highlight the DP-SGD algorithm that allows to train ML models with DP guarantees.

B.2 Differential Privacy Accounting in Machine Learning

Privacy accounting in DP-SGD is most commonly implemented by the moments accountant which keeps track of a bound on the moments of the privacy loss random variable at outcome R, defined by

$$c(R; M, \mathsf{aux}, D, D') = \log \frac{\Pr[M(\mathsf{aux}, D) \in R]}{\Pr[M(\mathsf{aux}, D') \in R]}. \tag{5}$$

for D, D', M and R as above, and an auxiliary input aux.

12

Algorithm 3: Differentially Private SGD [1]

Require: Training dataset $\mathcal{D}$ with data points $\{x_1, \ldots, x_N\}$, loss function l, learning rate η, noise multiplier σ, sampling rate q, clip norm c, number of training iterations I.

1: **Initialize** θ_0 randomly
2: **for** $t \in [I]$ **do**
3:     *Poisson Sampling*: Sample mini-batch L_t with per-point probability q from $\mathcal{D}$.
4:     For each $i \in L_t$, compute $g_t(x_i) \leftarrow \nabla_{\theta_t} l(\theta_t, x_i)$
5:     **1. Gradient Clipping**
6:     $\bar{g}_t(x_i) \leftarrow g_t(x_i) / \max\left(1, \frac{\|g_t(x_i)\|_2}{c}\right)$
7:     **2. Noise Addition**
8:     $\tilde{g}_t \leftarrow \frac{1}{|L_t|}\left(\sum_i \bar{g}_t(x_i) + \mathcal{N}(0, (\sigma c)^2 \mathbf{I})\right)$
9:     $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
10: **end for**
11: **Output** θ_T, privacy cost (ε, δ) computed using a privacy accounting method.

B.3 Formalizing Individualized Privacy

Individualized privacy guarantees with DP can be formalized as follows [15]:

Definition B.2. Let $d \in \mathcal{D}$ be a data point. M satisfies $(\varepsilon_d, \delta_d)$-DP if for all datasets $D \overset{d}{\sim} D'$, and for all result events $R \subseteq \mathcal{R}$

$$\mathbb{P}\left[M(D) \in R\right] \leq e^{\varepsilon_d} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta_d. \tag{6}$$

Note that other notions of DP (*e.g.*, RDP) can analogously be generalized to enable per-data point privacy guarantees.

B.4 PATE Algorithm

The Private Aggregation of Teacher Ensembles (PATE) algorithm [26] represents an alternative to DP-SGD for training ML models with privacy guarantees. This ensemble-based algorithm implements privacy guarantees through a knowledge transfer from the ensemble to a separate student model. More concretely, in PATE, the private training data is split into non-overlapping subsets and distributed among several teacher models. Once each teacher is trained on their own data subset, they perform a privacy-preserving knowledge transfer by jointly labeling an additional unlabeled public dataset. To implement DP guarantees, noise is added to the labeling process. On completion, an independent student model is trained on the public dataset using the generated labels, and thereby incorporating knowledge about the original training data without ever requiring access to it.

Individualized PATE. The individualized PATE variants by [5] are *Upsample* and *Weight*. Upsample duplicates data points and distributes them to different teachers in the PATE ensemble. Utility increase in this method result from the availability of more training data points for the teachers. Since the sensitivity for a duplicated data point increases (it can change the votes of all the teachers that are trained on its duplicates), privacy consumption of that data point is higher. Since each data point can be duplicated individually, this method allows for a fine-grained individualization. In contrast, the weight method allows for a per-teacher model privacy individualization. Data points with the same privacy budget are assigned to the same teacher model(s) and the impact of that teacher model's weight on the final vote is weighted according to its training data points' privacy preferences. Teachers that are trained on data points with low privacy requirements are weighted higher, increasing the privacy consumption of their training data.

B.5 The Lira Membership Inference Attack

The Lira membership inference attack [6] proceeds in three steps to determine which data points from a dataset $\mathcal{D} = \{x_1, \ldots, x_N\}$ were used to train a target model f: (1) First, multiple shadow models, similar to f, are trained on different subsets of $\mathcal{D}$. (2) Then, the mean and variance of two loss distributions $\mathcal{N}(\mu_{\text{in}}, \sigma_{\text{in}})$ and $\mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}})$ are estimated per data point x_i. Both distributions

13

are calculated from the logits of x_i at the target class y_i—the former one over the shadow models that x_i is a member of, the latter one on shadow models that x_i is not a member of. (3) Finally, the likelihood of a new data point x of class y being a member of the target model f is calculated as $\Lambda = \frac{p(f(x)_y \mid \mathcal{N}(\mu_{\text{in}}, \sigma_{\text{in}}^2))}{p(f(x)_y \mid \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}}^2))}$.

C   Details on the Methods

C.1   Sample

Leveraging Higher Sampling Rates for Increased Utility.   With higher sampling rates for certain data points, utility could, in principle be increased in several ways. (1) Larger sampling rates can be used to obtain higher mini-batch sizes B (while keeping the number of training iterations I constant). Line 3 in Algorithm 3 shows that noise is added to the aggregate of all gradients. Hence, with larger mini-batches, the signal-to-noise ratio is higher, which can improve training. (2) Alternatively, the mini-batch size B can be kept constant while increasing the number of training iterations I. Longer training can increase model performance. However, these two approaches result in a change of core training hyperparamters (mini-batch size and number of iterations). As we discuss in Section 3, changing training hyperparameters would require a separate fine-tuning, for example, to adapt the learning rate for larger mini-batches, as in (1) or longer training as in (2). Since the training parameters would change according to the privacy budgets encountered in the private training dataset, and the ratios of these budgets over the training data points, the hyperparameter-tuning would have to be repeated whenever the dataset is updated, individuals change their privacy preferences, or decide to withdraw their consent for leveraging their data for the ML model alltogether, yielding significant overheads. We, therefore, implement our Sample according to the third option (3), described in Section 3.3 which leverages higher sampling rates for improved utility by reducing the noise multiplier of the added noise σ. This allows us to perform an apple to apple comparison between both our methods and to the standard DP-SGD.

