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 $M \colon \mathcal{D}^* \to \mathcal{R}$ 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.

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.

Initialize θ_0 randomly
for $t \in [I]$ **do**
  Poisson Sampling: Sample mini-batch L_t with per-point probability q from $\mathcal{D}$.
  For each $i \in L_t$, compute $g_t(x_i) \leftarrow \nabla_{\theta_t} l(\theta_t, x_i)$
  1. Gradient Clipping
  $\bar{g}_t(x_i) \leftarrow g_t(x_i) / \max\left(1, \frac{\|g_t(x_i)\|_2}{c}\right)$
  2. Noise Addition
  $\tilde{g}_t \leftarrow \frac{1}{|L_t|}\left(\sum_i \bar{g}_t(x_i) + \mathcal{N}(0, (\sigma c)^2 \mathbf{I})\right)$
  $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
end for
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

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 $\Lambda = \frac{p(f(x)_y \mid \mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}}))}{p(f(x)_y \mid \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}}))}$.

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.

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

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

init $\varepsilon_{\text{high}}$: $\varepsilon_{\text{high}} \leftarrow \infty$
init $q_{\text{low}}, q_{\text{high}}$: $q_{\text{low}} \leftarrow 1\text{e}{-9}, q_{\text{high}} \leftarrow 0.1$
while $\varepsilon_{\text{high}} > \varepsilon$ **do**
  $q_{\text{high}} \leftarrow 2q_{\text{high}}$
  $\varepsilon_{\text{high}} \leftarrow I \cdot 2q_{\text{high}}^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
end while
while $\varepsilon - \varepsilon_{\text{high}} > \gamma$ **do**
  $q \leftarrow (q_{\text{low}} + q_{\text{high}})/2$
  $\varepsilon_{\text{temp}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
  if $\varepsilon_{\text{temp}} < \varepsilon$ **then**
    $q_{\text{high}} \leftarrow q$
    $\varepsilon_{\text{high}} \leftarrow \varepsilon_{\text{temp}}$
  else
    $q_{\text{low}} \leftarrow q$
  end if
end while
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 $\gamma = 0.01$

init $\varepsilon_{\text{high}}$: $\varepsilon_{\text{high}} \leftarrow \infty$
init $\sigma_{\text{low}}, \sigma_{\text{high}}$: $\sigma_{\text{low}} \leftarrow 0, \sigma_{\text{high}} \leftarrow 10$
while $\varepsilon_{\text{high}} > \varepsilon$ **do**
  $\sigma_{\text{high}} \leftarrow 2\sigma_{\text{high}}$
  $\varepsilon_{\text{high}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma_{\text{high}}^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
end while
while $\varepsilon - \varepsilon_{\text{high}} > \gamma$ **do**
  $\sigma \leftarrow (\sigma_{\text{low}} + \sigma_{\text{high}})/2$
  $\varepsilon_{\text{temp}} \leftarrow I \cdot 2q^2 \frac{1}{\sigma^2}$ {approximate epsilon according to Equation (2), we suppress α for simplicity}
  if $\varepsilon_{\text{temp}} < \varepsilon$ **then**
    $\sigma_{\text{high}} \leftarrow \sigma$
    $\varepsilon_{\text{high}} \leftarrow \varepsilon_{\text{temp}}$
  else
    $\sigma_{\text{low}} \leftarrow \sigma$
  end if
end while
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.

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 $\varepsilon = \{1.0 - 2.0 - 3.0\}$

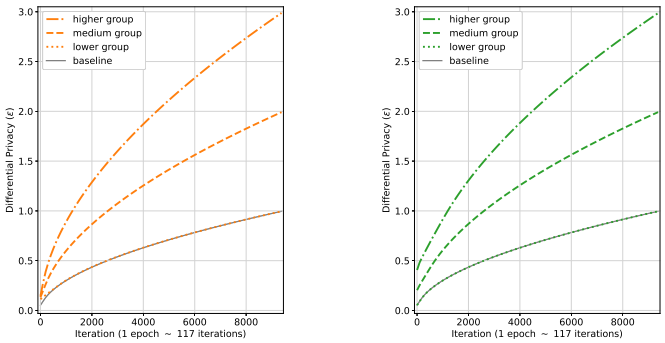
		DP-SGD			SCALE			SAMPLE	
DATASET	SETUP	σ	c	q	σ_{SCALE}	$\{\sigma_1, \ldots, \sigma_P\}$	$\{c_1, \ldots, c_P\}$	σ_{SAMPLE}	$\{q_1, \ldots, q_P\}$
MNIST	*34%-43%-23%*	3.425	0.2	0.008	2.063	{2.189, 1.310, 1.032}	{0.129, 0.216, 0.274}	2.024	{0.005, 0.009, 0.013}
	54%-37%-9%	3.425	0.2	0.008	2.418	{2.189, 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.

