

Masterarbeit am Institut für Informatik der Freien Universität Berlin,  
Arbeitsgruppe ID Management

# Personalizing Private Aggregation of Teacher Ensembles

**Christopher Mühl**

christopher.muehl@fu-berlin.de

Matrikelnummer: 4761282

Betreuerin: Franziska Boenisch

1. Gutachter: Prof. Dr. Marian Margraf

2. Gutachter: Prof. Dr. Tim Landgraf

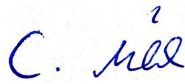
Berlin, den 16.06.2021



## Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 16.06.2021



---



# Acknowledgements

This thesis was done in cooperation with the Fraunhofer AISEC Institute.

My sincerest gratitude is designated to Franziska Boenisch for her excellent supervision and to Jannis Ihrig for many fruitful discussions. Thank you.



# Abstract

Due to the increasing number of applications for machine learning (ML) and the accompanying usage of sensitive data in the past years, privacy preservation is of high importance. Differential privacy (DP) has established itself as the most popular privacy definition in recent research besides many others. It enables worst-case privacy guarantees for all possible data at the same time. In practice, not all data require the same amount of privacy. Therefore, personalized DP which provides individual guarantees was proposed.

One state-of-the-art approach to preserve and to quantify DP for ML applications is the private aggregation of teacher ensembles (PATE). In a voting process, an ensemble of arbitrary ML models that was trained on partitions of sensitive data produces labels for a public unlabeled dataset. The DP of the sensitive data is measured during the votings. Afterwards, a target ML model is trained on the produced labels and the public data. Since the target model does not know any sensitive data, the privacy preservation is intuitively understandable.

In this thesis, three different extensions of the PATE approach that enable personalized DP are proposed. Depending on the privacy personalization, these approaches can reduce the privacy costs of data with high privacy preferences significantly by increasing the costs of data with lower preferences. Hence, more utility can be acquired and donors of sensitive data have the option to determine individual privacy preferences. Furthermore, it can be shown that data augmentation may improve the utility of teacher ensembles without increasing their privacy expenditure. Both improvements of PATE, namely personalization and data augmentation, enable more practical applications of privacy-preserving ML.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background and Notation</b>	<b>3</b>
2.1	Machine Learning . . . . .	3
2.2	Differential Privacy . . . . .	5
2.2.1	Rényi Differential Privacy . . . . .	6
2.2.2	Personalized Differential Privacy . . . . .	9
<b>3</b>	<b>Related Work</b>	<b>11</b>
3.1	Alternative Individual Privacy Definitions . . . . .	11
3.1.1	Heterogeneous Differential Privacy . . . . .	11
3.1.2	Per-Instance Differential Privacy . . . . .	12
3.2	Mechanisms to Meet PDP . . . . .	12
3.2.1	Stretching Mechanism . . . . .	12
3.2.2	Sample Mechanism . . . . .	13
3.2.3	Personalized Exponential Mechanism . . . . .	14
3.2.4	Partitioning Mechanisms . . . . .	15
3.2.5	Rényi Privacy Filter . . . . .	17
3.2.6	Personalized Moments Accounting . . . . .	19
3.3	Specific Applications of PDP . . . . .	21
3.4	Other Approaches to Individualized DP . . . . .	22
<b>4</b>	<b>Private Aggregation of Teacher Ensembles</b>	<b>23</b>
4.1	Privacy Guarantee of PATE . . . . .	25
4.2	Improved PATE . . . . .	26
4.2.1	Gaussian Mechanism . . . . .	28
4.2.2	Privacy Guarantee of the Improved PATE . . . . .	28
<b>5</b>	<b>Personalized PATE Variants</b>	<b>31</b>
5.1	Personalization Techniques for PATE . . . . .	31
5.2	Upsampling . . . . .	32
5.3	Vanishing . . . . .	33
5.4	Weighting . . . . .	35
5.5	Privacy Guarantee of Personalized PATE . . . . .	36

<b>6</b>	<b>Evaluation</b>	<b>39</b>
6.1	Datasets . . . . .	39
6.1.1	Handwritten Digits Dataset . . . . .	39
6.1.2	Census Income Dataset . . . . .	41
6.2	Experiments . . . . .	42
6.2.1	Utility . . . . .	43
6.2.2	The Advantage of Personalization . . . . .	44
<b>7</b>	<b>Discussion</b>	<b>51</b>
7.1	Assessment of the Results . . . . .	51
7.2	Comparison of the Personalized Variants . . . . .	52
7.3	Future Work . . . . .	52
<b>8</b>	<b>Conclusion</b>	<b>55</b>

# 1 Introduction

Nowadays, the amount of digital data and the number of techniques to generate value from these data are high and keep increasing. Such techniques can be simple summary statistics or aggregate functions, as well as complex *machine learning* (ML) algorithms. As a consequence, the need for privacy protection of data rises, especially of those with sensitive information, e.g. medical data. In the privacy research, more than 80 different privacy metrics [38] have been investigated whereof a prominent group of metrics is *differential privacy* (DP) [9]. The DP of a particular data point corresponds to the difference between the aggregate information about a dataset that would be published once with and once without that particular data point included. For example, consider the body weights of a group of people as the data of interest. These data are processed by calculating the average weight which is then published. Although the influence of particular persons on the average weight is small, their exact body weight can be determined if all other weights are known. Thus, a person that is not intended to know a particular body weight could ask the other participants for their weights to infer it.

There are scenarios different from the above example, where the potential damage of unintended knowledge could be more dramatic, e.g. with medical data. In order to protect each data point's privacy, a random value is added to the processed information before being published. Hence, even with knowledge about all but one data points, it is not possible to infer the missing value. Instead, it can only be estimated imprecisely. The precision of estimates depends on the amount of knowledge about other data points, and the random distribution from which the random value is sampled. The random distribution is the set of possible values combined with their probabilities. The higher the *variance*, i.e. the expected squared difference of a random value from the average random value, the less information about each data point can be discovered. Consequently the privacy of data points is higher.

The objective of data processing is to gain insights into specific properties of that data and transfer these insights to similar data, i.e. utility. Unfortunately, there is a trade-off between privacy and utility [44], i.e. the higher the variance of randomness used to induce privacy the less accurate is the processing result. DP guarantees that all sensitive data points do not exceed a certain privacy loss, i.e. an amount of released information, by being processed. In practice, not every data point requires the same amount of privacy. In order to maximize utility, each data point should

## 1 Introduction

spend that much information s.t. its personal privacy requirement is not violated. In other words, instead of homogeneous privacy, a heterogeneous, i.e. personalized privacy definition is required.

In the past six years, some research was conducted on the topic of personalizing DP. Similar definitions that relate privacy costs to particular data points were proposed. Most techniques to achieve such personalized privacy can not be applied directly to ML. Instead, simple processing mechanisms to analyze databases are targeted. However, two approaches enable *privacy-preserving ML* (PPML). Both extend popular techniques to induce and account privacy in the gradient descent of supervised ML. The personalized accounting increases the already high time-expenses even more. A popular alternative to such gradient descent-based DP mechanisms is an ensembling algorithm. Since it induces and accounts DP in the ensemble after the actual learning of sensitive data, the ensembling method comes with benefits and drawbacks compared to the other DP mechanisms for DP. One important advantage is that it enables an intuitive understanding of its privacy preservation. Unfortunately, there is no personalized variant of it so far. In order to close this gap and to enrich the research on personalized PPML, this thesis proposes three techniques to enable personalization for the ensembling mechanism.

The main contributions of this work are

- the development of three techniques to enable privacy personalization,
- the proposal to apply data augmentation within the approach to decrease privacy expenditure,
- and the observation that more uniformity of privacy personalization of sensitive data leads to a more efficient use of privacy in favor of utility.

The remainder of the work at hand is organized as follows. The essential background and notations are provided in Chapter 2. The subsequent chapter investigates extensions of DP and approaches to achieve them. Chapter 4 describes the main PPML approach to adapt. The elaborated personalized extensions are described in Chapter 5 which are evaluated afterwards in Chapter 6. Finally, Chapter 7 and Chapter 8 conclude this work.

## 2 Background and Notation

This chapter describes and defines the fundamental notions of this work, namely ML (Section 2.1), and DP (Section 2.2). Simultaneously, the main notations are provided.

### 2.1 Machine Learning

In general, ML is often associated with the approximation of a function that underlies a data distribution of interest. It is usually done by examining data points from that distribution. Three major categories of ML—namely, *supervised*, *unsupervised*, and *reinforcement* learning can be distinguished. Reinforcement learning aims to find good strategies in specific environments, e.g. a strong chess playing intelligence, and exceeds the scope of this thesis. The former category is of interest in this work. In supervised learning, each data point is assessed by a value that characterizes it. This value can either correspond to a discrete group of data points or to a position/rank within a continuous order of data points. In the former case, the groups are called classes and the value of each data point mapping it to a class is called *label*. A common goal of ML on labeled data is *classification*, i.e. the assignment of labels to unseen unlabeled data points. Analogously, a common ML task on data assessed by continuous values is called *regression*. Both, classification and regression can be summarized as *prediction*. Contrary to prediction, data can also be used for *generation* tasks, i.e. the artificial generation of so-called *synthetic* data, both in a supervised or in an unsupervised manner. In contrast to supervised learning and generation, unsupervised learning aims to find structure inherent in data that were not grouped or ordered a priori. In between supervised and unsupervised learning, there is another category called *semi-supervised* learning where some data are assigned by values while others are not.

The most important ML task in this work is classification. Its fundamentals can be described as follows. Let  $\mathcal{X}$  be any set that defines the form of all possible data points/feature samples of interest.  $\mathcal{X}$  is called *feature space* and it usually equals a discretization of  $\mathbb{R}^n$  with  $n \gg 1$  being the dimension of all data points. The feature space is partitioned so that all samples within the same partition have a specific commonality. The commonality is determined by the class that corresponds to the

## 2 Background and Notation

partition. The set of all classes  $\mathcal{Y}$  called *label space* is finite and is usually expressed by integers. Actual data in  $\mathcal{X}$  have a certain distribution that is described by a random variable  $X$ . Each feature sample  $x \in \mathcal{X}$  can be definitely assigned to a label  $y \in \mathcal{Y}$ . Hence, the labels have a distribution as well that can be described by a random variable  $Y$ . Since  $Y$  depends completely on  $X$ , there exists a function  $h: \mathcal{X} \rightarrow \mathcal{Y}$  with  $X \mapsto^h Y$ . The assignment of feature samples  $x \in \mathcal{X}$  to classes  $y \in \mathcal{Y}$  is referred to as classification.

**Example 2.1.** Consider  $\mathcal{X}$  to be the set of all possible  $1000 \times 1000$ -pixel RGB images s.t.  $\mathcal{X} = \{0, \dots, 255\}^{1000 \cdot 1000 \cdot 3}$ . Here,  $\{0, \dots, 255\}$  is the discretization of the intensity of one color channel in a pixel.  $X$  could be the distribution of images each of a cat or a dog with classes '0' for 'cat' and '1' for 'dog'. The function  $f$  is not defined explicitly, but humans and especially veterinaries are able to emulate  $h$  by determining the correct label  $y := h(x)$  for a given image  $x$  very accurately.

**Definition 2.1** (ML Model & Classifier). Let  $\mathcal{X}$  be a feature space and  $\mathcal{Y}$  be any finite or infinite set. A computable function  $\hat{h}_\theta: \mathcal{X} \rightarrow \mathcal{Y}$  that is based on a set of adjustable parameters  $\theta \in \mathbb{R}^n$  of any dimension  $n$  is considered an *ML model*. In case of  $\mathcal{Y}$  to be discrete classes as described above,  $\hat{h}_\theta$  is called a *classifier*.

**Definition 2.2** (Classifier's Prediction). Let  $m := |\mathcal{Y}|$  be the number of classes. Usually,  $\hat{h}_\theta$  computes a *confidence vector*  $\mathbf{p} := (p_1, \dots, p_m)^T := g_\theta(x)$  where  $g_\theta: \mathcal{X} \rightarrow [0, 1]^m$  is a function inherent in  $\hat{h}_\theta$ . Its values can be considered as probabilities corresponding to all classes s.t.  $p_y$  is the probability that  $x$  belongs to class  $y$ . The prediction  $\hat{y}$  of a classifier can be defined as

$$\hat{y} := \hat{h}_\theta(x) := \arg \max_y \{p_y\} . \quad (2.1)$$

*Remark.* A good classifier is similar to the true underlying function  $h$ . To achieve this, the classifier's inherent parameters  $\theta$  are adjusted in a so-called *training* process using a labeled dataset

$$D := \{(x_0, y_0), \dots, (x_N, y_N)\} \subseteq \mathcal{X} \times \mathcal{Y} =: \mathcal{D} \quad (2.2)$$

of length  $N$ .  $D$  is collected by repeatedly sampling from the distribution  $(X, Y)$ . In the case of unlabeled data, in this work a dataset is defined as

$$D := \{x_0, \dots, x_N\} \subseteq \mathcal{X} =: \mathcal{D} . \quad (2.3)$$

The classifier  $\hat{h}_\theta$  is trained until  $\hat{h}_\theta \approx f$ . The *accuracy* of  $\hat{h}_\theta$  regarding  $h$  can be empirically measured by the ratio of the frequency of agreements between  $\hat{h}_\theta$  and  $h$  according to their predictions on a set of data points to the total number of these data points.

This section is just a short and relatively imprecise introduction to a few important concepts of ML which are essential to this thesis. For a more detailed and comprehensive explanation, the interested reader is referred to [3, 15]. In the following section, another essential concept is described.

## 2.2 Differential Privacy

DP is defined in a way that it assigns one value ( $\epsilon$ -DP [9], Def. 1) or two values  $((\epsilon, \delta)$ -DP [11], Def. 2.4) to a data-processing algorithm—called *mechanism*—to express the point-wise maximal loss of privacy.

**Definition 2.3** (Neighboring Datasets (cf. [19], Def. 2)). Let  $\mathcal{D}$  and  $\mathcal{R}$  be the sets of all possible data points and all possible processing results of a mechanism on any dataset, respectively. Two datasets  $D, D' \subseteq \mathcal{D}$  are called *neighboring*, denoted by  $D \sim D'$ , if there exists exactly one data point  $d \in \mathcal{D}$  that is only included in  $D$  but not in  $D'$  or vice versa. In the former case it can be written  $D = D'_{+d}$  or  $D' = D_{-d}$ , and in the latter case  $D' = D_{+d}$  or  $D = D'_{-d}$ . In both cases, the notation  $D \stackrel{d}{\sim} D'$  can be used to point out that  $d$  is included in only one of them.

However, neighboring datasets were originally considered to have the same size s.t. both datasets share all elements except for one that is not included in the other set ([9], cf. Def. 1). Both definitions are used alternatively depending on what fits the interest best.

**Definition 2.4**  $((\epsilon, \delta)$ -Differential Privacy). Let  $\mathcal{D}$  and  $\mathcal{R}$  be the sets of all possible data points and all possible processing results of a mechanism on any dataset, respectively. A mechanism  $M: \mathcal{D}^* \rightarrow \mathcal{R}$  meets  $(\epsilon, \delta)$ -DP ([11], Def. 2.4) with  $\epsilon, \delta \in \mathbb{R}_+$  if for all neighboring datasets  $D, D' \subseteq \mathcal{D}$  and all result events  $R \subseteq \mathcal{R}$

$$\Pr [M(D) \in R] \leq e^\epsilon \cdot \Pr [M(D') \in R] + \delta. \quad (2.4)$$

*Remark.* In this work,  $\Pr[\cdot]$  denotes probability measures that correspond to probability distributions of specific random variables. Note that  $\mathcal{D}$  and  $\mathcal{R}$  can be arbitrary sets. Moreover, if  $\delta = 0$ ,  $M$  satisfies  $\epsilon$ -DP.

One important property of  $(\epsilon, \delta)$ -DP is its *composability*, meaning the way of combining the privacy costs of publishing the results of multiple mechanisms that processed the same dataset.