C.2   Scale

Deriving Noise Multiplier σ_{scale}.   Given the desired clip norm c found through hyperparameter tuning of the standard DP-SGD, we set the individual clip norms such that their weighted average yields c as $c = \frac{1}{N} \sum_{p=1}^{P} |G_p| \cdot c_p$. So, we derive the σ_{scale} in the following way:

$$c = \frac{1}{N} \sum_{p=1}^{P} |G_p| \cdot c_p \tag{7}$$

$$c = \frac{1}{N} \sum_{p=1}^{P} |G_p| \cdot \left(c \frac{\sigma_{\text{scale}}}{\sigma_p} \right) \tag{8}$$

$$c = c \sigma_{\text{scale}} \frac{1}{N} \sum_{p=1}^{P} \frac{|G_p|}{\sigma_p} \tag{9}$$

$$\sigma_{\text{scale}} = \left(\frac{1}{N} \sum_{p=1}^{P} \frac{|G_p|}{\sigma_p} \right)^{-1} \tag{10}$$

From (7) to (8), we use the equality between the scale of added noise $\sigma_{\text{scale}} c = \sigma_p c_p$. In (9), we extract terms that are independent of the privacy groups (c and σ_{scale}) before the summation.

C.3   Algorithmic Details

We specify our used sub-routines used for determining a sample rate or noise multiplier based on given privacy parameters in Algorithm 4 and Algorithm 5, respectively.

14

Algorithm 4: Subroutine getSampleRate. Is the equivalent to Opacus' function get_noise_multiplier [25] for deriving an adequate sample rate for given paramteers.

Require: Target ε, target δ, iterations I, noise multiplier σ, precision $\gamma = 0.01$

1: **init** $\varepsilon_{\text{high}}$: $\varepsilon_{\text{high}} \leftarrow \infty$
2: **init** $q_{\text{low}}, q_{\text{high}}$: $q_{\text{low}} \leftarrow 1\text{e}{-}9, q_{\text{high}} \leftarrow 0.1$
3: **while** $\varepsilon_{\text{high}} > \varepsilon$ **do**
4:     $q_{\text{high}} \leftarrow 2q_{\text{high}}$
5:     $\varepsilon_{\text{high}} \leftarrow I \cdot 2q_{\text{high}}^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
6: **end while**
7: **while** $\varepsilon - \varepsilon_{\text{high}} > \gamma$ **do**
8:     $q \leftarrow (q_{\text{low}} + q_{\text{high}})/2$
9:     $\varepsilon_{\text{temp}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
10:     **if** $\varepsilon_{\text{temp}} < \varepsilon$ **then**
11:         $q_{\text{high}} \leftarrow q$
12:         $\varepsilon_{\text{high}} \leftarrow \varepsilon_{\text{temp}}$
13:     **else**
14:         $q_{\text{low}} \leftarrow q$
15:     **end if**
16: **end while**
17: **Output** q_{high}

Algorithm 5: Subroutine getNoise. Implements Opacus' function get_noise_multiplier [25].

Require: Target ε, target δ, iterations I, sample rate q, precision $\gamma = 0.01$

1: **init** $\varepsilon_{\text{high}}$: $\varepsilon_{\text{high}} \leftarrow \infty$
2: **init** $\sigma_{\text{low}}, \sigma_{\text{high}}$: $\sigma_{\text{low}} \leftarrow 0, \sigma_{\text{high}} \leftarrow 10$
3: **while** $\varepsilon_{\text{high}} > \varepsilon$ **do**
4:     $\sigma_{\text{high}} \leftarrow 2\sigma_{\text{high}}$
5:     $\varepsilon_{\text{high}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma_{\text{high}}^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
6: **end while**
7: **while** $\varepsilon - \varepsilon_{\text{high}} > \gamma$ **do**
8:     $\sigma \leftarrow (\sigma_{\text{low}} + \sigma_{\text{high}})/2$
9:     $\varepsilon_{\text{temp}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
10:     **if** $\varepsilon_{\text{temp}} < \varepsilon$ **then**
11:         $\sigma_{\text{high}} \leftarrow \sigma$
12:         $\varepsilon_{\text{high}} \leftarrow \varepsilon_{\text{temp}}$
13:     **else**
14:         $\sigma_{\text{low}} \leftarrow \sigma$
15:     **end if**
16: **end while**
17: **Output** σ_{high}

D Additional Empirical Evaluation

We report the hyperparameters found for our individualized methods in Table 5. The training and standard DP-SGD hyperparameters are specified in Table 4. The performance of our individualized methods when using the hyperparameters of standard DP-SGD is presented in Table 7. Already when using these (non-individually tuned) hyperparameters, our methods yield a significant performance increase in comparison to standard DP-SGD. For MNIST, individual hyperparameter for our methods and individual setups did not yield significant improvements, therefore the results presented in Table 7 and Table 2 are identical for MNIST.

15

Computing Resources. The implementation of our methods does not increase computation time over the standard implementation of DP-SGD apart from the derivation of the privacy parameters that is performed once at the beginning of training. Hence, to run all experiments around our methods and their evaluation, we required, in total less than 16h of GPU time on a standard GeForce RTX 2080 Ti. We ran the experiment on combining individualized privacy assignment and accounting on the same machines RTX 2080Ti and the total compute time is also around 2h. To train all the shadow models for our membership inference attack and run inference on them, we ran on an A100 GPU and required a total runtime of roughly 32 hours.

Table 4: **DP-SGD Hyperparameters.** LR: learning rate, B: expected mini-batch size, I: number of iterations, C: clip norm, σ: noise multiplier in DP-SGD derived from the desired privacy budget $\varepsilon = 1$. Default target $\delta = 0.00001$.

DATASET	LR	B	I	C	σ
MNIST	0.6	512	9375~80 EPOCHS	0.2	3.42529
SVHN	0.2	1024	2146~30 EPOCHS	0.9	2.74658
CIFAR10	0.7	1024	1465~30 EPOCHS	0.4	3.29346

Table 5: **DP-SGD Hyperparameters (Individually Tuned).** LR: learning rate, B: expected mini-batch size, I: number of iterations, C: clip norm, σ. Default target $\delta = 0.00001$. Setup A is for privacy budgets $\varepsilon = \{1.0, 2.0, 3.0\}$ and their respective distribution of *34%-43%-23%*. Setup B is for the same privacy budgets but with their distributions *54%-37%-9%*.