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 $\boldsymbol{\varepsilon}$ 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	$\boldsymbol{D}$	*34%-43%-23%*	96.75	**97.81**	97.78
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$ 0.15	$\pm$ **0.09**	$\pm$ 0.08
	$\boldsymbol{D}$	*54%-37%-9%*	96.75	**97.6**	97.54
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$ 0.15	$\pm$ **0.11**	0.09
SVHN	$\boldsymbol{D}$	*34%-43%-23%*	83.26	**84.56**	84.48
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$0.31	$\pm$**0.25**	$\pm$0.25
	$\boldsymbol{D}$	*54%-37%-9%*	83.26	**84.32**	84.31
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$0.31	$\pm$**0.31**	$\pm$0.26
CIFAR10	$\boldsymbol{D}$	*34%-43%-23%*	52.77	54.89	**54.92**
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$ 0.65	$\pm$ 0.55	$\pm$**0.63**
	$\boldsymbol{D}$	*54%-37%-9%*	52.77	54.88	**55.00**
	$\boldsymbol{\varepsilon}$	*1.0-2.0-3.0*	$\pm$ 0.65	$\pm$ 0.45	$\pm$ **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-

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 ($\varepsilon = 2$).

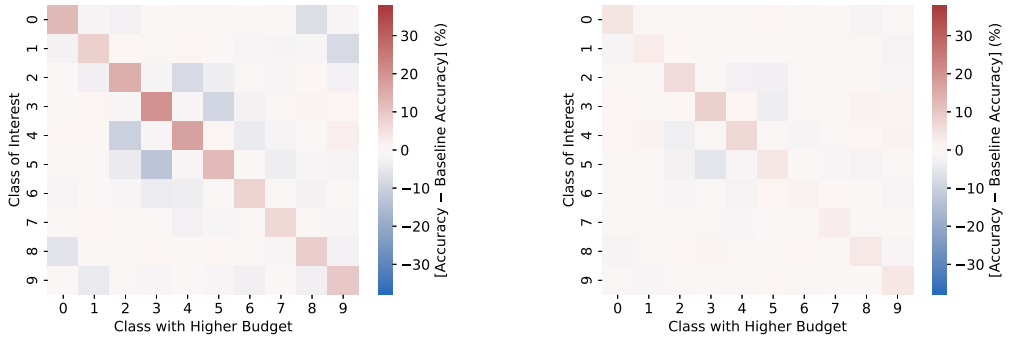
Classes	Privacy Budget		
	Higher ($\varepsilon = 3$)	Same ($\varepsilon = 2$)	Lower ($\varepsilon = 1$)
0	**67.76 ± 2.14**	55.24 ± 1.98	25.76 ± 2.52
1-9	53.41 ± 2.2	54.72 ± 2.49	**56.58 ± 2.29**
1	**80.84 ± 1.2**	72.79 ± 1.6	46.09 ± 3.91
0,2-9	51.65 ± 2.28	52.77 ± 2.53	**54.59 ± 2.23**
2	51.31 ± 3.53	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

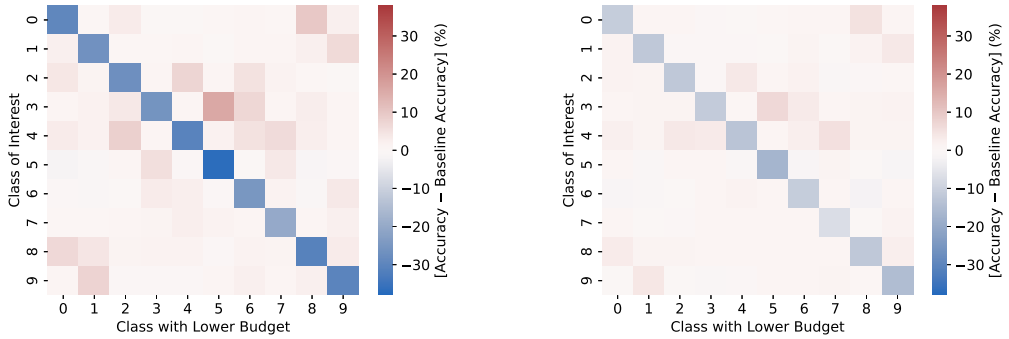
Membership inference success for a single target model of our Sample and Scale methods is shown in Figure 8. In Figure 9, we present the results over for 5 different target models for Sample. The figure highlights that over all target models, the two privacy groups' privacy risk is different: the group 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.