## 2 Background and Notation

**Proposition 2.1** (Composition of DP (cf. [11], Thm. 3.16)). *Let  $M_1: \mathcal{D}^* \rightarrow \mathcal{R}_1$ , and  $M_2: \mathcal{D}^* \rightarrow \mathcal{R}_2$  be two mechanisms that satisfy  $(\varepsilon_1, \delta_1)$ - and  $(\varepsilon_2, \delta_2)$ -DP, respectively. Here,  $\mathcal{D}$  and  $\mathcal{R}_1, \mathcal{R}_2$  are the sets of all possible data points and all possible processing results of  $M_1$  and  $M_2$  on any dataset, respectively. Then, the composed mechanism  $M_3: \mathcal{D}^* \rightarrow \mathcal{R}_1 \times \mathcal{R}_2$  with  $M_3(D) \mapsto (M_1(D), M_2(D))$  satisfies  $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$ -DP.*

*Proof (Sketch).* Let  $r_1, r_2, r'_1, r'_2 := M_1(D), M_2(D), M_1(D'), M_2(D')$  be the results of the two mechanisms  $M_1, M_2$  on any two neighboring datasets  $D, D' \in \mathcal{D}$ . In the worst case, the information from the difference between  $r_1$  and  $r'_1$  about  $d$  is not redundant to that from the difference between  $r_2$  and  $r'_2$ . Moreover, DP always gives worst-case guarantees. Hence, the privacy costs add up.  $\square$

### 2.2.1 Rényi Differential Privacy

$\varepsilon$ -DP is a very inflexible privacy notion which is not suitable to provide good privacy bounds for some mechanisms, e.g., those that use Gaussian noise. That is the main reason for the invention of  $(\varepsilon, \delta)$ -DP. However, a natural extension of DP was later examined that enables tighter privacy bounds of such mechanisms. This subsection provides the fundamentals of that DP notion.

In order to measure the dissimilarity and hence, indirectly the similarity of probability distributions, divergences are used. One prominent example is the Kullback-Leibler (KL) divergence [22]. A generalization of the KL divergence, the *Rényi divergence* which is specified by the parameter  $\alpha \in (1, \infty)$ , can be defined as follows.

**Definition 2.5** (Rényi Divergence). Let  $P$  and  $Q$  be two probability distributions over the same arbitrary sample space  $\mathcal{X}$ . The Rényi divergence ([26], Def. 3) of order  $\alpha$ , or  $\alpha$ -divergence for short, for  $P$  and  $Q$  is defined as

$$D_\alpha(P \parallel Q) := \frac{1}{\alpha - 1} \cdot \ln \mathbb{E}_{x \sim Q} \left[ \left( \frac{P(x)}{Q(x)} \right)^\alpha \right], \quad (2.5)$$

where  $\mathbb{E}[\cdot]$  outputs the expected value of a given random variable. Note that if  $\alpha = 1$ , the limit  $\alpha \downarrow 1$  is used and thus the KL divergence is obtained.

Based on the Rényi divergence a natural extension of DP—*Rényi DP* (RDP)—was proposed in 2017 [26]. In RDP, whole probability distributions over a mechanism's results corresponding to neighboring datasets are compared instead of particular result events' probabilities as in standard DP.



**Definition 2.6** (Rényi Differential Privacy). Let  $M: \mathcal{D}^* \rightarrow \mathcal{R}$  be a mechanism where  $\mathcal{D}$  and  $\mathcal{R}$  are the sets of all possible data points and all possible processing results of  $M$ , respectively. In this thesis,  $f_{M(D)}$  is the result distribution of  $M$  on any dataset  $D \in \mathcal{D}$ .  $M$  meets  $(\alpha, \varepsilon)$ -RDP ([26], Def. 4) if for all datasets  $D \sim D'$

$$D_\alpha(f_{M(D)} \parallel f_{M(D')}) \leq \varepsilon. \quad (2.6)$$

One important advantage of RDP over  $(\varepsilon, \delta)$ -DP is its smoother composability. In contrast to  $(\varepsilon, \delta)$ -DP, only the epsilon values are summed up in RDP. Lemma 2.2 states the RDP guarantee of two arbitrarily composed DP mechanisms whose RDP bounds are known. Note that instead of explicit sequentiality, independent mechanisms can be assumed for the sake of brevity. This assumption does not affect the statement since RDP is immune to post-processing.

**Lemma 2.2** (Composition of RDP (cf. [26], Prop. 1)). *Let  $\mathcal{D}$  and  $\mathcal{R}_1, \mathcal{R}_2$  be the sets of all possible data points and all possible processing results of two mechanisms  $M_1, M_2$  on any dataset, respectively. Let further  $M_1: \mathcal{D}^* \rightarrow \mathcal{R}_1$ , and  $M_2: \mathcal{D}^* \rightarrow \mathcal{R}_2$  satisfy  $(\alpha, \varepsilon_1)$ - and  $(\alpha, \varepsilon_2)$ -RDP, respectively. Let  $f_1, f_2 := f_{M_1(D)}, f_{M_2(D)}$  and  $f'_1, f'_2 := f_{M_1(D')}, f_{M_2(D')}$  denote the probability distributions of all possible results of  $M_1, M_2$  on any neighboring datasets  $D \sim D'$ . Then, the composed mechanism  $M_3: \mathcal{D}^* \rightarrow \mathcal{R}_1 \times \mathcal{R}_2$  with  $M_3(D) \mapsto (M_1(D), M_2(D))$  satisfies  $(\alpha, \varepsilon_1 + \varepsilon_2)$ -RDP for all  $\alpha \geq 1$ .*

*Proof* (cf. [26], Proof of Prop. 1). Let  $f_3 := f_{M_3(D)} := (f_1, f_2)$  and  $f'_3 := f_{M_3(D')} := (f'_1, f'_2)$  be the probability distributions of  $M_3$  on the neighboring datasets  $D \sim D'$ ,

## 2 Background and Notation

respectively. Then,

$$D_\alpha(f_3 \parallel f'_3) \tag{2.7}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \mathbb{E}_{r_3 \sim f'_3} \left[ \left( \frac{f_3(r_3)}{f'_3(r_3)} \right)^\alpha \right] \tag{2.8}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \int_{\mathcal{R}_3} \left( \frac{f_3(r_3)}{f'_3(r_3)} \right)^\alpha \cdot f'_3(r_3) \, dr_3 \tag{2.9}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \int_{\mathcal{R}_1 \times \mathcal{R}_2} \left( \frac{(f_1(r_1), f_2(r_2))}{(f'_1(r_1), f'_2(r_2))} \right)^\alpha \cdot (f'_1(r_1), f'_2(r_2)) \, d(r_1, r_2) \tag{2.10}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \int_{\mathcal{R}_1 \times \mathcal{R}_2} \left( \frac{f_1(r_1) \cdot f_2(r_2)}{f'_1(r_1) \cdot f'_2(r_2)} \right)^\alpha \cdot f'_1(r_1) \cdot f'_2(r_2) \, d(r_1, r_2) \tag{2.11}$$

$$\leq \frac{1}{\alpha - 1} \cdot \ln \int_{\mathcal{R}_1} \left( \int_{\mathcal{R}_2} \left( \frac{f_2(r_2)}{f'_2(r_2)} \right)^\alpha \cdot f'_2(r_2) \, dr_2 \right) \cdot \left( \frac{f_1(r_1)}{f'_1(r_1)} \right)^\alpha \cdot f'_1(r_1) \, dr_1 \tag{2.12}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \left( \int_{\mathcal{R}_1} \left( \frac{f_1(r_1)}{f'_1(r_1)} \right)^\alpha \cdot f'_1(r_1) \, dr_1 \cdot \int_{\mathcal{R}_2} \left( \frac{f_2(r_2)}{f'_2(r_2)} \right)^\alpha \cdot f'_2(r_2) \, dr_2 \right) \tag{2.13}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \left( \mathbb{E}_{r_1 \sim f'_1} \left[ \left( \frac{f_1(r_1)}{f'_1(r_1)} \right)^\alpha \right] \cdot \mathbb{E}_{r_2 \sim f'_2} \left[ \left( \frac{f_2(r_2)}{f'_2(r_2)} \right)^\alpha \right] \right) \tag{2.14}$$

$$= \frac{1}{\alpha - 1} \cdot \ln \mathbb{E}_{r_1 \sim f'_1} \left[ \left( \frac{f_1(r_1)}{f'_1(r_1)} \right)^\alpha \right] + \frac{1}{\alpha - 1} \cdot \ln \mathbb{E}_{r_2 \sim f'_2} \left[ \left( \frac{f_2(r_2)}{f'_2(r_2)} \right)^\alpha \right] \tag{2.15}$$

$$= D_\alpha(f_1 \parallel f'_1) + D_\alpha(f_2 \parallel f'_2) \tag{2.16}$$

$$\leq \varepsilon_1 + \varepsilon_2 \tag{2.17}$$

proofs the claim. Note that the inequality in line (2.12) results from a possible dependence between  $r_1$  and  $r_2$ .  $\square$

The relation between RDP and  $(\varepsilon, \delta)$ -DP is described in [26], Sec. 4. Accordingly,  $\varepsilon$ -DP coincides with  $(\infty, \varepsilon)$ -RDP which in turn implies  $(\alpha, \varepsilon)$ -RDP for all  $\alpha \geq 1$  by monotonicity. The inverse relation is stated in Lemma 2.3.

**Lemma 2.3** (RDP to  $(\varepsilon, \delta)$ -DP (cf. [26], Prop. 3)). *Let  $M$  be an  $(\alpha, \varepsilon)$ -RDP mechanism with  $\alpha > 1$  and  $\varepsilon \geq 0$ . Then,  $M$  also satisfies  $(\varepsilon', \delta)$ -DP with*

$$\varepsilon' = \varepsilon + \frac{\ln 1/\delta}{\alpha - 1} \tag{2.18}$$

for all  $\delta \in (0, 1)$ .

This transformation is proofed in [26] by showing the even stronger statement that  $\Pr[M(D) \in R] \leq \max\{\Pr[M(D') \in R], \delta\}$  using the Hölder inequality.

### 2.2.2 Personalized Differential Privacy

The goal of every mechanism should be to generate utility by means of its result  $r := M(D)$ . The definitions of DP (Definition 2.4) and RDP (Definition 2.6) provide the same privacy guarantee for all data that is processed by the same mechanism. In the literature, three similar personalized extensions of DP can be found that were all proposed in 2015, namely *heterogeneous DP* (HDP) [2], and *personalized/personalised DP* (PDP) [12, 19]. In this thesis, the notion of PDP is preferred as in most works in this field [5, 24, 28, 43].

Instead of a single privacy parameter epsilon, there needs to be one privacy parameter for each data point. In PDP, this is done by a *privacy specification*  $\Phi: O \rightarrow \mathbb{R}_+$  that maps each data owner  $o \in O$  to a parameter  $\varepsilon_o$ . Here, and in the following,  $O$  is the set of all data owners and a data point  $d_o$  is said to belong to an owner  $o$ . This is based on the assumption<sup>1</sup> that every data owner only provides one data point.

**Definition 2.7** (Personalized Differential Privacy). Let  $\mathcal{D}$  and  $\mathcal{R}$  be the sets of all possible data points and all possible processing results of a mechanism on any dataset, respectively. A mechanism  $M: \mathcal{D}^* \rightarrow \mathcal{R}$  meets  $\Phi$ -PDP (cf. [19], Def. 6) if for each data owner  $o \in O$ , all neighboring datasets  $D, D' \subseteq \mathcal{D}$  with  $D \stackrel{d_o}{\sim} D'$ , and all possible result events  $R \subseteq \mathcal{R}$

$$\Pr[M(D) \in R] \leq e^{\varepsilon_o} \cdot \Pr[M(D') \in R] . \quad (2.19)$$

Analogously, personalized  $(\varepsilon, \delta)$ -DP can be defined s.t. each data point  $d_o$  has an individual  $\varepsilon_o$  and  $\delta_o$ . PDP composes analog to  $(\varepsilon, \delta)$ -DP (see Proposition 2.1) with the difference that it applies element-wise. Therefore, the proof is analog to Proposition 2.1 as well.

**Lemma 2.4** (Composition of PDP). *Let  $M_1, M_2$  be two mechanisms satisfying  $(\varepsilon_o^{(1)}, \delta_o^{(1)})$ -, and  $(\varepsilon_o^{(2)}, \delta_o^{(2)})$ -DP for each data point  $d_o$  in a dataset  $D$ , respectively. The composition of  $M_1$  and  $M_2$  meets  $(\varepsilon_o^{(1)} + \varepsilon_o^{(2)}, \delta_o^{(1)} + \delta_o^{(2)})$ -DP for each data point  $d_o$  (cf. [19], Thm. 4).*

In the remainder of this work,  $\Phi$ -PDP is indicated only by pointing out individual epsilon values. Analogously,  $(\varepsilon, \delta)$ -DP, RDP, as well as other properties that are personalized are simply described as being individual per data point.

---

<sup>1</sup>Note that this assumption is made only for the sake of brevity while a violation of it would not result in any errors but in more complicated notations.



## 3 Related Work

This chapter addresses related research that deals with individualized DP.

### 3.1 Alternative Individual Privacy Definitions

Although DP and many of its variants were examined extensively, personalized DP is comparatively unpopular. However, two notions similar to PDP are described in this section.

#### 3.1.1 Heterogeneous Differential Privacy

HDP ([2], Def. 7) is very similar to PDP and shares the same goal, namely the extension of DP s.t. each data point has an individual privacy guarantee. Nevertheless, it is defined separately below, since one mechanism makes use of this exact notion (see Section 3.2.1).

**Definition 3.1** (Heterogeneous Differential Privacy). Let  $\mathcal{D}, \mathcal{R} \subseteq \mathbb{R}$  be the sets of all possible data points, and processing results  $R \in \mathcal{R}$  over all possible subsets of  $\mathcal{D}$ , respectively, s.t. data points have ratio scale<sup>1</sup> (cf. [2], Def. 4–7). Further, let  $\vec{v} \in [0, 1]^n$  be a *privacy vector* that determines the privacy values of all  $n$  data points s.t. the  $i$ -th data point requires  $(v_i \cdot \varepsilon)$ -DP. A mechanism  $M : \mathcal{D}^n \rightarrow \mathcal{R}$  meets  $(\varepsilon, \vec{v})$ -HDP ([2], Def. 7) if for all positions  $i \in \{0, \dots, n-1\}$ , for all datasets  $D, D' \in \mathcal{D}^n$  that only differ on their  $i$ -th data point and for all possible result events  $R \subseteq \mathcal{R}$

$$\Pr [M(D) \in R] \leq e^{v_i \cdot \varepsilon} \cdot \Pr [M(D') \in R] . \quad (3.1)$$

---

<sup>1</sup>In statistics, variables are classified regarding the nature of their information after Stevens' topology [35]. Thus, four classes of measurements of scale can be distinguished: The *nominal* scale only allows equality comparisons, while the *ordinal* scale additionally enables sorting, and the *interval* scale extends the possibilities of comparison by a distance measure. Finally, the *ratio* scale corresponds to a continuous space similar as the interval scale but in addition it includes a zero value and their values can be compared by ratios.

### 3.1.2 Per-Instance Differential Privacy

In 2017, a notion similar to HDP and PDP was proposed in [40] where a personalized DP of the fixed data point  $d_o$  regarding the fixed dataset  $D$  was considered. The methodical approach of that so-called *per-instance DP* (pDP) enables a more precise analysis and adjustment of privacy than PDP since it considers a concrete dataset instead of all possible datasets.