DATASET	METHOD	SETUP	LR	B	I	C	σ
MNIST	SAMPLE	A	0.6	512	9375~80 EPOCHS	0.2	3.42529
SVHN	SAMPLE	A	0.2	1024	5723~80 EPOCHS	0.6	2.53261
CIFAR10	SAMPLE	A	0.2	1024	2929~60 EPOCHS	1.0	2.65712
MNIST	SAMPLE	B	0.6	512	9375~80 EPOCHS	0.2	3.42529
SVHN	SAMPLE	B	0.1	1024	3577~50 EPOCHS	0.6	2.41421
CIFAR10	SAMPLE	B	0.1	1024	2929~60 EPOCHS	1.8	3.14049
MNIST	SCALE	A	0.6	512	9375~80 EPOCHS	0.2	3.42529
SVHN	SCALE	A	0.1	1024	3577~50 EPOCHS	2.0	2.09719
CIFAR10	SCALE	A	0.2	1024	3418~70 EPOCHS	1.1	2.88335
MNIST	SCALE	B	0.6	512	9375~80 EPOCHS	0.2	3.42529
SVHN	SCALE	B	0.1	1024	3577~50 EPOCHS	1.6	2.45703
CIFAR10	SCALE	B	0.1	1024	2929~60 EPOCHS	1.8	3.17792

We present the individualized privacy parameters identified for our methods in Table 6.

Table 6: **Individualization Parameters Computed by our Methods for Table 7.** We report the individualized privacy parameters identified for our Scale and Sample by Algorithm 2 and Algorithm 1, respectively. The parameters are obtained on the MNIST, SVHN, and CIFAR10 datasets when using the privacy budget distributions of Table 7 with $\varepsilon = \{1.0 - 2.0 - 3.0\}$

DATASET	SETUP	DP-SGD			SCALE			SAMPLE	
		σ	c	q	σ_{SCALE}	$\{\sigma_1,\ldots,\sigma_P\}$	$\{c_1,\ldots,c_P\}$	σ_{SAMPLE}	$\{q_1,\ldots,q_P\}$
MNIST	*34%-43%-23%*	3.425	0.2	0.008	2.063	{2.189, 1.310, 1.032}	{0.129, 0.216, 0.274}	2.024	{0.005, 0.009, 0.013}
	54%-37%-9%	3.425	0.2	0.008	2.418	{2.189, 1.310, 1.032}	{0.148, 0.248, 0.315}	2.376	{0.006, 0.011, 0.016}
SVHN	*34%-43%-23%*	2.747	0.9	0.014	1.896	{2.747, 1.589, 1.214}	{0.561, 0.970, 1.270}	1.667	{0.008, 0.015, 0.021}
	54%-37%-9%	2.747	0.9	0.014	2.180	{2.747, 1.589, 1.214}	{0.651, 1.125, 1.472}	1.937	{0.009, 0.018, 0.025}
CIFAR10	*34%-43%-23%*	3.293	0.4	0.020	2.244	{3.294, 1.868, 1.399}	{0.244, 0.430, 0.574}	1.965	{0.012, 0.022, 0.031}
	54%-37%-9%	3.293	0.4	0.020	2.594	{3.294, 1.868, 1.399}	{0.285, 0.502, 0.671}	2.300	{0.014, 0.026, 0.037}

D.1 Privacy Consumption of our Methods

We track privacy consumption of our methods over the course of training in Figure 4. The figure highlights the good calibration of our methods which causes all privacy groups to exhaust their budget after the pre-specified number of training iterations.

16

Table 7: **Model Test Accuracy after training with Standard DP-SGD vs our Individualized DP-SGD** using Sample or Scale. **D** is the distribution of privacy groups (percentages) and ε the privacy budget for a given group. The percentages of the three privacy groups are chosen according to Alaggan et al. [2] (first setup) and [23] (second setup). We used the hyperparameters found for standard DP-SGD, see Table 4 and report the standard deviation over 10 trials.

DATASET	SETUP		DP-SGD	SAMPLE	SCALE
MNIST	*D*	*34%-43%-23%*	96.75	**97.81**	97.78
	ε	*1.0-2.0-3.0*	± 0.15	± **0.09**	± 0.08
	D	*54%-37%-9%*	96.75	**97.6**	97.54
	ε	*1.0-2.0-3.0*	± 0.15	± **0.11**	0.09
SVHN	*D*	*34%-43%-23%*	83.26	**84.56**	84.48
	ε	*1.0-2.0-3.0*	±0.31	±**0.25**	±0.25
	D	*54%-37%-9%*	83.26	**84.32**	84.31
	ε	*1.0-2.0-3.0*	±0.31	±**0.31**	±0.26
CIFAR10	*D*	*34%-43%-23%*	52.77	54.89	**54.92**
	ε	*1.0-2.0-3.0*	± 0.65	± 0.55	±**0.63**
	D	*54%-37%-9%*	52.77	54.88	**55.00**
	ε	*1.0-2.0-3.0*	± 0.65	± 0.45	± **0.65**

(a) Sample

(b) Scale

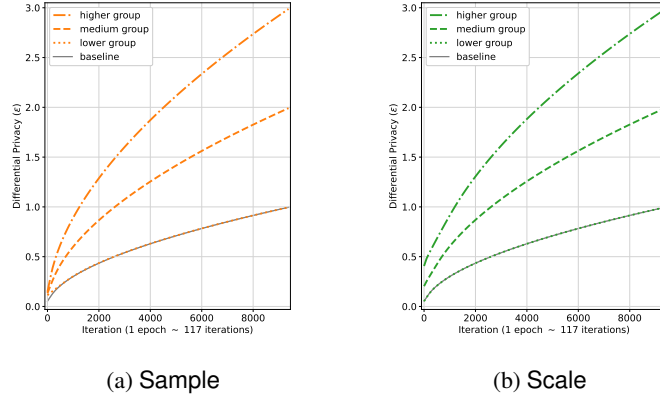



Figure 4: **Individual Privacy Costs on MNIST** for $\varepsilon \in \{1, 2, 3\}$ with Distribution $(54\%, 37\%, 9\%)$.

D.2 General Applicability of our Methods

We showcase the practical impact of individual privacy assignments on individual utility and demonstrate how our methods extend to many privacy groups and privacy budget distributions.