(a) Sample (higher budget) (b) Scale (higher budget)



(c) Sample (lower budget) (d) Scale (lower budget)

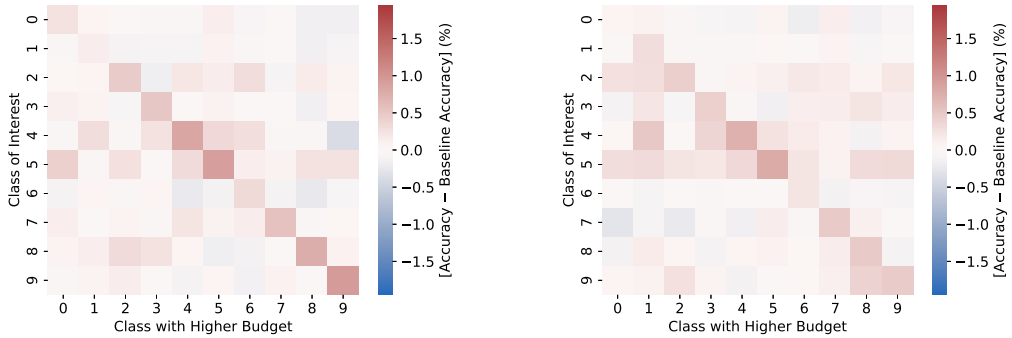
Figure 5: **CIFAR10: Accuracy Changes for Subgroups.** We assess how the test **Accuracy** of a **Class of interest** changes in comparison to the **Baseline Accuracy** (standard DP-SGD with $\varepsilon = 2$) when we, during training, assign a lower ($\varepsilon = 1$) or a higher ($\varepsilon = 3$) 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.

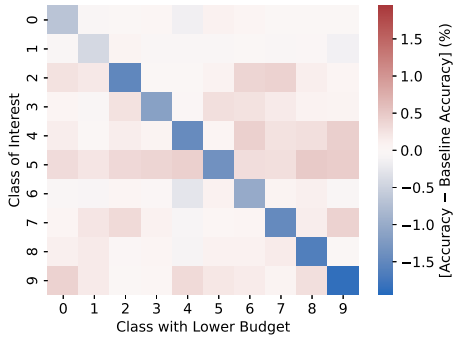
E Alternative Baselines

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.

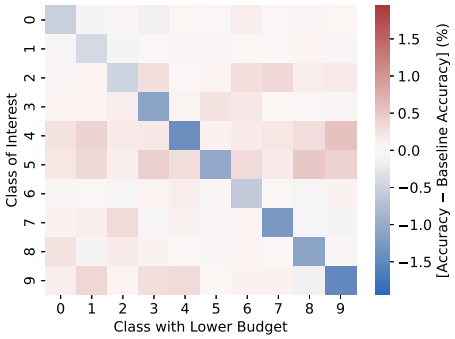


(a) Sample | higher budget

(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.

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.

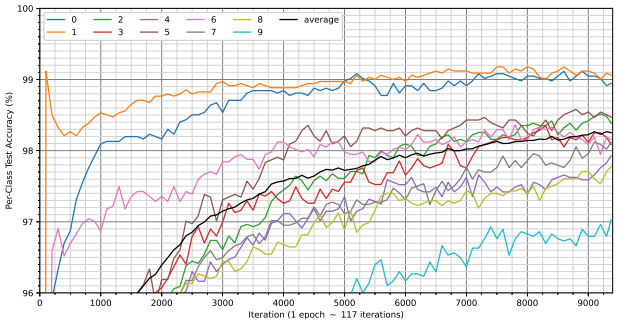
In summary, we observe that our methods outperform the other baselines.

F Alternative Individualization

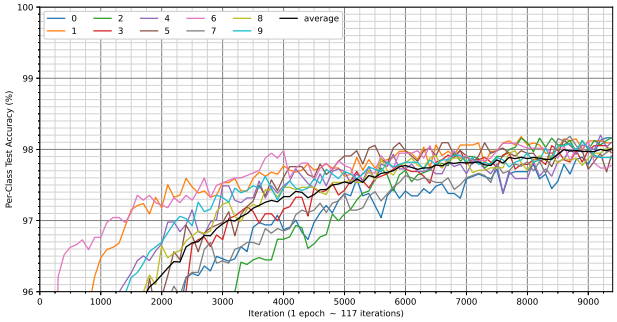
We present the alternative ways of individualizing privacy guarantees in DP-SGD that we considered in the design process of IDP-SGD and describe their drawbacks.