**Definition 3.2** (Per-Instance Differential Privacy). Let  $\mathcal{D}$  and  $\mathcal{R}$  be the sets of all possible data points and all possible processing results of a mechanism  $M: \mathcal{D}^* \rightarrow \mathcal{R}$  on any dataset, respectively. For a fixed data point  $d \in \mathcal{D}$ , and a fixed dataset  $D \subseteq \mathcal{D}$ ,  $M$  meets  $(\varepsilon, \delta)$ -pDP ([40], Def. 2.2) for  $d$  regarding  $D$  if for every  $R \in \mathcal{R}$

$$\Pr [M(D) \in R] \leq e^\varepsilon \cdot \Pr [M(D_{+d}) \in R] + \delta, \text{ and} \quad (3.2)$$

$$\Pr [M(D_{+d}) \in R] \leq e^\varepsilon \cdot \Pr [M(D) \in R] + \delta. \quad (3.3)$$

However, there are implementation problems of pDP that were not solved yet. For example, attackers could exploit the fact that pDP depends on the dataset by inserting corrupted data. Another severe problem is that the privacy of a data point cannot be determined before all other data points are fixed.

## 3.2 Mechanisms to Meet PDP

After several notions of personalized DP were defined, this section describes some mechanisms that satisfy PDP. Note that if not stated differently, the same notations as in Section 2.2.2 are used.

### 3.2.1 Stretching Mechanism

A useful measure to evaluate a mechanism is the *global sensitivity* ([11], Def. 3.1).

**Definition 3.3** (Global, Local, Modular Sensitivity). The global sensitivity

$$\Delta(M) := \max_{D \sim D'} \|M(D) - M(D')\|_1, \quad (3.4)$$

or  $\Delta_M$  for short, is the maximum distance of a mechanism  $M$  on any neighboring datasets  $D, D' \in \mathcal{D}$ . Here,  $\|\cdot\|_1$  denotes the  $\ell_1$ -norm and  $- : \mathcal{R}^2 \rightarrow \mathbb{R}^d$  refers to

some difference measure on  $\mathcal{R}$  with output dimension  $d$ . If  $D$  is fixed,  $\Delta_M$  is the *local sensitivity*, instead ([2], Def. 7.1). Analogously, the *modular sensitivity*  $\Delta_{M,i}$  ([2], Def. 8) is the global sensitivity of  $M$  if  $D$  and  $D'$  are any non-fixed datasets that differ exactly on the  $i$ -th data point, i.e. neighboring in the original sense.

The notion of sensitivity can be used to determine the magnitude of distortion needed for the intended amount of privacy, e.g. as in the Laplace-mechanism [10]. The modular sensitivity in particular, can be used to achieve HDP.

The only mechanism that was proposed especially for HDP is the *stretching mechanism* ([2], Sec. 3.2). Intuitively, the stretching mechanism scales all data points by individual factors so that a homogeneous perturbation determined by  $\varepsilon$  affects each data point with individual relative intensity. The scale values are searched in an optimization process, beforehand. In that process, a diagonal matrix  $T : [0, 1]^n \rightarrow \mathbb{R}^{n \times n}$  is approximated s.t.

$$\Delta_i(M(T(\vec{v}) \cdot D)) \leq v_i \cdot \Delta_M \quad (3.5)$$

holds for all  $i$ . It is calculated successively by maximizing each the  $i$ -th diagonal value  $t_i$  of  $T$  so that the above inequality still holds. Thus, its modular sensitivity  $\Delta_{M,i}$  is shrunk by the factor  $v_i$  for each privacy preference  $v_i \in \vec{v}$  in order to achieve HDP. Besides the resource-expensive optimization process for finding adequate scale values, a major drawback of the stretching mechanism is that it is only defined for ratio scaled datasets and real-valued mechanisms. Hence, it cannot directly be used for ML algorithms. A similar mechanism that is also based on manipulating sensitivity and assumes real-valued results was proposed in [6].

### 3.2.2 Sample Mechanism

In [19], two mechanisms that meet PDP were proposed. The first one is the *sample mechanism* that randomly keeps data points with a probability corresponding to their owner's privacy preferences.

**Definition 3.4** (Sample Mechanism). Formally, let  $\min_o \varepsilon_o \leq \varepsilon \leq \max_o \varepsilon_o$  be a threshold,  $M$  be an  $\varepsilon$ -DP mechanism,  $D$  be a dataset, and  $\Phi$  be a privacy specification. The sample mechanism  $S_M$  ([19], Def. 9) builds a new dataset  $\tilde{D} \subseteq D$  where each data point  $d_o \in D$  has probability

$$p_{d_o} := \begin{cases} \frac{e^{\varepsilon_o} - 1}{e^\varepsilon - 1} & \text{if } \varepsilon_o < \varepsilon \\ 1 & \text{else} \end{cases} \quad (3.6)$$

to be contained by  $\tilde{D}$ . Then,  $M$  is applied to  $\tilde{D}$ .

### 3 Related Work

The sample mechanism protects privacy when used for summary statistics but it is not adequate for ML algorithms. When thinking about model inversion attacks like the one proposed in [14], the sample mechanism cannot hold the privacy guarantee that it promises. This results from the fact that model inversion attacks are able to regenerate data points. Therefore, it is insufficient to lower their probability to be included in the training dataset. Instead the information about them has to be perturbed.

#### 3.2.3 Personalized Exponential Mechanism

The second mechanism proposed in [19] is an extension of the exponential mechanism ([25], Def. 2) which is a popular mechanism to achieve  $\varepsilon$ -DP for simple aggregation functions. Both exponential mechanisms determine the probability of each result to be output by the mechanism on a given dataset. In order to define them, two other functions need to be defined beforehand.

**Definition 3.5** (Difference between Datasets). Let  $\mathcal{D}$  be the set of all possible data points. Then, for any two datasets  $D, \tilde{D} \subseteq \mathcal{D}$ ,

$$D \oplus \tilde{D} := (D \setminus \tilde{D}) \cup (\tilde{D} \setminus D) \quad (3.7)$$

is the difference between  $D$  and  $\tilde{D}$  ([19], Chpt. 3, Sec. 3), i.e. set of data points in which  $D$  and  $\tilde{D}$  differ.

**Definition 3.6** (Substitution Distance). Let  $D \subseteq \mathcal{D}$  be any dataset,  $M$  be a mechanism, and  $r$  be any possible result of  $M$ . Then,

$$\sigma_M(D, r) := \max_{M(\tilde{D})=r} |D \oplus \tilde{D}|, \quad (3.8)$$

or  $\sigma_r$  for short, denotes the substitution distance ([19], Chpt. 3, Sec. 3) of  $M$  on  $D$  to achieve  $r$ , i.e. the negated minimal number of data points to be substituted in  $D$  s.t.  $M(\tilde{D}) = r$  where  $\tilde{D}$  is the substituted dataset.

If  $M$  already outputs  $r$  on  $D$ , then  $\sigma_r = 0$ . The original exponential mechanism sets each result's probability proportional to  $e^{\varepsilon \cdot \sigma_r}$ . Contrary, the *personalized exponential mechanism* (Definition 3.8) sets each result's probability proportional to  $e^{\frac{1}{2} \cdot \varepsilon \cdot \rho_r}$  where  $\rho_r$  is the personalized substitution distance defined below.

**Definition 3.7** (Personalized Substitution Distance). Let  $D \subseteq \mathcal{D}$  be any dataset,  $M$  be a mechanism,  $r$  be any possible result of  $M$ , and  $\Phi$  be a privacy specification.



Then,

$$\rho_M(D, \Phi, r) := \max_{M(\tilde{D})=r} \sum_{d_o \in D \oplus \tilde{D}} -\varepsilon_o, \quad (3.9)$$

or  $\rho_r$  for short, denotes the personalized substitution distance ([19], Def. 10, Eq. 5) of  $M$  on  $D$  to achieve  $r$ .

Finally, the personalized exponential mechanism is defined as follows.

**Definition 3.8** (Personalized Exponential Mechanism). Let  $M$  be a mechanism,  $\mathcal{R}$  be the set of  $M$ 's possible results,  $D$  be a dataset, and  $\Phi$  be a privacy specification. The personalized exponential mechanism ([19], Def. 10, Eq. 4)  $\mathcal{PE}_{M,\Phi}$  outputs each result  $r \in \mathcal{R}$  with probability

$$\Pr[\mathcal{PE}_{M,\Phi}(D) = r] := \frac{e^{\frac{1}{2} \cdot \rho_r}}{\sum_{\tilde{r} \in \mathcal{R}} e^{\frac{1}{2} \cdot \rho_{\tilde{r}}}}. \quad (3.10)$$

An even further extension of the personalized exponential mechanism was proposed in [28] (Def. 6)—the *utility-aware personalized exponential mechanism*. That mechanism additionally takes into account a quantitative distance  $z_r$  between the actual result and each possible result with  $z_r := |M(D) - r|$ . Hence, each result's probability is proportional to  $e^{\varepsilon \cdot \rho_r - z_r}$ .

All variants of the exponential mechanism are not suitable for ML algorithms due to several reasons. Most ML algorithms are intrinsically probabilistic and thus, several results on the same dataset are possible. Conversely, the exponential mechanisms assume the underlying mechanisms to be deterministic. Further, the exponential mechanisms are based on the computation of the underlying mechanism on a large number of datasets. But since the time costs to compute the result of an ML algorithm are very high, it is infeasible to compute the probability of each ML result. Note that the exponential mechanism and its variants are similar to the softmax activation function in ML.

### 3.2.4 Partitioning Mechanisms

Promoting the concept of PDP, the authors of [24] proposed PDP-achieving mechanisms based on partitioning. A *general partitioning mechanism* is defined as follows.

### 3 Related Work

**Definition 3.9** (General Partitioning Algorithm). Let  $\text{partition}_{\Phi,k}(D) := D_1, \dots, D_k$  be an algorithm that outputs  $k$  partitions of the dataset  $D$  given privacy specification  $\Phi$  and a number  $k$ . Let  $M_\varepsilon$  be a privacy-adjustable mechanism that processes data with  $\varepsilon$ -DP. Finally, let  $\text{ensemble}(r_1, \dots, r_k) = r$  be an ensemble algorithm that combines  $k$  results  $r_i \in \mathcal{R}$  to a final result  $r \in \mathcal{R}$ . Then, the general partitioning mechanism ([24], Def. 3) is defined by

$$\text{GP}_M(D) := \text{ensemble}(M_{\varepsilon_1}(D_1), \dots, M_{\varepsilon_k}(D_k)) . \quad (3.11)$$

Consequently, each data point has a DP guarantee that corresponds to the epsilon of its partition which in turn equals the smallest one among its data points. In [24], two different partitioning algorithms that implement the general partitioning algorithm are elaborated while the search for good ensemble methods is left for subsequent research. Both algorithms are described below.

**Definition 3.10** (Privacy-Aware Partitioning). The first proposed partitioning algorithm minimizes the waste of privacy budgets and is hence called *privacy-aware partitioning* ([24], Def. 4). Before partitioning, the dataset is sorted according to the privacy budgets  $(\varepsilon_1 \leq \dots \leq \varepsilon_n)$ . Then, a brute-force algorithm tries every partitioning where only adjacent epsilons are within the same partition. The output is the partitioning that minimizes the value of the *waste function*

$$\Omega_\Phi(D) := \sum_{i=1}^k \omega_\Phi(D_i) . \quad (3.12)$$

$\Omega_\Phi(D)$  sums up the squared wastes  $\omega_\Phi(D_i) := \sum_{d_o \in D_i} (\varepsilon_o - \varepsilon_{\min,i})^2$  of each partition  $D_i$  where  $\varepsilon_{\min,i}$  is the smallest privacy budget corresponding to a data point in  $D_i$ . In addition to minimizing privacy budget waste, a minimum threshold  $m$  of data points per partition has to be ensured. That threshold depends on the applied mechanisms. The overall time consumption to find the best privacy-aware partitioning is in  $O(\frac{n^2}{m} \cdot \log n)$  where  $n$  is the size of  $D$  ([24], Sec. 4.1).

**Definition 3.11** (Utility-Based Partitioning). In order to maximize utility while complying with personal privacy requirements, the second approach takes into account the intended mechanism, as well as the privacy preferences from  $\Phi$ . Using any function  $\lambda_{M,\Phi}(D_i) =: \lambda_i$  that measures the utility of  $M$  on  $D_i$  given  $\Phi$ , the *utility-based partitioning* ([24], Def. 5) maximizes the *utility function*

$$\Lambda_{M,\Phi}(D) := \sum_{i=1}^k \lambda_i \quad (3.13)$$

of  $M$  on the entire dataset  $D$  given  $\Phi$ . In a brute-force search algorithm analogous to privacy-aware partitioning,  $\Lambda_{M,\Phi}(D)$  is maximized as well as the number  $k$  of

partitions. The overall time effort to find the best utility-based partitioning is in  $O(n \cdot \log n)$  ([24], Sec. 4.2). Note that although this partitioning algorithm is in a lower complexity class than the previous one, its time consumption would be much higher for ML algorithms since ML models have to be trained for every possibility in the brute-force algorithm. Therefore, this algorithm is practically infeasible to apply onto ML algorithms.

### 3.2.5 Rényi Privacy Filter

The only mechanism mentioned so far that can be applied to ML algorithms is the privacy-aware partitioning. In this section, an additional adequate mechanism that is based on RDP is described.

#### Privacy Filter

Intuitively, a privacy filter is a termination criterion for an adaptive composition of privacy-preserving mechanisms to ensure that a certain privacy budget is not exceeded. In an adaptive composition, some mechanisms depend on the output of the previous one. A privacy filter can be implemented by calculating the composed DP loss after every mechanism's application and revert the one that exceeds the DP budget, afterwards. Instead of reverting the application, a privacy filter would actually predict the exceedance beforehand and prevent the application, but the preceding description is more intuitive while equivalent in most cases. For the sake of brevity, the implementation details of the privacy loss estimation are skipped and the interested reader is referred to Algo. 2, Def. 4.1, and Lemma 4.2 in [13]. For RDP with a fixed  $\alpha$ , the composition of privacy losses in an adaptive composition of  $k$  mechanisms can be bounded by the sum of all particular losses  $\varepsilon_i$  each of mechanism  $M_i$ .

For a personalized privacy filter, the privacy losses of each data point must be tracked individually. Whenever the personal privacy budget of a data point is exhausted, the previous mechanism's application is reverted and the particular point is removed from the current dataset. In the case of an adaptive composition sequence where each mechanism depends on the previous one's output, the whole sequence' application is reverted. A privacy filter that tracks RDP costs and prevents them to exceed a certain privacy budget is called *Rényi privacy filter* [13], or RDP filter for short. Instead, a privacy filter that tracks  $(\varepsilon, \delta)$ -DP costs is called *DP filter*. An RDP filter can be converted into a DP filter using Lemma 2.3. Hence, the sum of squared RDP costs  $\sum_{i=1}^k \varepsilon_i^2$  of the composed mechanisms  $M_1, \dots, M_k$  must not exceed  $2 \cdot (\sqrt{\ln 1/\delta} + \varepsilon - \sqrt{\ln 1/\delta})$  ([13], Thm. 4.7) where  $\varepsilon$  and  $\delta$  constitute a privacy specification in terms of

### 3 Related Work

$(\varepsilon, \delta)$ -DP.

### Private Gradient Descent with Filtering

One important approach to achieve standard DP or RDP for ML is the private gradient descent (PGD) ([13], Algo. 7) that was proposed in [34]. Therein, first the gradient of the loss function corresponding to the ML model's weights and one data point is computed. Then, this gradient is clipped s.t. its  $\ell_2$ -norm is smaller or equal to some threshold. Afterwards, a Gaussian noise is added to the clipped gradient. Finally, the model's weights are updated by subtracting this noisy clipped gradient in the descent step.

An RDP filter enables personalized privacy accounting in PGD. In this case, each gradient is clipped individually and data points that reached their privacy limit are not included in the current dataset. The PGD with filtering, expressed in Algorithm 1, satisfies  $(\alpha, \frac{\alpha\beta_d}{2\sigma^2c^2})$ -RDP for each data point  $d \in \mathcal{D}$  ([13], Prop. 6.2).