Practical Impact. We run experiments on the CIFAR10 dataset where we assign higher or lower privacy budgets to one of the 10 classes. We select all data points from the class 0 as the first privacy group and assign to it either higher ($\varepsilon = 3$), the same ($\varepsilon = 2$), or lower ($\varepsilon = 1$) privacy budgets in comparison to all other data points from the other classes ($\varepsilon = 2$). Table 8 shows that the choice of privacy budget for a single group also impacts the other groups. We observe that by only changing the privacy budget for the selected group (in this case for class 0), we can flip its performance (its accuracy from being higher to being lower) in comparison to the accuracy of the other group (consisting of remaining classes). In the example of class 0, the accuracy is 67.76% when assigned high privacy budget ($\varepsilon = 3$), which is a higher accuracy than for all other classes that have an average accuracy of around 53.41% and assigned the privacy budget $\varepsilon = 2$. Then, by modifying only the privacy budget of class 0 and by assigning to it the low privacy budget ($\varepsilon = 1$), its accuracy drops to a mere 25.76% and is below the accuracy of 56.58% for the remaining classes. We visualize the impact of the chosen privacy budget on utility over all classes (instead of only the class 0) in Figure 5 and Figure 6 for CIFAR10 and MNIST, respectively.

More Privacy Groups. We present in Table 9 the test accuracy for ten privacy groups, corresponding to the ten classes of the CIFAR10 dataset when each of the privacy groups obtains a different privacy budget. We obtained these budgets by manually tuning them such that the accuracy gap be-

17

tween the privacy groups is minimized. We also visualize the accuracy over training in Figure 7. The figure visualizes that our methods are able to make all privacy groups converge to similar accuracies.

Table 8: **Accuracy for Subgroups.** We assess the accuracy of subgroups when their privacy budgets differ. We select a single class for a given group and assign either higher, the same, or lower privacy budgets in comparison to groups with other classes. We change the privacy budgets only for bolded classes in a given experiment while all other classes have the same privacy budget ($\varepsilon = 2$).

Classes	Privacy Budget		
	Higher ($\varepsilon = 3$)	Same ($\varepsilon = 2$)	Lower ($\varepsilon = 1$)
0	**67.76 ± 2.14**	55.24 ± 1.98	25.76 ± 2.52
1-9	53.41 ± 2.2	54.72 ± 2.49	**56.58 ± 2.29**
1	**80.84 ± 1.2**	72.79 ± 1.6	46.09 ± 3.91
0,2-9	51.65 ± 2.28	52.77 ± 2.53	**54.59 ± 2.23**
2	51.31 ± 3.53	36.59 ± 3.33	9.53 ± 1.41
0-1,3-9	**54.26 ± 2.84**	56.79 ± 2.34	**58.87 ± 2.18**
3	52.88 ± 2.6	32.64 ± 1.91	6.67 ± 1.09
0-2,4-9	**54.62 ± 2.42**	57.23 ± 2.5	**58.75 ± 2.42**
4	**56.99 ± 1.87**	40.06 ± 2.88	9.44 ± 1.68
0-3,5-9	54.41 ± 2.21	56.41 ± 2.39	**58.18 ± 2.02**
5	**64.11 ± 2.27**	51.86 ± 2.21	15.04 ± 2.31
0-4,6-9	53.28 ± 2.16	55.1 ± 2.46	**57.54 ± 2.49**
6	**73.54 ± 2.36**	65.8 ± 4.25	40.6 ± 4.18
0-5,7-9	52.05 ± 2.21	53.55 ± 2.23	**56.06 ± 2.33**
7	**68.22 ± 1.17**	61.15 ± 1.99	41.08 ± 2.75
0-6,8-9	52.5 ± 2.43	54.07 ± 2.49	**56.01 ± 2.76**
8	**77.39 ± 1.24**	68.53 ± 2.34	37.42 ± 3.37
0-7,9	51.51 ± 2.54	53.25 ± 2.45	**55.28 ± 2.37**
9	**72.82 ± 1.48**	63.08 ± 1.9	32.79 ± 2.73
1-8	52.05 ± 2.32	53.85 ± 2.5	**56.06 ± 2.52**

D.3 Additional Results for MIA

Membership inference success for a single target model of our Sample and Scale methods is shown in Figure 8. In Figure 9, we present the results over for 5 different target models for Sample. The figure highlights that over all target models, the two privacy groups' privacy risk is different: the group with higher protection $\varepsilon = 10$ constantly has a lower AUC than the group with lower protection $\varepsilon = 20$. The test statistics over the different privacy groups' Lira likelihood scores for all five target models are shown in Table 10.

D.4 Comparison to Individualized Privacy with IPATE

We present the comparison between our IDP-SGD and IPATE [5] in Table 11. In PATE, accuracy refers to the student model accuracy. The results in Table 11 are averaged over three experiments for IPATE and ten runs for IDP-SGD. Note that the accuracies we report for IPATE differ from the accuracy values reported by [5] in Table 1, since they report average voting accuracy (*i.e.*, how correct are individual teacher model votes), whereas we report the resulting student model accuracies, which corresponds to the final performance of the method. Note that IPATE does not apply the performance improvements suggested by Papernot et al. [27] (*e.g.*,virtual adversarial training) or MixMatch which could increase the student model's performance.

18

(a) **Sample** (higher budget)

(b) **Scale** (higher budget)

(c) **Sample** (lower budget)

(d) **Scale** (lower budget)