F.1 Individual Per-Data Point Noise

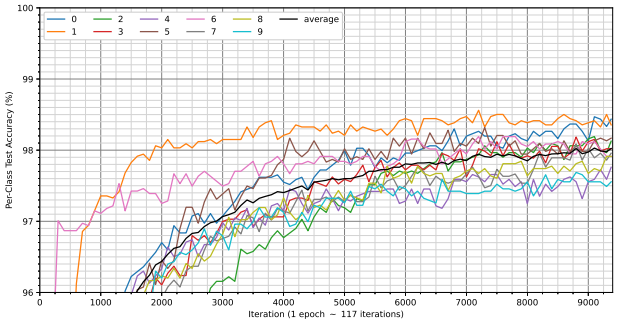
Individualized privacy could, in principle also be obtained by adding different amounts of noise to different data points. Every of the P privacy group would have their individual $\{\sigma_1, \ldots, \sigma_P\}$. Utility improvements would result from some data points requiring smaller amounts of added 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,



(a) Baseline



(b) Sample

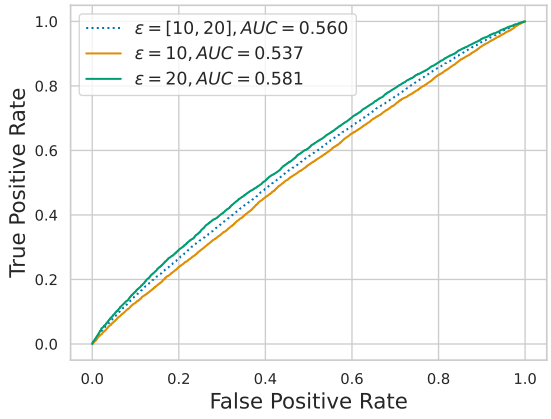


(c) Scale

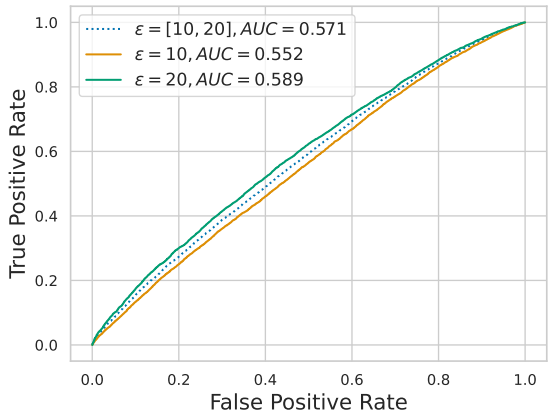
Figure 7: **Per-class test accuracies over CIFAR10 training with Per-Class Privacy Budgets.** We manually tune the per-class privacy budgets for Sample to obtain the same per-class accuracy at the end of training, see Table 9. Comparison with the Baseline (a) where all classes obtain $\varepsilon = 3$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



(a) Sample



(b) Scale

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.

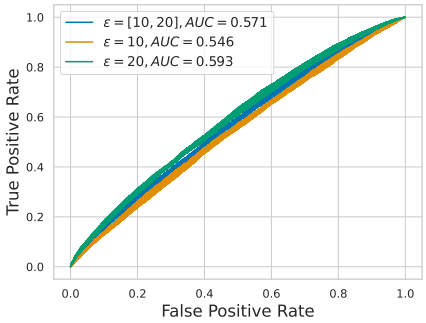


Figure 9: **True-Positive rate vs. False-Positive Rate of Lira Membership Inference Attacks Per Privacy-Budget.** We follow the same setup as in Figure 2. We run the experiment for five different target models and aggregate the results with the error bars for the Sample method.

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)-

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

Proof for Theorem 3.2

Proof. Analogous to the previous proof, by our definition, it holds that $\forall p \in [1, P-1]$ the $\varepsilon_p < \varepsilon_P$. From an M that satisfies $(\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}, \delta)$-DP, it, therefore, holds that for all p the $\mathbb{P}[M(D) \in R] \leq e^{\varepsilon_p} \cdot \mathbb{P}[M(D') \in R] + \delta \leq e^{\varepsilon_P} \cdot \mathbb{P}[M(D') \in R] + \delta$. This inequality shows that M satisfies (ε_P, δ)-DP. □

G.2 Privacy Proofs for our Methods

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

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

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

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

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

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

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

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