---

**Algorithm 1:** PGD with filtering (cf. [13], Algo. 8)

---

**input :** dataset  $D$ , loss function  $\mathcal{L}$ , learning rate  $\eta$ , noise scale  $\sigma$ , clipping threshold  $c$ , privacy budget  $\beta_d$  for each data point  $d$   
**output:** learned model weights  $\theta$

---

```

initialize weights  $\theta$  at random, epoch  $\tau := 1$ , current dataset  $\tilde{D} := D$ 
while  $\tilde{D} \neq \emptyset$  do
    for  $d \in \tilde{D}$  do
         $g_\tau(d) \leftarrow \nabla_\theta \mathcal{L}(\theta; d)$  ▷ compute gradient
         $\bar{g}_\tau(d) \leftarrow g_\tau(d) \cdot \min\left(1, \frac{\min(c, \sqrt{\beta_d - \sum_{i=1}^{\tau-1} \|\bar{g}_i(d)\|_2^2})}{\|g_\tau(d)\|_2}\right)$  ▷ clip gradient
        if  $\sum_{i=1}^{\tau-1} \|\bar{g}_i(d)\|_2^2 \geq \beta$  then
             $\tilde{D} \leftarrow \tilde{D} - d$  ▷ remove point if privacy depleted
         $\bar{g}_\tau \leftarrow \frac{1}{n} \cdot \sum_{d \in \tilde{D}} \bar{g}_\tau(d)$  ▷ combine clipped gradients
         $\tilde{g}_\tau \leftarrow \bar{g}_\tau + \mathcal{N}(\mathbf{0}, \frac{1}{|\tilde{D}|} \sigma^2 c^2 \mathbf{I})$  ▷ add Gaussian noise
         $\theta \leftarrow \theta - \eta \cdot \tilde{g}_\tau$  ▷ take gradient step
return  $\theta$ 

```

---

Note that  $\mathbf{0}$ , and  $\mathbf{I}$  are the zero vector, and the identity matrix, respectively, both with dimension  $k := \dim(\theta)$ . Further,  $\mathcal{N}(\mathbf{a}, \mathbf{b})$  is the multidimensional Gaussian/normal distribution with mean  $\mathbf{a} \in \mathbb{R}^k$ , and covariance matrix  $\mathbf{b} \in \mathbb{R}^{k \times k}$  s.t. its probability density function is  $f(x) := \frac{\exp(-\frac{1}{2}(x-\mathbf{a})^T \mathbf{b}^{-1}(x-\mathbf{a}))}{\sqrt{(2\pi)^k |\mathbf{b}|}}$ .

### 3.2.6 Personalized Moments Accounting

In another work [1], a tighter bound on the privacy loss of the PGD without filtering was given by applying the so-called *moments accountant*.

#### Moments Accountant

Gradient Descents of ML models with several adjustable layers can be considered adaptively composed sequences of mechanisms since the gradient of each layer depends on the data and/or the output of the preceding layer that is named *aux* in the following. In every training step, the moments accountant is applied to each mechanism in an adaptively composed sequence. It bounds the influence of one arbitrary data point to the mechanism's result. This is done by applying the *moment generating function* (Definition 3.14) to the *privacy loss random variable* (Definition 3.13)—the ratio of the mechanism's result probabilities on two neighboring datasets. These probabilities correspond to the probability distributions of the noisy average clipped loss gradients. The difference between both distributions is a distortion of one of their location parameters resulting from the data point's clipped gradient that is only included in one dataset. Using the *composability* and the *tail bound* of the moments accountant, a privacy guarantee can be calculated for the whole sequence of adaptively composed mechanisms. For the following definitions let  $M$  be a mechanism,  $r$  be any result of  $M$ ,  $D$  and  $D'$  be neighboring datasets with  $D \stackrel{d}{\sim} D'$ , and *aux* be an auxiliary input that is the output of all previous mechanisms on  $D$ .

**Definition 3.12** (Privacy Loss). The privacy loss ([1], Eq. 1) of  $M$  on  $D$  regarding  $r$  is

$$\xi(r; M, \text{aux}, D, D') := \ln \frac{\Pr[M(\text{aux}, D) = r]}{\Pr[M(\text{aux}, D') = r]} . \quad (3.14)$$

Since there are many possible results of  $M$  on  $D$  and  $D'$ , the mere privacy loss is not very useful. Instead, the distribution of the privacy loss over all possible results is of interest. It is called privacy loss random variable.

**Definition 3.13** (Privacy Loss Random Variable). The privacy loss random variable ([18], Def. 3) is the privacy loss over all results given by

$$\Xi(M, \text{aux}, D, D') := \xi_{r \sim M(\text{aux}, D)}(r; M, \text{aux}, D, D') . \quad (3.15)$$

The privacy loss random variable is used to account privacy costs by applying the moment generating function to it.

### 3 Related Work

**Definition 3.14** (Moment generating function). The moment generating function ([1], Eq. 2) of order  $\mu$  of a random variable  $X$  is defined as

$$\alpha(\mu; X) := \mathbb{E}[\exp(\mu \cdot X)] . \quad (3.16)$$

**Definition 3.15** (Moments Accountant). Applying the moment generating function to the logarithm of the privacy loss random variable, the moments accountant ([1], Sec. 3.2) is given by

$$\alpha_M(\mu) := \max_{\text{aux}, D, D'} (\alpha(\mu; \ln \Xi(M, \text{aux}, D, D'))) . \quad (3.17)$$

The moments accountants of all particular mechanisms can be summed up due to the composability ([1], Thm. 2.1). It is stated as follows.

**Proposition 3.1** (Composability of the Moments Accountant). *Let  $M$  consist of a sequence of  $k$  adaptive mechanisms  $M_1, \dots, M_k$  where  $M_i : \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \rightarrow \mathcal{R}_i$ . Then, for any  $\mu$*

$$\alpha_M(\mu) \leq \sum_{i=1}^k \alpha_{M_i}(\mu) . \quad (3.18)$$

The composability is proofed in the appendix of [1]. It is done by simple transformations, starting at the privacy loss of an adaptive sequence of mechanisms. This final sum of particular moments accountants can be converted into an  $(\varepsilon, \delta)$ -DP guarantee using the tail bound ([1], Thm. 2.2) that is stated as follows.

**Proposition 3.2** (Tail Bound). *For any  $\varepsilon > 0$ ,  $M$  is  $(\varepsilon, \delta)$ -DP for*

$$\delta = \min_{\mu} \exp(\alpha_M(\mu) - \mu\varepsilon) . \quad (3.19)$$

A proof of the tail bound that is based on Markov's inequality can be found in the appendix of [1].

For the tail bound, it suffices to evaluate a few small integer orders  $\mu$  since  $\alpha_M(\mu)$  increases exponentially in  $\mu$  while  $\mu\varepsilon$  only increases linearly in  $\mu$ . Thus, the authors of [1] only used  $\mu \in \{1, \dots, 32\}$ . Moreover,  $\Pr[M(\text{aux}, D) = r]$  and  $\Pr[M(\text{aux}, D') = r]$  each correspond to the average clipped gradient plus Gaussian noise  $\tilde{g}_\tau = \bar{g}_\tau + \mathcal{N}(\mathbf{0}, |\tilde{D}|^{-1} \sigma^2 c^2 \mathbf{I})$  as in Algorithm 1. They only differ in the clipped gradient  $\bar{g}_\tau(d)$  of the single data point  $d$ . A more detailed description can be found in [1], Lem. 3.

### Personalized Moments Accountants

In [18], the moments accountant was divided into a downwards and an upwards moments accountant. These two parts are proposed to enable PDP by an individual moments accounting for each data point.

**Definition 3.16** (Upwards & Downwards Moments Accountant). The downwards moments accountant ([18], Def. 4) is defined as

$$\check{\alpha}_M(\mu; \text{aux}, D, d) := \alpha(\mu; \ln \Xi(M, \text{aux}, D, D_{-d})) \quad (3.20)$$

while the upwards moments accountant ([18], Def. 5) is defined as

$$\hat{\alpha}_M(\mu; \text{aux}, D) := \alpha(\mu; \ln \Xi(M, \text{aux}, D, D_{+d})) . \quad (3.21)$$

Using both, the original moments accountant can be recovered ([18], Eq. 1) by

$$\alpha_M(\mu) = \max_{\text{aux}, D} \left\{ \hat{\alpha}_M(\mu; \text{aux}, D), \max_{d \in \mathcal{D}} (\check{\alpha}_M(\mu; \text{aux}, D, d)) \right\} . \quad (3.22)$$

## 3.3 Specific Applications of PDP

PDP was used for different problems. In this section, some examples are listed.

One specific ML problem is to predict missing attributes of data points in a dataset. A key algorithm to do so is probabilistic matrix factorization (PMF). Researchers applied PDP through a modified sample mechanism to PMF in order to build a recommendation scheme [43]. In another work, a PDP model for social networks based on social distance was proposed [5]. It maximizes data utility via a game-theoretical approach, namely by achieving the Bayesian Nash equilibrium of data owners' and adversaries' strategies. The first approach of local PDP for histogram estimation was proposed in [39]. Histogram estimation is an important non-ML aggregation function where the number of data points that have a one as the value of a certain binary attribute, i.e. a count estimate, is estimated.

The notion of PDP (Section 2.2.2) enables a heterogeneous and thus personalized specification of privacy requirements within a dataset. Several mechanisms that achieve PDP were presented in Section 3.2, albeit only three of them suffice to be applied for ML. Thereof, privacy-aware partitioning (Definition 3.10) is the simplest approach but it has two drawbacks. On the one hand, the utility of ML algorithms might decrease significantly due to partitioning of the dataset into smaller groups. On

### 3 Related Work

the other hand, the greater the group size the greater is the waste of privacy budgets since all data points in the same group are processed with the same privacy guarantee. Neither of these problems occur with the Rényi privacy filter applied to PGD (Section 3.2.5) which was designed specifically for ML. The personalized moments accountants (Section 3.2.6) could constitute a minor improvement to the Rényi privacy filter. But since the concrete implementation details, as well as its complexity were not provided by their inventors, it cannot be considered an adequate mechanism for PDP by now. Consequently, PGD with filtering can be considered as the most promising state-of-the-art mechanism for personalized PPML.

## 3.4 Other Approaches to Individualized DP

The following two approaches contrast the idea of PDP but implement individualized DP as well.

The authors of [21] aimed at taking into account the individual privacy preferences of data owners similarly to PDP and HDP. But unlike PDP and HDP, each data owner's privacy preference is not applied directly to her data but used as a vote for a global privacy parameter instead. As a result, standard DP is used while the preferences of many owners can not be met. This voting approach actually achieves homogeneous DP.

A completely different scenario was considered by the authors of [20] in 2019. Therein, each data owner shares her data with multiple data users, but she has different trust to each of them. Therefore, a data owner creates perturbed copies of her data using *local DP* (LDP) [42]. Then, she provides the users access to one or more copies according to the trust.



## 4 Private Aggregation of Teacher Ensembles

Besides the PGD ([34]) and the moments accountant ([1]) that improves it, there is a fundamentally different approach for ML to satisfy DP. *Private Aggregation of Teacher Ensembles* (PATE) [30] is an ensembling method that achieves DP for arbitrary ML models in a semi-supervised manner. In the PATE approach, several ML models—called *teachers*—are trained each on a different partition of the original training dataset. Afterwards, these teachers determine labels for an unlabeled dataset by a noisy voting. The newly labeled dataset is then used to train another model—called *student*—that will be the final model to be published. By providing the labels for the unlabeled dataset, the teachers’ knowledge is transferred to the student. The privacy for the training data stems from both, the fact that the student has never seen the training data itself, as well as the noise incorporated in the voting process. The combination of both causes can be analyzed theoretically to provide a DP guarantee while the former cause even enables an intuitive understanding about the privacy protection to non-experts. A scheme of PATE is illustrated in Figure 4.1.

Four different PATE approaches were explored in [30], namely distillation, active learning, semi-supervised learning (the standard approach described above), and generative semi-supervised learning that is based on generative adversarial networks (GANs). The former two approaches were shown to perform worst while the GAN-based approach performs best [30]. Nevertheless, only the standard PATE approach is considered in this section since it constitutes the fundamental idea and an adaptation of it (see Section 4.2) even surpasses the generative PATE version [31].

Formally, let  $D$  be a labeled dataset as in Section 2.1 and let  $m := |\mathcal{Y}|$  be the number of different labels.  $D$  is partitioned into  $k$  disjoint datasets  $D_1, \dots, D_k$ , each corresponding to one teacher  $\hat{h}_{\theta_i}$  with  $i \in \{1, \dots, k\}$ . After training every teacher on its particular partition of the dataset, the unlabeled dataset  $\dot{D} := \{x_1, \dots, x_N\}$  of size  $N$  is labeled by the teacher voting. Thus, each point  $x \in \dot{D}$  is evaluated by each teacher.

**Definition 4.1** (Vote Count). The vote count  $n: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{N}$  of any class  $j \in \mathcal{Y}$  for

#### 4 Private Aggregation of Teacher Ensembles

any data point  $x \in \mathcal{X}$  is defined by

$$n_j(x) = \sum_{i=1}^k \mathbb{1}(\hat{h}_{\theta_i}(x) = j), \quad (4.1)$$

where  $k$  is the number of teachers, and the *indicator function*  $\mathbb{1}: \{\perp, \top\} \rightarrow \{0, 1\}$  maps  $\perp$  to zero and  $\top$  to one.

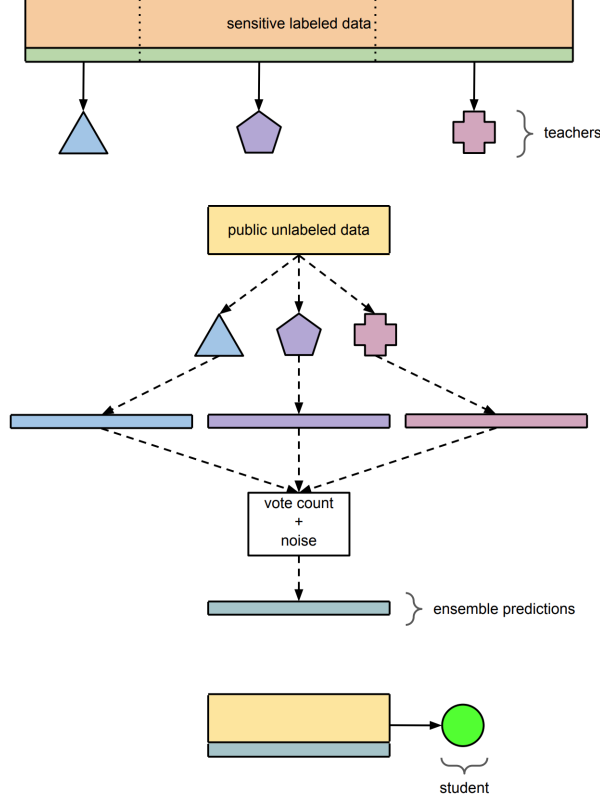


Figure 4.1: **PATE**. At first, the teachers—which could be of any ML model architecture adequate for the sensitive labeled data—are trained. Afterwards, public unlabeled data are given to the teachers and the resulting votings are aggregated s.t. labels are created. Finally, the student is trained on the public data together with the created labels.

In the first proposal of PATE [30], a Laplacian noise parameterized by some real value  $\gamma$  is added to the vote count. The teacher ensemble can predict a label for a data point by requesting each teacher’s prediction for it and then taking the class with the highest vote count as label prediction. In that case, every teacher would have a definite influence on the prediction whereby the privacy of its training data points would be very high. Therefore, the vote counts are perturbed by random noise.

**Definition 4.2** (Laplacian NoisyMax Aggregator). Let  $n_j$  be the vote count (Definition 4.1) of teachers for class  $j$  for each  $j \in \mathcal{Y}$ . The aggregation method of the teacher ensemble that applies Laplacian noise is called *Laplacian NoisyMax* (LNMax) aggregator. The LNMax aggregator  $\text{LNMax}: \mathbb{R} \times \mathcal{X} \rightarrow \mathcal{Y}$  with parameter  $\gamma \in \mathbb{R}$  on any data point  $x \in \mathcal{X}$  is defined by

$$\text{LNMax}_\gamma(x) := \arg \max_j \left\{ n_j(x) + \text{Lap} \left( 0, \frac{1}{\gamma} \right) \right\}. \quad (4.2)$$

$\text{Lap}(a, b)$  with parameters  $a, b \in \mathbb{R}$  is the Laplacian distribution which has the *probability density function* (PDF)  $f(x) = \frac{1}{2b} \exp(-\frac{|x-a|}{b})$ .