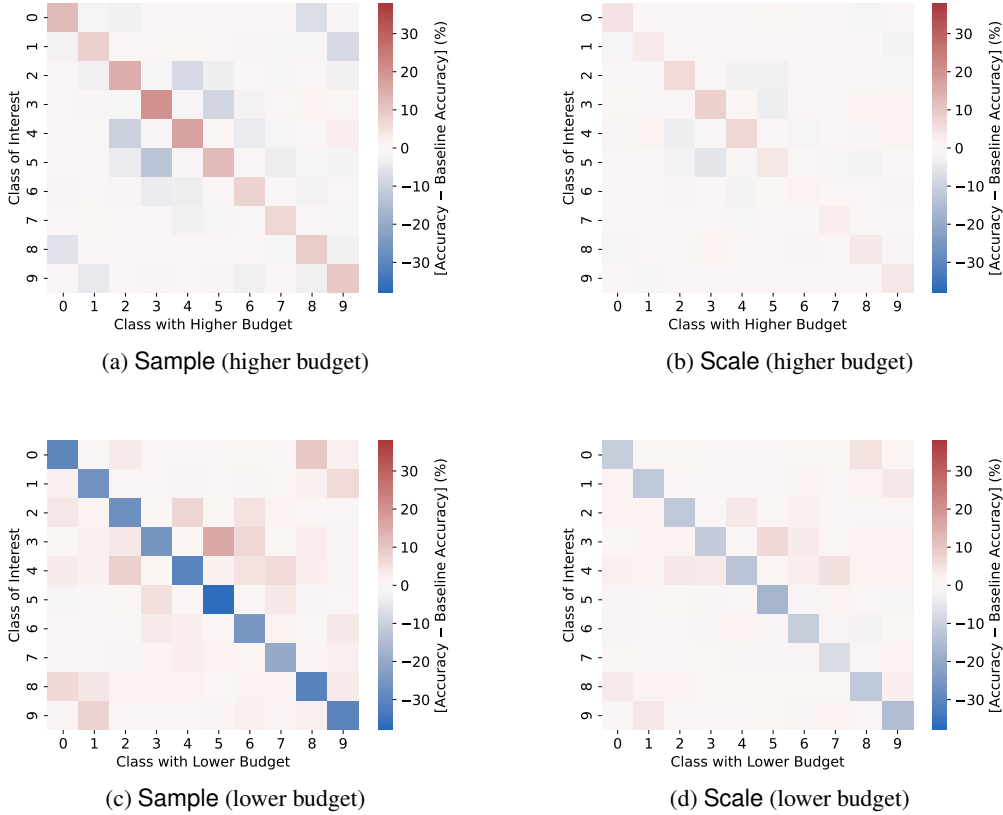



Figure 5: **CIFAR10: Accuracy Changes for Subgroups.** We assess how the test **Accuracy** of a **Class of interest** changes in comparison to the **Baseline Accuracy** (standard DP-SGD with $\varepsilon = 2$) when we, during training, assign a lower ($\varepsilon = 1$) or a higher ($\varepsilon = 3$) privacy budget to data points from a class (shown on the x-axis). The diagonals show that by increasing a class' privacy budget (lower privacy), their utility increases, while it decreases with the decrease of privacy budget (higher privacy). Similar results for MNIST can be found in Figure 6.

D.5   Integrating Privacy Accounting and Assignment

The main goal behind individualized accounting is to obtain tighter privacy analysis (recall our discussion in Section 2.2). Instead of tracking a single privacy loss estimate across all data points, an individual privacy loss is kept track of for each data point. Feldman and Zrnic [9] defines a new individual privacy filter, which drops data points that exceed the individualized privacy loss from further processing. However, the same privacy budget is assigned to each data point. The individual assignment of a privacy budget to each data point is a natural extension of individualized accounting and can be directly incorporated into this framework. Therefore, a given data point has its own privacy filter and is dropped from the analysis once the filter indicates that the data point's privacy budget is exhausted.

E   Alternative Baselines

Throughout this work, we compare our methods with Standard DP-SGD which cannot take into account different privacy requirements at the same time. Thus, we consider it to apply the highest privacy protection required to all data points equally. Nonetheless, we can think of two more ways to use Standard DP-SGD on data having heterogeneous privacy requirements. Those approaches have different benefits and drawbacks and might perform better than our chosen baseline approach in some scenarios.

19

(a) Sample | higher budget

(b) Scale | higher budget

(c) Sample | lower budget

(d) Scale | lower budget

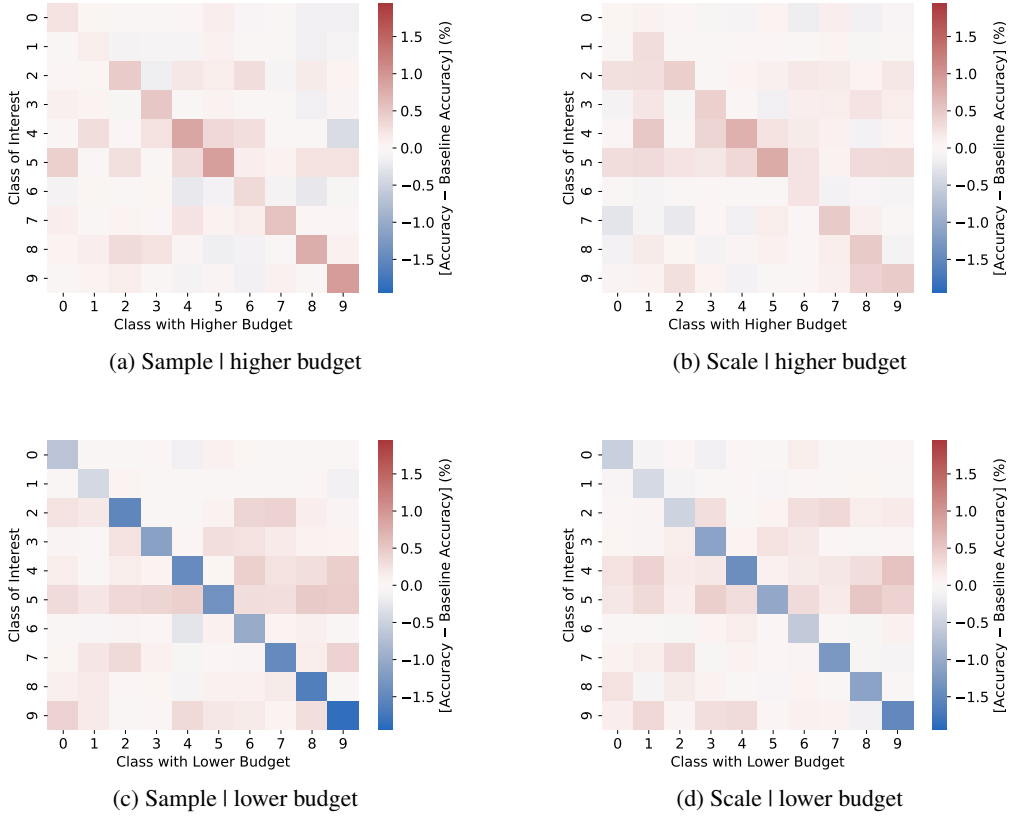



Figure 6: **MNIST: Accuracy Changes for Subgroups.** We assess how the test accuracy of a class changes (in comparison to standard DP-SGD with $\varepsilon = 2$) when we, during training, assign a lower ($\varepsilon = 1$) or a higher ($\varepsilon = 3$) privacy budget to data points from this class. The diagonals show that by increasing a class' privacy budget (lower privacy), their utility increases, while it decreases with the decrease of privacy budget (higher privacy).

E.1 Exclude Lower Privacy Groups

Instead of applying the strongest privacy protection, the deciding ML expert could entirely exclude data of low privacy groups from training for loosening the restrictions on the remaining data points' influence on model updates. In some cases, it would be worth giving up the information and privacy budgets of those lower privacy groups to achieve utility improvements. This approach performs poorly if important information is wasted, *e.g.*, most data of one class has the highest privacy requirement.

E.2 Learn Privacy Groups Separately

It is also possible to make use of all privacy budgets, independent of their diversity, although only using Standard DP-SGD. Namely, a model can be trained on each privacy group separately one after another, whereby the corresponding lowest budget is regarded for each group. A drawback of this approach is that the model could forget its knowledge about previously learned privacy groups.

E.3 Empirical Comparison of Baselines

We empirically evaluate against these two additional baselines using the MNIST dataset. For baseline E.1, we include all data points with a privacy budget of $\varepsilon \geq 2$ und use $\varepsilon = 2$ as the privacy budget for training. After hyperparameter tuning, the training on the remaining data points (43%+23% and 37%+9% of the total data) yields the accuracy reported in Appendix E.3.

20

Table 9: **Per-class Individual Privacy Assignments.** We manually optimize the per-class individual privacy budgets for Sample such that the model achieves the same accuracy over all classes. The resulting per-class privacy budgets yield the maximum gap Δ between the highest and lowest accuracy level of only 0.39% for Sample, and 0.88% for Scale. For the baseline ($\varepsilon = 3$ for all classes) $\Delta = 2.03$ is significantly higher, highlighting that our approach can successfully minimize the accuracy gap between different privacy groups. We run the experiment on the MNIST dataset and report average per-class test-accuracies over three separate runs. See each privacy group's test accuracy over training in Figure 7.

Class	0	1	2	3	4	5	6	7	8	9	Δ
Baseline ($\varepsilon = 3$)	98.95	99.06	98.39	98.09	97.93	98.47	98.16	98.12	97.78	97.03	2.03
Budgets	0.75	0.5	2.0	2.6	4.1	2.1	2.05	3.0	3.1	6.1	/
Sample	98.16	98.09	98.16	97.95	98.10	97.91	97.77	97.99	98.02	97.89	0.39
Scale	98.44	98.36	98.13	98.02	97.76	98.17	97.91	97.96	97.91	97.56	0.88

Table 10: **Statistical differences between Privacy Groups.** We conduct a student t-test to determine if the Lira likelihood scores for data points with privacy budget $\varepsilon = 10$ differ from the ones of data points with $\varepsilon = 20$. All results with $p < 0.05$ indicate statistically significant differences. Results for Sample.

Target Model	Δ	p
1	5.16	2.49e-07
2	2.41	0.016
3	1.84	0.066
4	-4.03	5.52e-05
5	2.537	0.011

For baseline E.2, we also did hyperparameter tuning and used the best noise multiplier of 2.5 for training. We trained the groups sequentially, always continuing training with the next group once the privacy budget of the previous groups was exhausted. We evaluated both starting with the privacy group that has loosest and strongest preferences (orders [3,2,1] and [1,2,3], respectively). Starting with the group that has strongest privacy requirements and ending on the group that has loosest privacy requirements yielded the best results which we report in Appendix E.3.

In summary, we observe that our methods outperform the other baselines.

F   Alternative Individualization

We present the alternative ways of individualizing privacy guarantees in DP-SGD that we considered in the design process of IDP-SGD and describe their drawbacks.

F.1   Individual Per-Data Point Noise

Individualized privacy could, in principle also be obtained by adding different amounts of noise to different data points. Every of the P privacy group would have their individual $\{\sigma_1, \ldots, \sigma_P\}$. Utility improvements would result from some data points requiring smaller amounts of added noise. Note however, that in DP-SGD, while clipping is performed on a per-data point basis, noise addition is performed on a per-mini-batch basis (line 8 in Algorithm 3). Hence, there are two possibilities to implement individual noise addition: either (i) by operating on mini-batch sizes of 1, or (ii) by implementing a two-step sampling approach which first randomly samples a privacy group for a given training iteration and then applies the standard Poisson sampling to obtain the mini-batch consisting of data points from this group. While both approaches are conceptually correct, they exhibit significant drawbacks. Approach (i) first slows down training performance due to more operations requiring to be carried out on individual data points, rather than a mini-batch. Second, due to the weak signal-to-noise ratio when adding noise to individual gradients, model performance is likely to degrade. Finally, sampling cannot be performed with Poisson anymore since with Poisson sampling,

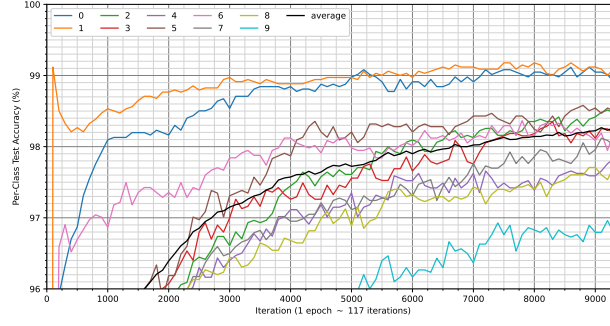
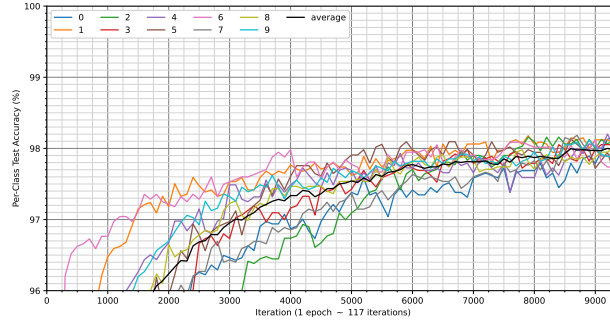
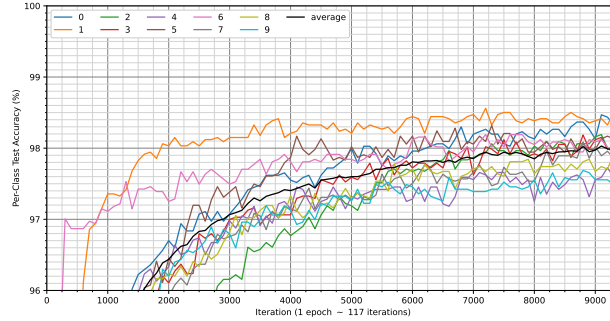
21