In practice, the teacher ensemble produces labels for a public unlabeled dataset that is later used to train a student ML model. Each ensemble's voting comes with small privacy costs for the training data of the teachers, but the resulting training dataset can be used limitless without increasing these costs. The votings are stopped, if the privacy budget of the sensitive data is exhausted. Formally, the resulting dataset  $\hat{D} = \{(x_1, \hat{y}_1), \dots, (x_N, \hat{y}_N)\}$  is used to train the student model  $\hat{h}_{\theta_{\text{student}}}$ .

## 4.1 Privacy Guarantee of PATE

There are two privacy bounds of the teachers' prediction depending on the kind of noise that is used. One bound only considers the kind and scale of noise while the other additionally regards the specific vote counts of a prediction. Due to their properties, the privacy bounds can be considered as loose and tight bound, respectively.

**Proposition 4.1** (Loose Bound of LNMax). *The LNMax aggregator with parameter  $\gamma$  satisfies  $(2\gamma, 0)$ -DP ([30], Thm. 2).*

This is analog to the Laplace mechanism's privacy cost. Since each particular data point influences only one teacher which in turn might change its vote, up to two vote counts (one of the increased label and one of the decreased label due to the changed vote) can differ by at most one on neighboring datasets  $D, D'$ . The claimed privacy bound is achieved by composability. Note that the loose privacy bound is not dependent on  $k$ .

A tighter bound of the privacy cost of a teacher voting can be computed by taking into account its vote counts that depend on the concrete predictions of the teachers and thus on the data.

**Proposition 4.2** (Tight Bound of LNMax). *Let  $M$  be the LNMax aggregator with parameter  $\gamma$  and let  $q \geq \Pr[M(D) \neq r]$  for some result  $r$ . Further, let  $\mu, \gamma \geq 0$  and  $q < \frac{e^{2\gamma}-1}{e^{4\gamma}-1}$ . Then, for any aux and any neighbor  $D'$  of  $D$ , the moments accountant of  $M$  is bounded by*

$$\alpha(\mu; \text{aux}, D, D') \leq \ln \left( (1-q) \cdot \left( \frac{1-q}{1-e^{2\gamma} \cdot q} \right)^\mu + q \cdot \exp(2\gamma \cdot \mu) \right) \quad (4.3)$$

(Thm. 3, [30]). The corresponding proof is based on construction. The probability threshold  $q$  can be bounded as well. For this purpose, the label  $j^*$  is considered to have largest vote count  $n_{j^*} \geq n_j$  for all labels  $j$ . Thus,

$$\Pr[M(D) \neq j^*] \leq \sum_{j \neq j^*} \frac{2 + \gamma(n_{j^*} - n_j)}{4e^{\gamma(n_{j^*} - n_j)}} \quad (4.4)$$

holds. This inequality can be proofed via convolution of the difference between two gamma distributions with parameters  $(2, 1)$  which equal the sum of two Laplace distributions with parameter 1 ([30], Lem. 4). Substituting  $q$  by this sum, and trying out a few small<sup>1</sup> integers for  $\mu$ , the tight bound of the teacher voting can be computed.

## 4.2 Improved PATE

One year after the PATE approach was proposed, a work about improvements of it was published [31]. These improvements comprise the use of a different noise distribution, the filtering of unconfident teacher votings using a threshold  $T \in [0, k]$ , and an interactive demanding of teacher labels. In principle, any noise can be used to induce randomness and thus enable privacy. Depending on the concrete noise distribution, the privacy guarantee is computed differently. The first adaptation enables lower noise scales while maintaining similar privacy guarantees. This is especially relevant when having many labels. The second adaptation prevents both, high privacy costs for single teacher votings, as well as low-quality labels due to disagreeing teachers. Similarly, the third adaptation enforces high-utility answers by ignoring data points the student already agrees on with the teacher ensemble. All three improvements of PATE can be considered as separate aggregation mechanisms.

Analog to the LNMax (Definition 4.2), the aggregation mechanism that only outputs the majority vote of the teachers perturbed by Gaussian noise is called *Gaussian NoisyMax* (GNMax) aggregator. It is defined as follows.

**Definition 4.3** (Gaussian NoisyMax Aggregator). Let  $n_j$  be the vote count (Definition 4.1) of teachers for class  $j$  for each  $j \in \mathcal{Y}$ . The GNMax aggregator  $\text{GNMax}: \mathbb{R} \times$

<sup>1</sup>In [30],  $\mu \in \{2, \dots, 8\}$  was proposed without further justification.

$\mathcal{X} \rightarrow \mathcal{Y}$  (cf. [31], Sec. 4.1) with parameter  $\sigma \in \mathbb{R}$  on any data point  $x \in \mathcal{X}$  is defined by

$$\text{GNMax}_\sigma(x) = \arg \max_j \{n_j(x) + \mathcal{N}(0, \sigma^2)\} . \quad (4.5)$$

$\mathcal{N}(a, b)$  is the Gaussian distribution with mean  $a \in \mathbb{R}$  and variance  $b \in \mathbb{R}$ .

A slightly extended aggregation mechanism is the *Confident-GNMax*. It additionally avoids the use of aggregated teacher labels if there is no consensus among the teachers and is defined in Algorithm 2. When taking into account the student's knowledge as well, the privacy cost can be estimated tighter when the student already consents with the teachers. This aggregation mechanism is called *Interactive-GNMax* and is defined in Algorithm 3.

---

**Algorithm 2:** Confident-GNMax Aggregator (cf. [31], Algo. 1)

---

**input** : data point  $x$ , threshold  $T$ , noise standard deviations  $\sigma_1, \sigma_2$

**output:** predicted label  $\hat{y}$

---

```

if  $\max_j \{n_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq T$  ▷ check if teachers consent
  then
    | return  $\arg \max_j \{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$  ▷ aggregate label
  else
    | return  $\perp$  ▷ output no label without teacher consensus

```

---



---

**Algorithm 3:** Interactive-GNMax Aggregator (cf. [31], Algo. 2)

---

**input** : data point  $x$ , threshold  $T$ , noise standard deviations  $\sigma_1, \sigma_2$ , confidence  $\gamma$ , number of teachers  $k$

**output:** predicted label  $\hat{y}$

---

```

 $\mathbf{p} \leftarrow g_{\theta_{\text{student}}}(x)$  ▷ get confidences of student on  $x$ 
if  $\max_j \{n_j(x) - k \cdot p_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq T$  ▷ check if student disagrees with teachers
  then
    | return  $\arg \max_j \{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$  ▷ aggregate label
  else if  $\max_j \{p_j(x)\} > \gamma$  ▷ check student's confidence
    then
      | return  $\arg \max_j \{p_j(x)\}$  ▷ reinforce student's prediction
    else
      | return  $\perp$  ▷ output no label without teacher consensus

```

---

### 4.2.1 Gaussian Mechanism

The *Gaussian mechanism* is a perturbation technique for data-processing functions using Gaussian noise. Compared to Laplacian noise, the tails of Gaussian noise diminish far more rapidly [31]. This advantage comes at the price of worse  $\varepsilon$ -DP guarantees, namely  $\infty$ -DP. However, RDP allows to provide small and simple privacy guarantees for the Gaussian mechanism.

**Definition 4.4** (Gaussian mechanism). Let  $f: \mathcal{D} \rightarrow \mathbb{R}$  be a function. Then, the Gaussian mechanism with noise scale  $\sigma$  is defined by

$$M_{f,\sigma}(D) = f(D) + \mathcal{N}(0, \sigma^2) . \quad (4.6)$$

**Lemma 4.3** (RDP guarantee of Gaussian mechanism). Let  $\Delta_f$  be the sensitivity of  $f$ . Then,  $M_{f,\sigma}$  satisfies  $(\alpha, \Delta_f^2 \cdot \alpha/2\sigma^2)$ -RDP ([26], Prop. 7).

This is proofed by direct computation in [26].

### 4.2.2 Privacy Guarantee of the Improved PATE

The privacy cost of the improved PATE is based on those of the GNMax since both extensions—the Confident-GNMax, and the Interactive-GNMax—are only heuristics for when to apply a voting. Note that the privacy costs are first measured in an RDP manner for a better analysis of Gaussian noise and thereafter they are converted from RDP to  $(\varepsilon, \delta)$ -DP.

**Lemma 4.4** (Loose Bound of GNMax). The GNMax satisfies  $(\alpha, \alpha/\sigma^2)$ -RDP for all  $\alpha \geq 1$  ([31], Prop. 8).

*Proof.* Every data point influences one teacher. Hence, if one data point changes, its corresponding teacher might vote differently. That difference is an increase and a decrease of two vote counts. Thus, the GNMax can be considered as the composition of two Gaussian mechanisms each with sensitivity = 1 where the vote counts are the underlying functions. Therefore, the loose bound is proofed using Lemma 4.3 and Lemma 2.2 (cf. [31]).  $\square$

This general privacy bound can be tightened in many cases by taking into account every vote count  $n_j$  which depends on the used dataset  $D$ .

**Lemma 4.5** (Tight Bound of GNMax). *Let  $M$  be the GNMax aggregator of function  $f$  with parameter  $\sigma$ . Then, for any class  $j^*$  the statement*

$$\Pr[M(D) \neq j^*] \leq \frac{1}{2} \sum_{j \neq j^*} \operatorname{erfc}\left(\frac{n_{j^*} - n_j}{2\sigma}\right) \quad (4.7)$$

([31], Prop. 7) holds.  $\operatorname{erfc}(\cdot)$  denotes the complementary error function defined by

$$\operatorname{erfc}(a) := \frac{2}{\sqrt{\pi}} \int_a^\infty e^{-t^2} dt. \quad (4.8)$$

Equation (4.7) can be used to compute a data-dependent privacy bound of the GNMax as follows. Let  $M$  simultaneously satisfy  $(\alpha_1, \varepsilon_1)$ -RDP and  $(\alpha_2, \varepsilon_2)$ -RDP. Both RDP bounds can be computed by applying the loose bound for two different alpha values. Suppose that  $1 \geq q \geq \Pr[M(D) \neq j^*]$  holds for a likely teacher voting  $j^*$ . Additionally suppose that  $\alpha \leq \alpha_1$  and  $q \leq e^{(\alpha_2-1) \cdot \varepsilon_2} / \left(\frac{\alpha_1}{\alpha_1-1} \cdot \frac{\alpha_2}{\alpha_2-1}\right)^{\alpha_2}$ . Then,  $M$  satisfies  $(\alpha, \varepsilon)$ -RDP for any neighboring dataset  $D'$  of  $D$  with

$$\varepsilon = \frac{1}{\alpha - 1} \cdot \ln \left( (1 - q) \cdot \left( \frac{1 - q}{1 - (q \cdot e^{\varepsilon_2})^{\frac{\alpha_2-1}{\alpha_2}}} \right)^{\alpha-1} + q \cdot \left( \frac{e^{\varepsilon_1}}{q^{\frac{1}{\alpha_1-1}}} \right)^{\alpha-1} \right) \quad (4.9)$$

(cf. [31], Thm. 6).

Lemma 4.5 is proofed in [31] by applying Jensen's inequality and showing the monotonicity of the bound. The optimal orders of the two RDP guarantees of  $M$  are  $\alpha_1 = \sigma \cdot \sqrt{\ln 1/q}$  and  $\alpha_2 = \alpha_1 + 1$  ([31], Prop. 10). The privacy cost of the GNMax can be bounded by the minimum of the data-dependent bound (Lemma 4.5), and the general bound (Lemma 4.4). The data-dependent privacy bound is only used if the requirements stated above are met and then optimal orders for the RDP guarantees are used. Note that the specific privacy costs of GNMax aggregations must not be published or else the privacy costs increase as a consequence.





## 5 Personalized PATE Variants

This chapter describes the main contribution of this thesis, namely three adaptations of PATE that achieve PDP.

### 5.1 Personalization Techniques for PATE

In order for a PATE to achieve PDP, the influence of each data point on the voting has to be controlled. However, every data point learned by a teacher is assumed to completely influence that teacher's voting behavior which is due to the worst-case guarantee property of DP. Thus, the influences of its data points are equal and consequently PDP can not be achieved trivially on data point level but on teacher level instead. The following three ideas enable personalized privacy expenditures.

1. One idea is to violate the partitioning principle of PATE so that data points could be used to train several teachers. This approach is described extensively in Section 5.2 and is named *upsampling PATE* referring to the upsampling technique that is popular in ML.
2. Another technique is about asking teachers with different frequencies in votings s.t. not all teachers participate in every voting. By varying the participation frequencies, individual privacy losses can be achieved on teacher level. This approach is named *vanishing PATE* and it is described in Section 5.3.
3. A similar approach that also enables privacy personalization on teacher level is to individualize the teachers' influence on votings. Hence, each teacher's vote is weighted differently so that some teachers have higher influence on the ensembled prediction than others. Section 5.4 formally introduces this approach named *weighting PATE*.

An approach that at first glance seems to provide PDP, is to perturb each teacher's prediction with an individual amount of noise. Though this is a fallacy, since all perturbed predictions are summed up in the aggregation mechanism and hence all noises can be considered as one stronger noise. A different idea is to train each teacher

with an individual DP guarantee or directly with PDP mechanisms. Such approaches would have some major drawbacks, like the worse accuracy of teachers. Moreover, the privacy preservation would be both, within teacher trainings as well as in votings. It is not clear, how to account privately learned teachers in votings.

All personalized variants that are described in the subsequent sections in detail share certain properties. Therefore, the concepts of individual sensitivity and the personalized GNMax are defined here to prevent redundancy.

**Definition 5.1** (Individual Sensitivity). Let  $\mathcal{D}$  be the set of all possible data points and let  $\mathcal{R} \subseteq \mathbb{R}$  be the set of all possible results of a mechanism  $M: \mathcal{D}^* \rightarrow \mathcal{R}$ . The *individual sensitivity* of  $M$  regarding the data point  $d_o \in \mathcal{D}$  is defined by

$$\Delta_M^{(o)} := \Delta^{(o)}(M) := \max_{D \stackrel{d_o}{\sim} D'} \|M(D) - M(D')\|_1 \quad (5.1)$$

analog to the global sensitivity from Definition 3.3.

**Definition 5.2** (Personalized Gaussian NoisyMax Aggregator). Let  $\bar{n}_j: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{N}$  be any personalized vote count of teachers for each class  $j \in \mathcal{Y}$ . The *personalized GNMax* (pGNMax) aggregator  $\text{pGNMax}: \mathbb{R} \times \mathcal{X} \rightarrow \mathcal{Y}$  with noise scale  $\sigma \in \mathbb{R}_+$  on any data point  $x \in \mathcal{X}$  is defined by

$$\text{pGNMax}(x) := \arg \max_j \{ \bar{n}_j(x) + \mathcal{N}(0, \sigma^2) \} . \quad (5.2)$$

The pGNMax is a generalization of the GNMax using an arbitrary vote count function. The personalization techniques for PATE define different vote counts and have individual sensitivities. Analog to Algorithm 2 and Algorithm 3, the *Confident-pGNMax* and the *Interactive-pGNMax* can be defined, respectively, by using any of the three personalization techniques proposed in this chapter.

## 5.2 Upsampling

The influence of each data point on teacher votings can be varied by varying the number of teachers that are trained on it. Hence, the higher the personal privacy budget of one point the more teachers can include that particular point in their training dataset. This technique does not change the non-personalized PATE except that the sensitive data are divided into overlapping sets instead of distinct partitions. This specific variant of pGNMax is named the *upsampling GNMax* (uGNMax). It is illustrated in Figure 5.1. The personalized vote count of the uGNMax aggregator actually equals the non-personalized vote count (Definition 4.1).