(a) Baseline

(b) Sample

(c) Scale

Figure 7: **Per-class test accuracies over CIFAR10 training with Per-Class Privacy Budgets.** We manually tune the per-class privacy budgets for Sample to obtain the same per-class accuracy at the end of training, see Table 9. Comparison with the Baseline (a) where all classes obtain $\varepsilon = 3$ highlight that Sample (b) and Scale (c) successfully reduce the accuracy gap between the different classes.

it is not possible to pre-determine and specify exact mini-batch sizes, instead these depend on the outcome of the random sampling process. Approach (ii) could overcome the first two issues. However, the different groups sizes are still strictly smaller than the entire dataset and large parts of DP-SGD's degrading the tight privacy bounds obtained by privacy amplification through subsampling.[6] The privacy amplification through subsampling allows to scale down the noise σ by the factor B/N (with

---

[6]Note that there exist other, less popular approaches to implement DP in ML than the DP-SGD algorithm, such as Differentially Private Follow-the-Regularized-Leader (DP-FTRL) which do not rely on subsampling but instead obtain tighter privacy bounds from adding correlated noise over the training iterations. However, since DP-FTRL under-performs DP-SGD for high-privacy regimes, and unfolds it advantages mainly in FL scenarios

22

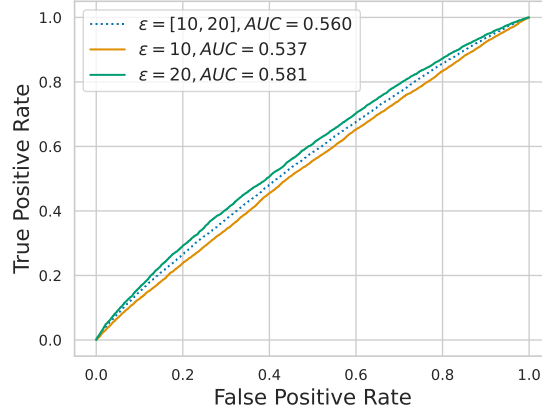
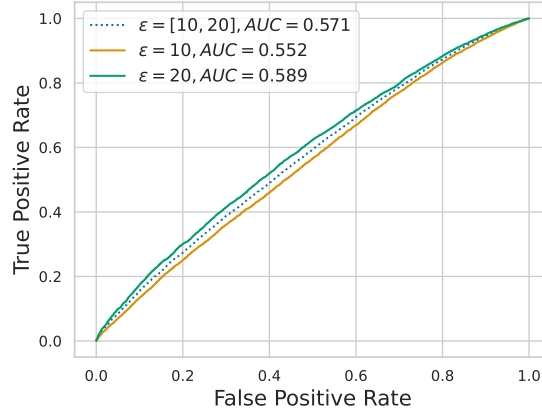



Figure 8: **True-Positive rate vs. False-Positive Rate of Lira Membership Inference Attacks Per Privacy-Budget.** We follow the same setup as in Figure 2. We show the single target model for both (a) Sample and (b) Scale methods.

B being the expected mini-batch size, N the total number of data points, and $B \ll N$) while still ensuring the same ε as with σ [16]. This privacy amplification is crucial to the practical performance (privacy-utility trade-offs) of DP-SGD. Hence, by using the P privacy groups of sizes $\{N_1, \ldots, N_P\}$ with $N_i \ll N$, the factors $B/N_i \ll B/N$ for all $i\{1, \ldots, P\}$. This effect cancels out, or in the worst case even inverts the privacy-utility benefits that should arise from assigning individual data points less noise based on their privacy preference in our individualization.

F.2   Duplicating Data Points

When duplicating data points in the training dataset, similar to the Upsampling mechanism in IPATE [5], the DP-SGD algorithm itself does not need to be adapted. Instead, different privacy levels of different data points stem from their individual number of replication within the training data. This approach offers a very fine-grained control on individual privacy levels, since, in principle, each data point could be replicated a different number of times. Utility gain would result from the larger training dataset. However, this type of upsampling opens the possibility for the same data point to be present multiple times in the mini-batch used for training in a given iteration. This stands in contrast

---

where the same data is only learned from once, or with a small number of epochs, we consider the approach outside of the scope of this work.

23

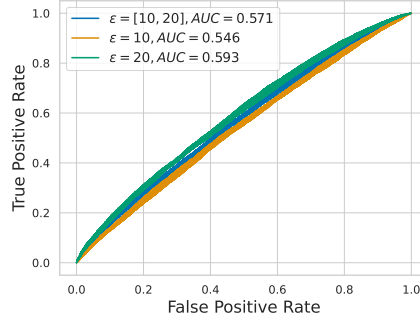



Figure 9: **True-Positive rate vs. False-Positive Rate of Lira Membership Inference Attacks Per Privacy-Budget.** We follow the same setup as in Figure 2. We run the experiment for five different target models and aggregate the results with the error bars for the Sample method.

Table 11: **Comparison between IDP-SGD and Individualized PATE (IPATE).** We select the weighting mechanism from IPATE that performs better than the upsampling method. The *Setup* indicates the size (in %) of privacy groups. We present the accuracy (%) for training with $\varepsilon = \{1.0, 2.0, 3.0\}$ privacy budgets, respectively to the order of the privacy groups.

DATASET	SETUP	BASELINE PATE	IPATE	SAMPLE	SCALE
MNIST	34%-43%-23%	91.17± 1.25	95.27 ± 0.33	**97.81**	97.78
	54%-37%-9%		95.74 ± 0.43	**97.6**	97.54
SVHN	34%-43%-23%	22.46 ± 5.19	41.45 ± 1.69	**84.56**	84.48
	54%-37%-9%		44.64 ± 0.55	**84.32**	84.31
CIFAR10	34%-43%-23%	24.83 ± 1.56	33.20 ± 0.94	54.89	**54.92**
	54%-37%-9%		35.59 ± 0.73	54.88	**55.00**

Table 12: **Empirical evaluation against other baselines.** We report the obtained test accuracy obtained with our two methods vs. two other baselines for individualized privacy on the MNIST dataset. Similar to Table 7, we use $\varepsilon = 1, 2, 3$. Both our Sample and Scale outperform the other baselines.

Setup	DP-SGD	E.1 Baseline	E.2 Baseline	Sample	Scale
34%-43%-23%	96.75	97.6	97.4	97.81	97.78
54%-37%-9%	96.75	97.1	97.3	97.6	97.54

to the original DP-SGD, where participation of each data point for training at a given iteration is determined by an independent Bernoulli trial, and hence, a data point can be either included once or not at all in a mini-batch. The possibility for a data point to be included multiple times n inside the same mini-batch changes the sensitivity of the mechanism from c to nc. According to [20], when noise is added according to σ, a mechanism with sensitivity nc is $(\alpha, \frac{2(nc)^2}{2\sigma^2})$-RDP. The quadratic influence of the sensitivity to privacy bound results in a severe increase in the RDP ε, making the approach suboptimal in terms of privacy-utility guarantees. Additionally, upsampling leads to an effective increase in a data point's the sample-rate which further increases privacy costs.

G   Additional Proofs

G.1   Additional Proofs for Individualized Privacy

Proof for Theorem 3.1

Proof. First note that (ε_1, δ)-DP can be considered as a special case of $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP, where $\forall p \in [1, P] \ \varepsilon_p = \varepsilon_1$. We can, hence apply Equation (3) and see that an M that satisfies (ε_1, δ)-

24

748 DP has a privacy guarantee of $\mathbb{P}\left[M(D) \in R\right] \le e^{\varepsilon_1} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta$. Given that by our definition
749 $\forall p \in [2, P]$ it holds that $\varepsilon_p > \varepsilon_1$, for all p, it holds that $\mathbb{P}\left[M(D) \in R\right] \le e^{\varepsilon_1} \cdot \mathbb{P}\left[M(D') \in R\right] \le$
750 $e^{\varepsilon_p} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta$. From this inequality, it follows that M also satisfies $(\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_P\}, \delta)$-
751 DP. $\qquad\square$

Proof for Theorem 3.2

753 *Proof.* Analogous to the previous proof, by our definition, it holds that $\forall p \in [1, P-1]$ the
754 $\varepsilon_p < \varepsilon_P$. From an M that satisfies $(\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_P\}, \delta)$-DP, it, therefore, holds that for all p
755 the $\mathbb{P}\left[M(D) \in R\right] \le e^{\varepsilon_p} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta \le e^{\varepsilon_P} \cdot \mathbb{P}\left[M(D') \in R\right] + \delta$. This inequality shows
756 that M satisfies (ε_P, δ)-DP. $\qquad\square$

G.2 Privacy Proofs for our Methods

758 Either of our methods can be considered as an SGM with the difference that it has different parameters
759 from the point of view of each privacy group. This is because for each group, we have to examine
760 neighboring datasets which differ in an arbitrary data point from that group. Our **Sample** method
761 ensures an individual sample rate for all points of each group, while our **Scale** method applies an
762 individual noise multiplier for all points of each group.

763 **Theorem G.1.** *Our **Sample** mechanism satisfies* $(\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_P\}, \delta)$*-DP.*

764 *Proof.* We prove the bound for any particular privacy group separately. Fix $p \in \{1, \dots, P\}$, let
765 $D \subseteq \mathcal{D}$ be the training dataset, and select any $x_i \in D$ that belongs to group $\mathcal{G}_p$. We are interested
766 in comparing outcomes of mechanism M on D with its outcomes on $D' = D \setminus \{x_i\}$ where M
767 represents a particular model update of **Sample**. We get Gaussian mixtures

$$
\begin{aligned}
M\left(D'\right) &= \sum_{L \subset D} \pi_L \mathcal{N}\left(f\left(L\right), \sigma_{\text{sample}}^2 \mathbf{I}^d\right) && \text{and} \\
M\left(D\right) &= \sum_{L \subset D} \pi_L \left((1 - q_p) \mathcal{N}\left(f\left(L\right), \sigma_{\text{sample}}^2 \mathbf{I}^d\right) + q_p \mathcal{N}\left(f\left(L \cup \{x_i\}\right), \sigma_{\text{sample}}^2 \mathbf{I}^d\right)\right) &&,
\end{aligned}
$$

768 where $f\left(L\right)$ is the clipped gradient of the current mini-batch L, π_L is its probability, $\sigma_{\text{sample}} > 0$ is
769 the noise scale, $\mathbf{I}^d$ is the identity matrix, and $0 < q_p \le 1$ is the individual sample rate of x_i and every
770 other point in $\mathcal{G}_p$. Note that σ_{sample} is actually multiplied by the clip norm $c_{\text{sample}} > 0$. Gaussian
771 mechanisms are invariant regarding scaling of their sensitivity and noise scale, but instead depend on
772 the relationship between sensitivity and noise scale, called the noise multiplier. Hence, we can ignore
773 c_{sample} and consider f to have sensitivity 1.

774 Now we can see that the Gaussian mixtures of our **Sample** are equivalent to those corresponding
775 to the original SGM from Mironov et al. [21], Thm. 4, when we parameterize it with sample
776 rate q_p and noise scale σ_{sample} which are individual per group. Therefore, all RDP bounds of the
777 original SGD apply, especially $(\alpha, \bar{\varepsilon}_p)$-RDP with $\bar{\varepsilon}_p = 2q_p^2 \frac{\alpha}{\sigma_{\text{sample}}^2}$ in a particular parameter regime
778 (cfg. Thm. 11 from Mironov et al. [21]). As a final step of the proof, we need to convert from
779 $(\alpha, \bar{\varepsilon}_p)$-RDP guarantees to (ε_p, δ)-DP guarantees following Mironov [20] (see Section 2.1). Note that
780 our individual parameters have been selected before the start of training so that each group's privacy
781 budget is exhausted at the intended number of iterations (see Algorithm 1). $\qquad\square$

782 **Theorem G.2.** *Our **Scale** mechanism satisfies* $(\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_P\}, \delta)$*-DP.*

783 *Proof.* This proof is analog to the proof of Theorem G.1 with the difference that we have a global
784 sample rate q but individual noise multipliers σ_p and clip norms c_p. Moreover, Algorithm 2 is used to
785 configure parameters prior to training. $\qquad\square$

25