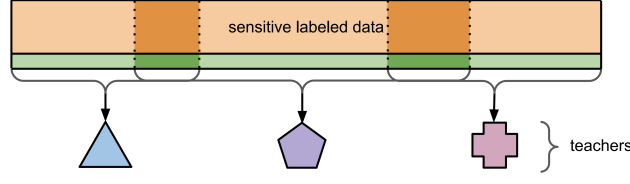


Figure 5.1: **Upsampling PATE.** PATE with upsampling teachers differs from PATE exclusively in the allocation of sensitive data points to teachers. Thus, data with higher privacy budgets are given to multiple teachers.

**Theorem 5.1** (Individual Sensitivity of the uGNMax). *Analog to the GNMax, the uGNMax can be considered as the composition of two Gaussian mechanisms regarding each data point  $d_o$  since every teacher might vote differently on neighboring datasets. The individual sensitivity of each Gaussian mechanism equals  $t_o$ . For brevity, it can be said that the individual sensitivity of uGNMax is the same.*

*Proof.* Analog to the GNMax, every teacher can change two vote counts in the uGNMax. But in contrast to the GNMax aggregator, each data point  $d_o$  has influence on  $t_o$  teachers. Therefore,  $d_o$  potentially leads to an increase of one vote count and a decrease of another vote count for each teacher it was trained on. In the worst case all these teachers consent on their label predictions without  $d_o$  and consent on a different label with  $d_o$  in their corresponding training datasets. Thus, the individual sensitivity of uGNMax regarding any data point  $d_o$  is  $t_o$ .  $\square$

Upsampling enables privacy personalization on data point level. On the one hand, upsampling allows very few and even single data points to have different privacy budgets than all others. On the other hand, privacy levels of only discrete distances are possible since data points can only have integer-valued duplications. The distances between several privacy levels can be approximated with arbitrary precision by increasing the number of duplications of all data points. However, more duplications of sensitive data require more teachers to be trained in order to keep a good privacy-utility-tradeoff. An increasing number of teachers to be trained coincides with higher time and space consumptions.

## 5.3 Vanishing

Another way to vary the influence of different sensitive data points on votings is for the teachers that were trained on them to omit participation in some votings. Thus, only the data points corresponding to participating teachers have privacy costs.

By decreasing the participation frequency of a teacher, the privacy costs of all its data points are decreased as well. If it is based on the Gaussian mechanism, such a mechanism is named *vanishing GNM*ax (vGNMax). It is illustrated in Figure 5.2.

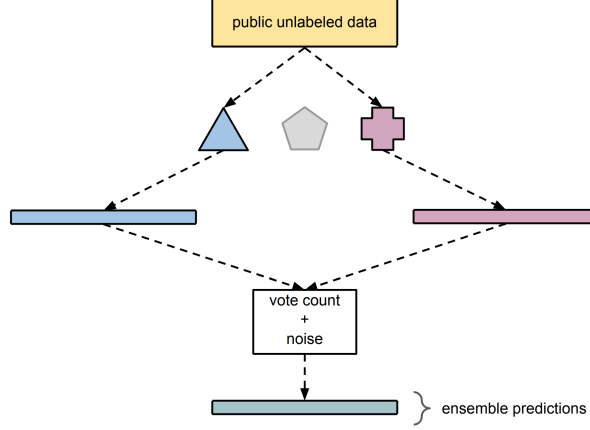


Figure 5.2: **Vanishing PATE**. Teachers participate on votings with a frequency according to their corresponding sensitive data points in vanishing PATE. Only the participating teachers produce privacy loss for their data.

**Definition 5.3** (Vanishing Vote Count). Let  $s := (s_1, \dots, s_k) \in \{0, 1\}^k$  be the current selection of the  $k$  teachers  $\{\hat{h}_{\theta_1}, \dots, \hat{h}_{\theta_k}\}$ , meaning that the  $i$ -th teacher's participation is determined by  $s_i$ . The vanishing vote count  $\hat{n}: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{N}$  of any class  $j \in \mathcal{Y}$  for any data point  $x \in \mathcal{X}$  is defined by

$$\hat{n}_j(x) := \sum_{i=1}^k s_i \cdot \mathbb{1}(\hat{h}_{\theta_i}(x) = j) . \quad (5.3)$$

**Theorem 5.2** (Individual Sensitivity of the vGNMax). *Analog to the GNM*ax and *uGNM*ax aggregators, the vGNMax can be considered as the composition of two Gaussian mechanisms regarding each data point  $d_o$  since every teacher might vote differently on neighboring datasets. The individual sensitivity of each Gaussian mechanism equals  $s_i$  regarding  $d_o$  learned by the  $i$ -th teacher. For brevity, it can be said that the individual sensitivity of vGNMax is the same.

*Proof.* Analog to the GNMax, every teacher can change two vote counts in the vGNMax. But in contrast, not all teachers participate in each voting. Therefore, only the teachers used in the current voting spend privacy of their corresponding data points by potentially change their vote if any data point  $d_o$  would not have been learned. Since each data point is only used to train one teacher, the claimed individual sensitivity is achieved.  $\square$

Vanishing personalizes privacy on teacher level, i.e. data points corresponding to the same teacher have equal privacy costs. The absence of some teachers in a voting results either in stronger relative perturbation of the fewer present teachers' vote counts, or in an adjustment of the induced noise' scale. The latter case leads to increased privacy costs. To minimize these costs, the same number of teachers  $\pm 1$  is sampled randomly for each voting. Another problem of vGNMax is the determination of participations. In order to minimize privacy expenditure, the number of participating teachers should be maximized in every voting. Nevertheless, teachers should still participate in a probabilistic manner so that the expertise of the teacher ensemble varies. Otherwise, it would be possible that the same ensemble of participating teachers that lacks knowledge about a certain class repeatedly predicts wrong labels for points of that class.

## 5.4 Weighting

Similar to the vGNMax aggregator, the teachers can be weighted differently instead of having different participation frequencies in the voting processes. This mechanism of weighting teachers that is based on the Gaussian mechanism is named *weighting GNM*ax (wGNMax). It is illustrated in Figure 5.3.

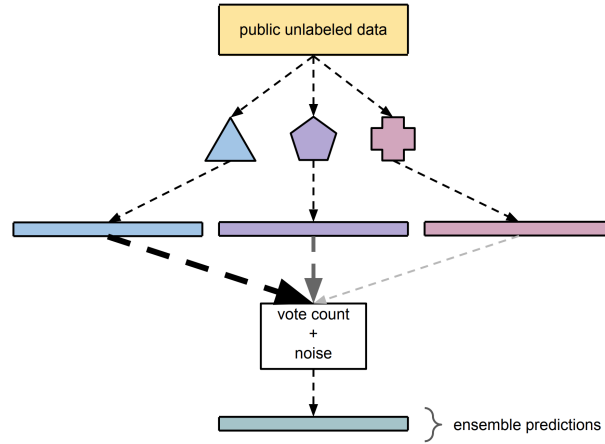


Figure 5.3: **Weighting PATE**. In weighting PATE, every teacher has a specific weight according to its corresponding sensitive data points. Its influence on votings directly depends on its weight.

**Definition 5.4** (Weighting Vote Count). Let  $\psi_i \in \mathbb{R}_+$  be the weight of the  $i$ -th teacher  $\hat{h}_{\theta_i}$  for all  $i \in \{1, \dots, k\}$ . The weighting vote count  $\tilde{n}: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{N}$  of any

class  $j \in \mathcal{Y}$  for any data point  $x \in \mathcal{X}$  is defined by

$$\tilde{n}_j(x) := \sum_{i=1}^k \psi_i \cdot \mathbb{1}(\hat{h}_{\theta_i}(x) = j) . \quad (5.4)$$

**Theorem 5.3** (Individual Sensitivity of the wGNMax). *Analog to the GNMax, uGNMax, and vGNMax aggregators, the wGNMax can be considered as the composition of two Gaussian mechanisms regarding each data point  $d_o$  since every teacher might vote differently on neighboring datasets. The individual sensitivity of each Gaussian mechanism equals  $\psi_i$  regarding  $d_o$  learned by the  $i$ -th teacher. For brevity, it can be said that the individual sensitivity of wGNMax is the same.*

*Proof.* As in the GNMax, every teacher might change two vote counts on neighboring datasets in the wGNMax. Contrary to the GNMax aggregator, the vote of the  $i$ -th teacher has weight  $\psi_i$ . Therefore,  $d_o$  learned by the  $i$ -th teacher potentially leads to an increase of one vote count and a decrease of another vote count by  $\psi_i$ .  $\square$

Analog to the vGNMax, the wGNMax aggregator provide PDP on teacher level. Both require enough data points that share the same privacy requirements s.t. at least one teacher can be trained with high accuracy. Alternatively, points with different budgets are combined to train the same teacher. In that case, the higher budgets are not completely exploited.

## 5.5 Privacy Guarantee of Personalized PATE

Different individual sensitivities for the data used in PATE that require privacy result in different RDP bounds. The loose bound that is also essential to the tight bound is affected as follows.

**Theorem 5.4** (Individual Loose Bound of Personalized GNMax). *For any data point  $d_o \in \mathcal{D}$ , a personalized GNMax aggregator  $M$  with noise scale  $\sigma \in \mathbb{R}_+$  satisfies an individual  $(\alpha, (\Delta_M^{(o)})^2 \cdot \alpha / \sigma^2)$ -RDP for all  $\alpha \geq 1$ .*

*Proof.* The personalized GNMax can be considered as the composition of two Gaussian mechanisms—since two vote counts can be changed by one teacher if it was trained on with or without a specific data point—where the vote counts are the underlying functions. Using Lemma 4.3 and Lemma 2.2, the claimed RDP guarantee is achieved.  $\square$

**Corollary** (Scaling Invariance of the Individual Loose Bound). *Let  $c \in \mathbb{R}_+$  be any positive scalar. Let  $M$  be a personalized GNMax aggregator with noise scale  $\sigma \in \mathbb{R}_+$  and an individual sensitivity  $\Delta_M^{(o)} \in \mathbb{R}_+$  for some data point  $d_o$ . Furthermore, let  $\tilde{M}$  be another personalized GNMax aggregator with noise scale  $\tilde{\sigma} = c \cdot \sigma$  and individual sensitivity  $\Delta_{\tilde{M}}^{(o)} = c \cdot \Delta_M^{(o)}$  for  $d_o$ . Then, the individual loose bounds of  $M$  and  $\tilde{M}$  regarding  $d_o$  are equal.*

*Proof.* Fix  $\alpha > 1$ .  $M, \tilde{M}$  satisfy individual  $(\alpha, \varepsilon)$ -, and  $(\alpha, \tilde{\varepsilon})$ -RDP for  $d_o$ . The equality of  $\varepsilon$  and  $\tilde{\varepsilon}$  is verified by direct computation.

$$\tilde{\varepsilon} := \left( \Delta_{\tilde{M}}^{(o)} \right)^2 \cdot \alpha / \tilde{\sigma}^2 \quad (5.5)$$

$$= \left( c \cdot \Delta_M^{(o)} \right)^2 \cdot \alpha / (c \cdot \sigma)^2 \quad (5.6)$$

$$= c^2 \cdot \left( \Delta_M^{(o)} \right)^2 \cdot \alpha / (c^2 \cdot \sigma^2) \quad (5.7)$$

$$= \left( \Delta_M^{(o)} \right)^2 \cdot \alpha / \sigma^2 \quad (5.8)$$

$$=: \varepsilon \quad (5.9)$$

□

Considering Section 5.5, it is obvious that the wGNMax with any parameters is equivalent to the wGNMax with the same parameters, except that the weights and noise standard deviation are scaled by any positive scalar  $c \in \mathbb{R}_+$ . Analogously, the uGNMax with any parameters is similar<sup>1</sup> to the uGNMax with the same parameters, except that the duplications, number of teachers, and noise standard deviation are scaled by  $c$ . In a nutshell, personalized PATE can be considered as multiple differently scaled PATEs corresponding to groups of data points that share the same individual sensitivity. So, the individual loose and tight bound regarding each group have to be computed s.t. the individual relative noise scale is divided by the individual sensitivity and therefore the individual sensitivity in turn is set to one afterwards. The individual tight bound of the pGNMax is hence equivalent to the tight bound of the GNMax, except for the scaled noise.

<sup>1</sup>If the additional teachers behave exactly as the others, both uGNMax aggregators would be equivalent. But in practice, every additional teacher is slightly different than others due to the probabilistic training and different training sets.





## 6 Evaluation

This chapter provides an evaluation of the proposed PATE extensions from Chapter 5. They were implemented in the programming language Python [37] (version 3.8) in an object-oriented manner. The most important code libraries that were used are NumPy [16], pandas [36], scikit-learn [32], and TensorFlow [7]. The visualization of experimentation results was done within Jupyter [33] notebooks using Matplotlib [17], and seaborn [41]. Except for the tight bound computation that was taken from [29], the complete PATE algorithm and the personalized extensions were newly implemented.

### 6.1 Datasets

Two fundamentally different datasets were used to evaluate the proposed approaches. The first one contains images of handwritten digits while the second one consists of census<sup>1</sup> data.

#### 6.1.1 Handwritten Digits Dataset

Several datasets are very popular within the ML community. Among them, the "Modified National Institute of Standards and Technology" (MNIST) database [23] of handwritten digits is an image dataset for classification tasks. It contains 70 000 ( $28 \times 28$ )-pixel grayscale images of the digits zero to nine (see Figure 6.1). The typical ML task for that dataset is to classify the images, i.e. predict numbers for unseen images after learning on a subset of the data.

Usually, *convolutional neural networks* (CNNs) (see [3], Sec. 5.5.6; [15], Chpt. 9) are used for image data. Their name stems from a kind of layer that applies a mathematical operation called convolution onto a 2-dimensional or 3-dimensional input matrix by moving a filter kernel over it. Convolutional layers output each a processed form of the input matrix for each filter kernel they contain. E.g. one

---

<sup>1</sup>a large survey to gather information about the population of a country

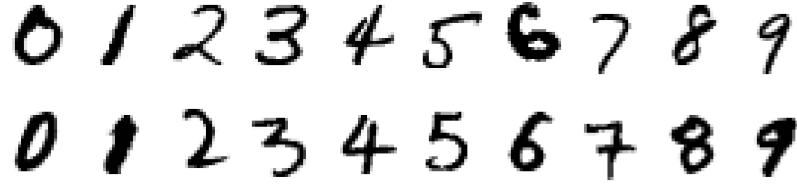


Figure 6.1: **MNIST Dataset**. This is an illustration of 20 randomly picked data points (2 of each class) in the MNIST dataset.

filter could emphasize vertical edges in an image by calculating vertical gradients of brightness values in local groups of pixels. By arranging multiple convolutional layers one after another, even more complex patterns can be emphasized.

A simple CNN architecture from [4] was used for the experiments. The architecture is described in Table 6.1. Note that the first and fifth layer were randomly initialized by sampling from the He uniform distribution. This CNN architecture was chosen since it achieves practical accuracies, requires little resources, and does not overfit on small training datasets which are required for PATE.

number	type of layer	parameters
1	convolutional	32 (3, 3)-kernels, relu activation
2	batch normalization	-
3	max pooling	size (2, 2)
4	flatten	-
5	fully connected	100 nodes, relu activation
6	batch normalization	-
7	fully connected	10 nodes, softmax activation

Table 6.1: **CNN for MNIST**. This is a tabular description of the CNN architecture that is used in this work for training on the MNIST dataset.

For the MNIST dataset, both, students and teachers were CNN models of the above kind. While teacher models were trained on 240 data points, students were trained on 50 to 2000 points, depending on the personalization parameters and the privacy consumption of concrete votings. In order to analyze the relationship between privacy and utility, labels were taken to train the student until any point of the lower budget

reached a multiple of 0.1 for each multiple of 0.1 until 2000 labels were taken. The training was done by batches of 10% of the training set size, but in every case between 16 and 64. Moreover, the Adam optimizer, and the categorical cross entropy loss were used for the gradient descent. For all parameters not covered by Table 6.1, the default settings from Tensorflow [7] (version 2.4.1) were kept.

In order to improve training performances, a data augmentation was used for both, teachers and students. It was done by a random rotation of up to  $\pm 7.5^\circ$  and a random shift of up to 7% in horizontal and vertical direction. Note that the preprocessing of data points per model does not affect privacy losses. In the teachers' case, each data point is assumed to completely influence its corresponding teacher in every vote. Thus, data augmentation that is not dependent on other data points does not change the privacy guarantee. In the student's case, the data to be trained on already consumed privacy budgets and data augmentation of it can be considered post-processing. DP guarantees are immune to post-processing ([11], Prop. 2.1).

### 6.1.2 Census Income Dataset

In order to evaluate our approaches on different conditions, i.e. different data and ML models, the adult income dataset that is provided by the "UCI Machine Learning Repository" [8] was selected as a second dataset. It comprises 48 842 data points from the US census database of the year 1994. An extraction of a few data points are shown in Table 6.2. The features are age, workclass, fnlwgt<sup>2</sup>, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country, and income. The income feature is used as label for the classification task.

age	workclass	fnlwgt	education	...	native-country	income
30	State-gov	141297	Bachelors		India	>50K
34	Private	245487	7th-8th		Mexico	<=50K
25	Self-emp-not-inc	176756	HS-grad	...	United-States	<=50K
38	Private	28887	11th		United-States	<=50K
43	Self-emp-not-inc	292175	Masters		United-States	>50K

Table 6.2: **Adult Dataset.** This is an extraction of five arbitrarily selected data points in the adult income dataset.

An adequate ML model for the classification task of the adult income dataset is a *random forest*. A random forest comprises several *decision trees*. A decision tree in turn divides the training data by means of a threshold regarding one feature s.t.

<sup>2</sup>the final weight indicates the number of citizens with similar features

## 6 Evaluation

data points with different labels are separated best by the threshold. This process is repeated during training until a certain termination criterion is met. In the prediction process, a data point is led along the decision boundaries to a leaf node. Then, the ratio of the number of training data points to have a certain label to the total number of points within the leaf is given as probability for that label. A random forest is an ensemble of multiple decision trees that were all trained on the same training data.

For the experiments in this work, 100 decision trees were used for the random forest models. Contrary to the MNIST dataset, no data augmentation was used, since none could be found to improve learning performances. Furthermore, all 3 620 damaged data points were removed and all categorical features were transformed into numerical features by assigning a number to each expression of a feature. Additionally, all numerical features were normalized to values between zero and one.

### 6.2 Experiments

In this section, the experiments to evaluate the personalized PATE variants are analyzed. For the sake of brevity, only the MNIST experiments are visualized since the information of interest is very similar on both datasets.

In all experiments, the Confident-GNMax (see Algorithm 2) with parameters as in [31] is used. The parameters are shown in Table 6.3.

dataset	# teachers	# data (public/private/test)	$\sigma_1$	$\sigma_2$	$T$	$\delta$
MNIST	250	60 000/9 000/1 000	150	40	200	$10^{-5}$
Adult	250	37 222/7 000/1 000	200	40	300	$10^{-5}$

Table 6.3: **PATE Parameters.** This shows the parameters for the Confident-GNMax on the Adult and MNIST datasets used in this work.

For the MNIST experiments, three different random seeds were used for each parameter combination of the personalized while ten were used for the non-personalized approach. For the Adult experiments, ten different random seeds were used for each parameter combination, instead. The teachers created labels for the public dataset five times where each time the set was shuffled differently. Hence, more data could be generated without training more teachers. The resulting labels were used for the training dataset of a student model. Every tenth of  $\epsilon$ , the right amount of labels was used so that their privacy costs of the lower group did not exceed that value. Hence, many students were trained on different sizes of training sets with different privacy costs.

All personalized PATE variants were adjusted to be comparable to the non-personalized algorithm. This means in particular that more teachers were used in the upsampling approach s.t. every teacher still had the same amount of training data as without upsampling. Additionally, the noise scale was increased accordingly. For example, one half of the sensitive data is duplicated once while the other half is not duplicated. Then, the total number of data points to use in PATE increases by 50%. In order to be best comparable to the GNMax, 375 teachers and a noise scale of 60 are taken instead of 250 teachers and a noise scale of 40. In the case of vanishing PATE, the noise was scaled down according to the number of participating teachers so that the same voting quality as in non-personalized PATE was kept. For example, half of the teachers only participated in every second voting. Then, 187 or 188 teachers voted in every voting. Therefore, the original noise scale was scaled by  $\frac{\# \text{ participating teachers}}{\text{total } \# \text{ teachers}}$  to keep a comparable voting quality. Lastly, the number of weighted teachers, as well as the noise scale remained the same, but the weights were selected s.t. the sum of all teachers' weights equaled the number of teachers. These adjustments kept the original optimizations regarding the privacy-utility-tradeoff of the Confident-GNMax for the datasets. For example, the extra information from data with higher budgets in upsampling PATE could be used to increase the training set size per teacher. This would result in slightly higher accuracies of the teachers. Instead, more teachers were used and the noise scale was increased while every teacher had the same amount of data. Thus, the teachers had equal accuracies but the privacy loss was decreased significantly.

### 6.2.1 Utility

The utility of teacher and student models was measured by their accuracy on the test dataset. The dependence of the accuracy on the amount of training data is visualized in Figure 6.2. For all variants, the accuracies of the teachers are mostly between 85% and 94% and averaging at  $\approx 90.2\%$  for 240 training data points per teacher. This leads to accuracies of the voted labels compared to the correct labels of the public dataset at  $\approx 97.7\%$  after  $\approx 43\%$  of the voted labels were rejected for all variants. Note that the teachers' accuracies could be higher when using stronger models and more computing resources, but the effect on the rejecting rate and the label quality would be marginal. Due to the used data augmentation, the achieved teacher accuracies are still significantly higher than those in [31]. The accuracies there have a mean of 83.86% and the voting accuracy is 93.18%. The same model architecture (see Table 6.1) was used for the students and for the teachers. But more training time was provided for students for the benefit of higher student accuracies. Analog to the teachers, the accuracy of students is equal across the different variants, depending on the number of training points. The difference lies within the privacy loss and computing resources. For the Adult experiments, the teachers' accuracies are mostly between 75% and 85% and their average is at  $\approx 81\%$ . Each teacher was

## 6 Evaluation

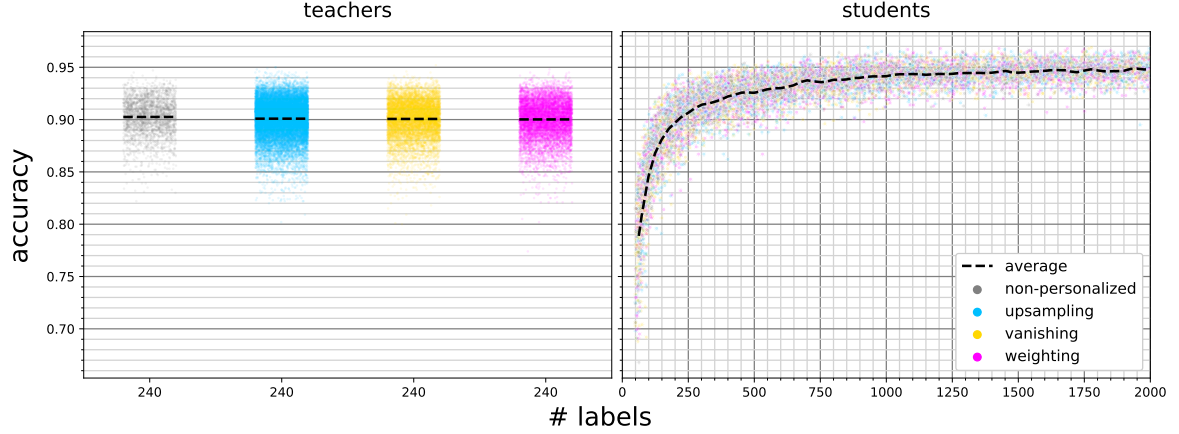


Figure 6.2: **Information vs. Utility.** The accuracies of teacher and student models compared to the number of their training data points are illustrated. The number of teacher models evaluated per PATE variant was 32 244 for uGNMax, and 28 496 for both, vGNMax and wGNMax while 2 500 of the non-personalized GNMax were used. 6 355, 6 339, and 6 361 students are evaluated for the uGNMax, vGNMax, and wGNMax, respectively. For standard GNMax, 1 337 student models are shown. For every PATE, the first labels—until any cost of data with the lower budget reached a multiple of 0.1 for each multiple until 2 000 labels were reached—were taken to train a student.

trained on 149 training points. The rejecting rate was  $\approx 64.3\%$  and the accuracy of the voted labels that averages at  $86.3\%$  is mostly between  $84\%$  and  $88\%$ . The students' accuracies reach a plateau at about 300 training data points with  $\approx 81\%$  until  $\approx 82\%$  at 2 000 data points.

### 6.2.2 The Advantage of Personalization

The definitions of DP and its non-personalized variants imply a worst-case privacy guarantee for every data point in a processed dataset. As a result, the lowest privacy budget of all data points has to be guaranteed for all data. Therefore, when comparing personalized with non-personalized privacy accounting, the lowest privacy budgets can be assumed to equal those from the non-personalized one. In addition, higher budgets can be assumed for some points. For the sake of brevity and due to the fact that all personalized PATE variants provide privacy guarantees for groups of sensitive data points, the private dataset was divided into two groups each with its own privacy budget. One group always had an  $(\epsilon, \delta)$ -DP budget of  $\epsilon = 1$  while the other had  $\epsilon \in \{1.5, 2, 3, 5\}$  that is the lower budget plus 50%, 100%, 200%, or 400%,

respectively. The groups were distributed s.t. their relative sizes were (25% | 75%), (50% | 50%), or (75% | 25%).

The goal of personalization is to increase utility by using more information from sensitive data. So, data points with higher privacy budgets can provide more information while discharging those with lower budgets. An important problem is, how to adjust the personalized PATE variants so that both, the higher budgets as well as the lower budgets are exhausted at about the same time. Otherwise, the privacy budget of one group of data points is wasted and the maximally possible number of labels is not produced. For the loose bound and the same alphas for both groups, this problem can easily be solved by relating the squares of the individual sensitivities according to the desired ratio. For the tight bound and different alphas, the problem can not be solved analytically since it is data dependent and the best alphas for both groups decrease at different rates. The lower the alpha value in the RDP bound the lower its epsilon value for each Gaussian mechanism (Definition 4.4) in all votings, but the higher the constant costs when RDP costs are transformed to  $(\epsilon, \delta)$ -DP costs. As a result, the best alpha, in the sense that the DP guarantee is the smallest, decreases with an increasing number of votings. The decreasing alphas are illustrated in Figure 6.3.

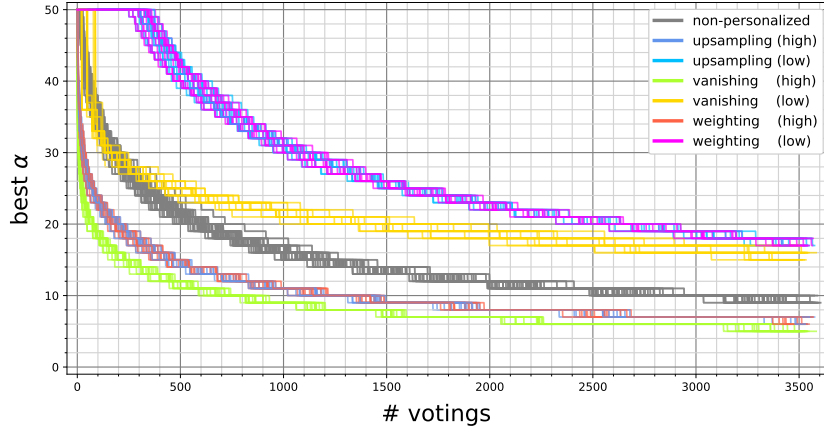


Figure 6.3: **Best Alpha.** The history of best alpha values over about 3 500 votings for each PATE variant is illustrated. After each voting, for all integer orders between two and 50 the corresponding RDP bounds of all previous votings are evaluated. The order whose RDP bound results in the minimal  $(\epsilon, \delta)$ -DP cost is considered as the best alpha. For the personalized variants, the budgets (1 | 9) with a distribution of (50% | 50%) were used. 15 voting histories for each personalized variant while 50 for the non-personalized variant are shown.

## 6 Evaluation

In order to solve the problem of finding adequate ratios for the above budget groups, experiments where the parameters were set so that the loose bounds met the ratio to test were conducted. This means that individual sensitivities should equal the square roots of the ratios. As stated in Theorem 5.1, the individual sensitivity of the uGNMax regarding any data point equals its number of duplications. Therefore, to test a wide spectrum of ratios and to use as few duplications in the uGNMax as possible, the squares of integers up to seven were tried for the higher budget. Hence, the budget combinations were  $(1 \mid 4)$ ,  $(1 \mid 9)$ ,  $(1 \mid 16)$ ,  $(1 \mid 25)$ ,  $(1 \mid 36)$ , and  $(1 \mid 49)$ . This is due to the fact, that the number of duplications of data points corresponds to the sensitivity which in turn should equal the square root of the budget. Figure 6.4

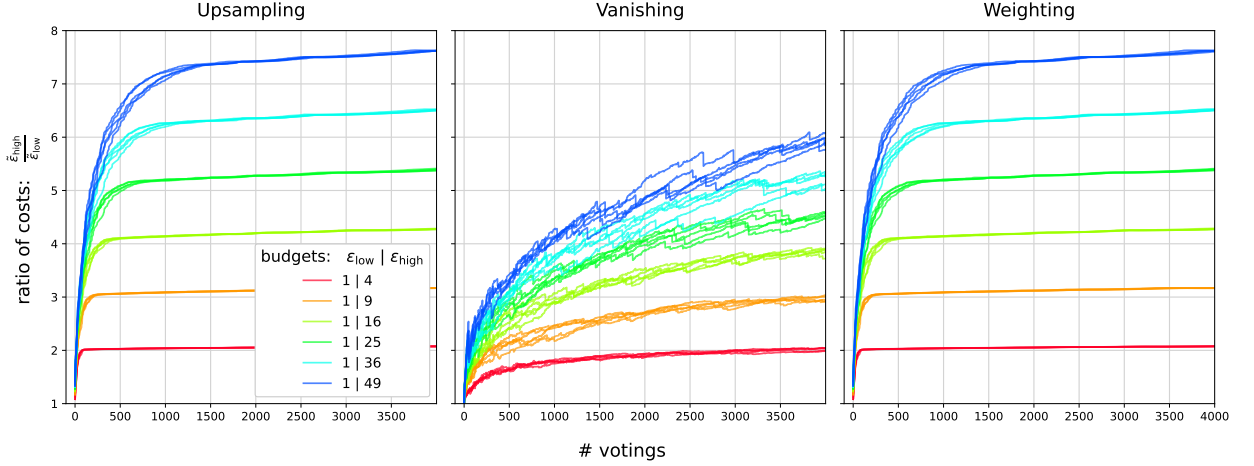


Figure 6.4: **Adjustment of Ratios.** The ratio between privacy costs over 9 000 votings is shown for several combinations of personalizations and for all GNMax variants. The costs correspond to two groups of sensitive data, one with a higher privacy budget, and one with a lower budget. Six different higher budgets were tried and are shown. The ratio of the higher budget among all sensitive data points was always 50%. One ensemble of teachers was used to vote five times for all labels in the public dataset that was shuffled differently for each combination.

visualizes the experiments. Therein, the ratio between privacy costs for data points with the higher budget and the lower budget are illustrated after each voting on the 9 000 public data points for each budget combination. After some votings, the ratio almost stays constantly at the square root of the budgets' ratio. It can be observed that the ratios for upsampling and weighting behave almost equal while those for vanishing behave differently in that they increase slower. For the following experiments, the sensitivities for upsampling and weighting were set so that their ratios were equal to the square of the intended budgets' ratio. For vanishing, the sensitivities were set so that their ratio equaled the double of the square of the intended budgets' ratio, instead. Note that only the intended budgets are considered



in the following, instead of the used ratios.

The effect of privacy personalization is best visualized by the development of privacy costs over the number of labels that are created by PATE votings. Figure 6.5 visualizes this relationship for all budget and distribution combinations and for all personalized variants as well as the non-personalized GNMax. It can be observed that in contrast to the non-personalized GNMax, the personalized variants have two privacy costs at the same time. Each cost correspond to one group of sensitive data points sharing one privacy budget. The privacy costs of both groups spread the more labels are generated. Another observation is that both, the upsampling, and the weighting variant have equivalent privacy losses on all combinations while the vanishing variant always has higher privacy costs for both privacy groups. Furthermore, the higher the difference between both budgets the higher the spread. Moreover, a higher percentage of data points belonging to the group with a higher budget leads to lower costs for both groups.

By counting the labels that were not rejected at the last voting before any of both privacy budgets exceeded, the personalization advantage can be quantified. Table 6.4 and Table 6.5 illustrate these quantifications differentiated by the four used higher budget values  $\{1.5, 2, 3, 5\}$ , the relative ratios  $\{25\%, 50\%, 75\%\}$  of the higher budget, and the three pGNMax variants. The personalization advantage can be defined by the increase of the number of labels relative to the number corresponding to the non-personalized GNMax. It reaches from  $\approx 22\%$  at weak parameters to more than 800% at strong parameters on the MNIST dataset in the sense of less or more total budget. Respectively, the advantage reaches from  $\approx 24\%$  to over 925% on the Adult dataset.

Both tables (Table 6.4, Table 6.5) indicate an interesting relation between utility and personalization distribution. E.g. the budgets  $(1 \mid 3)$  with the distribution  $(50\% \mid 50\%)$  and the budgets  $(1 \mid 5)$  with the distribution  $(75\% \mid 25\%)$  have the same average and total budgets. But more labels were created by the first one. This indicates that the more uniformly a total budget is distributed on the sensitive data, the more efficient is the utility-privacy-tradeoff. This conjecture is confirmed again by comparing the number of labels produced by the non-personalized GNMax at a budget that equals the average budget in pGNMax. For example the number of labels produced by the non-personalized GNMax until its  $(\epsilon, \delta)$ -DP cost reached  $\epsilon = 3$  averages at 1 766 while the uGNMax with budgets  $(1 \mid 5)$  distributed by  $(50\% \mid 50\%)$  produced about 1 683 labels. Further exemplary comparisons can be made by estimating the number of labels that were produced until the corresponding budgets were exhausted in Figure 6.5. Table 6.6 and Table 6.7 show the corresponding average test accuracies of students per personalization.

## 6 Evaluation

higher budget in $\varepsilon$	25% ratio			50% ratio			75% ratio		
	u	v	w	u	v	w	u	v	w
1.5	<b>271</b>	52	267	338	176	<b>340</b>	<b>412</b>	320	408
2	<b>335</b>	36	329	477	256	<b>481</b>	655	520	<b>655</b>
3	<b>462</b>	37	459	<b>820</b>	488	819	1 258	1 068	<b>1 264</b>
5	<b>775</b>	65	758	<b>1 683</b>	1 184	1 682	>2 000	>2 000	>2 000
baseline	<b>221</b>								

Table 6.4: **Labels per Personalization (MNIST)**. This table shows the number of labels created by each personalized GNMax for each personalization that was tried in this work for the MNIST dataset. For each personalization, a lower budget of  $\varepsilon = 1$  and a higher budget of  $\varepsilon \in \{1.5, 2, 3, 5\}$  was used. The ratio of sensitive data with the higher budget was one of  $\{25\%, 50\%, 75\%\}$ . The higher budgets are differentiated by rows while the relative ratios of higher budgets are differentiated by the columns. The three values in each cell correspond to the upsampling (u), vanishing (v), and weighting (w) GNMax. All values constitute the mean over 15 different voting processes rounded down. Rejected labels and those where either the standard (lower) budget ( $\varepsilon = 1$ ) or the higher budget was exhausted, are not counted. The non-personalized GNMax with  $\varepsilon = 1$  has its mean at 221 labels over 50 different voting processes.

higher budget in $\varepsilon$	25% ratio			50% ratio			75% ratio		
	u	v	w	u	v	w	u	v	w
1.5	239	54	<b>242</b>	296	160	<b>299</b>	359	280	<b>360</b>
2	292	39	<b>294</b>	<b>419</b>	229	418	561	456	<b>564</b>
3	404	34	<b>408</b>	<b>706</b>	414	706	<b>1 080</b>	919	1 079
5	668	58	<b>671</b>	1 426	993	<b>1 427</b>	>2 000	>2 000	>2 000
baseline	<b>195</b>								

Table 6.5: **Labels per Personalization (Adult)**. This table shows the number of labels created by each personalized GNMax for each personalization that was tried in this work for the Adult dataset, analog to Table 6.4. For each personalization, a lower budget of  $\varepsilon = 1$  and a higher budget of  $\varepsilon \in \{1.5, 2, 3, 5\}$  was used. The ratio of sensitive data with the higher budget was one of  $\{25\%, 50\%, 75\%\}$ . The higher budgets are differentiated by rows while the relative ratios of higher budgets are differentiated by the columns. The three values in each cell correspond to the upsampling (u), vanishing (v), and weighting (w) GNMax. All values constitute the mean over 50 different voting processes rounded down. Rejected labels and those where either the standard (lower) budget ( $\varepsilon = 1$ ) or the higher budget was exhausted, are not counted. The non-personalized GNMax with  $\varepsilon = 1$  has its mean at 195 labels over 50 different voting processes.

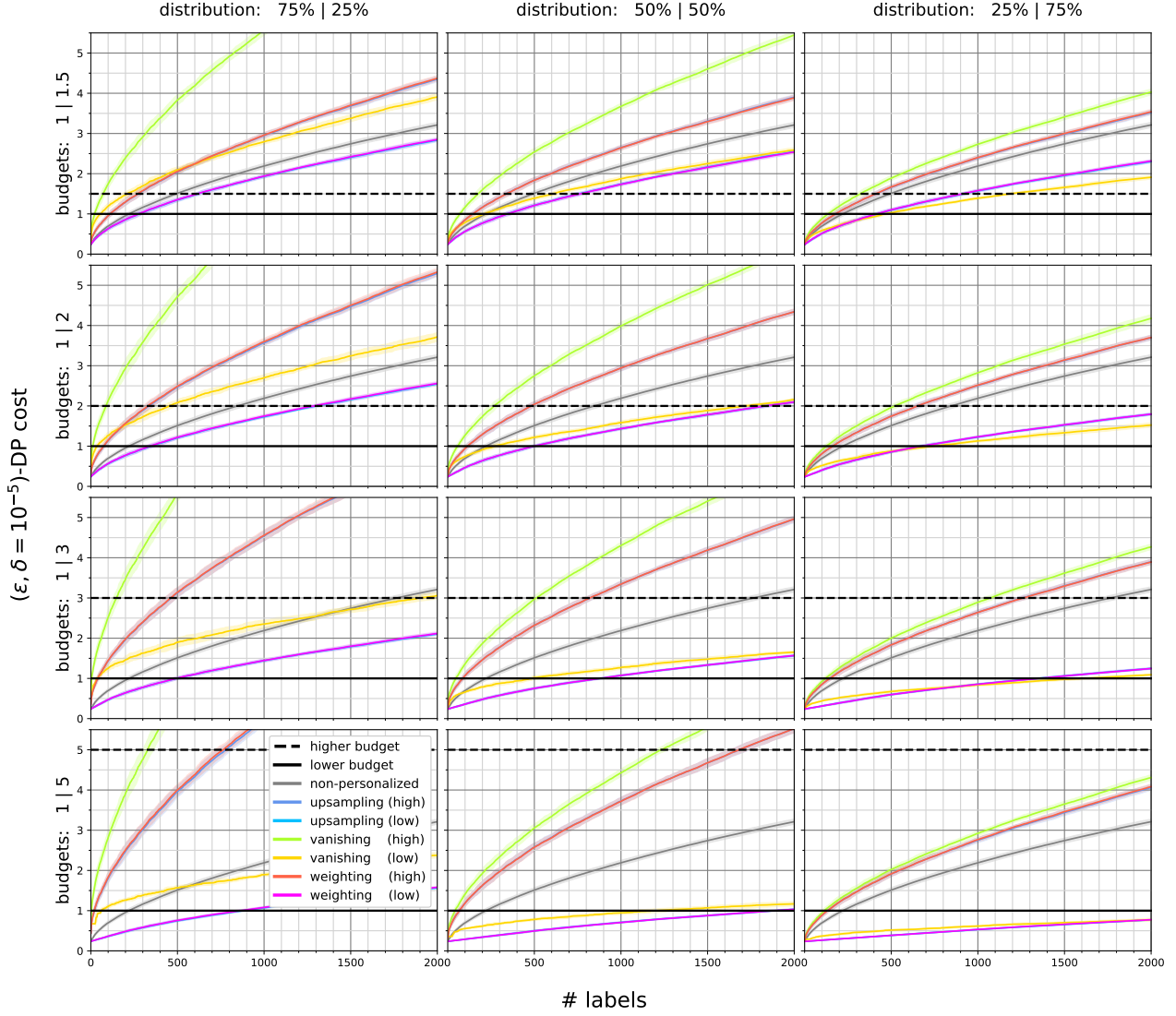


Figure 6.5: **Privacy Loss Comparison.** The history of privacy costs over the first 2000 created labels are shown for twelve different budget and distribution combinations for the GNMax and its personalized variants. The distribution is described by the relative ratio of sensitive data with the lower budget left and the relative ratio of sensitive data with the higher budget right. Each line corresponds to the average of accumulated privacy costs in  $(\epsilon, \delta)$ -DP with  $\delta = 10^{-1}$  after subsequent votings that each creates a label. The average value is taken from all experiments for one specific combination of personalization and GNMax variant and is surrounded by a confidence interval depending on the standard deviation. For each combination of budgets, distribution, and variant, 15 different voting processes are evaluated (50 for the non-personalized GNMax).

## 6 Evaluation

higher budget in $\varepsilon$	25% ratio			50% ratio			75% ratio		
	u	v	w	u	v	w	u	v	w
1.5	90.64	73.51	<b>90.92</b>	<b>91.41</b>	88.67	91.35	92.75	92.45	<b>92.91</b>
2	91.61	67.09	<b>91.77</b>	<b>93.01</b>	90.83	92.83	<b>93.4</b>	92.47	92.81
3	92.39	69.13	<b>92.41</b>	93.75	92.49	<b>93.81</b>	94.47	94.33	<b>94.52</b>
5	<b>93.6</b>	79.79	93.18	94.78	94.48	<b>94.99</b>	-	-	-
baseline	<b>90.15</b>								

Table 6.6: **Student Accuracy per Personalization (MNIST)**. This table shows the average student accuracy in % on the test set for each personalized GNMax and for each personalization that was tried in this work for the MNIST dataset. For each personalization, a lower budget of  $\varepsilon = 1$  and a higher budget of  $\varepsilon \in \{1.5, 2, 3, 5\}$  was used. The ratio of sensitive data with the higher budget was one of  $\{25\%, 50\%, 75\%\}$ . The higher budgets are differentiated by rows while the relative ratios of higher budgets are differentiated by the columns. The three values in each cell correspond to the upsampling (u), vanishing (v), and weighting (w) GNMax. All values constitute the rounded mean over 15 different voting processes. The non-personalized GNMax with  $\varepsilon = 1$  has its mean at 90.15% test accuracy over 50 different voting processes. The last three values are missing since the budgets were not exhausted at 2000 labels.

higher budget in $\varepsilon$	25% ratio			50% ratio			75% ratio		
	u	v	w	u	v	w	u	v	w
1.5	80.37	78.98	<b>81.45</b>	81.57	80.91	<b>81.61</b>	81.91	81.63	<b>81.91</b>
2	<b>81.69</b>	77.98	81.61	81.85	81.41	<b>81.85</b>	81.84	81.94	<b>81.99</b>
3	81.85	78.01	<b>82.15</b>	82.0	81.94	<b>82.01</b>	82.09	<b>82.23</b>	82.16
5	<b>82.11</b>	78.73	82.01	<b>82.24</b>	82.01	82.17	-	-	-
baseline	<b>80.63</b>								

Table 6.7: **Student Accuracy per Personalization (Adult)**. This table shows the average student accuracy in % on the test set for each personalized GNMax and for each personalization that was tried in this work for the MNIST dataset. For each personalization, a lower budget of  $\varepsilon = 1$  and a higher budget of  $\varepsilon \in \{1.5, 2, 3, 5\}$  was used. The ratio of sensitive data with the higher budget was one of  $\{25\%, 50\%, 75\%\}$ . The higher budgets are differentiated by rows while the relative ratios of higher budgets are differentiated by the columns. The three values in each cell correspond to the upsampling (u), vanishing (v), and weighting (w) GNMax. All values constitute the rounded mean over 50 different voting processes. The non-personalized GNMax with  $\varepsilon = 1$  has its mean at 90.15% test accuracy over 50 different voting processes. The last three values are missing since the budgets were not exhausted at 2000 labels.

## 7 Discussion

In this chapter, the main results of the work at hand as well as future perspectives are discussed.

### 7.1 Assessment of the Results

PATE is a suitable alternative to PGD (see Algorithm 1) and the moments accountant (see Definition 3.15) for PPML in some cases. Although PATE is simpler, it requires a larger amount of sensitive data and a public unlabeled dataset. Note that public labeled data can be used to train the student model without producing new labels by the teacher ensemble. The proposed techniques in this work improve the PATE approach. On the tested datasets, privacy costs of  $\varepsilon \leq 1$  can be achieved for all personalizations due to data augmentation while producing enough labels for practical accuracies. Moreover, the usage of higher personalized privacy budgets due to the proposed personalized PATE variants increases the number of produced labels even further. Therefore, privacy guarantees below the recommended value of  $\varepsilon = 1$  are feasible in practice.

The utility of ML models that stems from the learning of sensitive data does not depend on the amount and quality of the data, exclusively. Models and trainings can be optimized in order to increase the accuracy as well. Moreover, there are different semi-supervised learning techniques to make use of unlabeled data, like *virtual adversarial training* (VAT) [27]. The authors of [31] applied VAT to the training of students after producing labels. Accordingly, they achieved a student accuracy of 98.5% using 286 labels that are only 93.18% accurate. Since no VAT was used for this work and the student's optimization is inferior to that in [31], the students only achieve accuracies of about 95%, although more labels of higher accuracy are used. However, the benefits of the proposed techniques are unquestioned.

## 7.2 Comparison of the Personalized Variants

The three personalization techniques for PATE proposed in this work were only evaluated for the improved PATE (see Section 4.2) based on the Gaussian mechanism. Nevertheless, similar properties can be assumed when a different kind of noise is used.

While uGNMax and wGNMax perform equivalently in terms of privacy and utility, the vGNMax aggregator performs always worse and on some personalizations even worse than the non-personalized GNMax. It should therefore not be used except as a supplement to the others. On the one hand, uGNMax requires more computing resources, compared to wGNMax. On the other hand, it can provide privacy personalization to very small groups of data points that share a unique privacy preference and even single points. Actually both, uGNMax and wGNMax approach discretization problems that come with the personalization of PATE differently.

It is possible to combine the three personalized PATE variants. The concrete combination should depend on the data and their distribution of privacy budgets. If data that share the same privacy level suffice to train one or more teachers, it is to be preferred to train an accurate number of teachers on them and to weight them afterwards in votings according to the privacy level. Instead, if that group of data does not provide enough information to train a teacher, these data should be upsampled and given to several teachers together with other points. Contrary, the vanishing technique should only be used additionally to the other variants when any privacy budget is already exhausted. Then, the corresponding teachers do not participate in further votings but the other teachers can continue to spend their information to produce more labels. The upsampling and weighting techniques would stop to produce labels in that case and thus waste some privacy budget. However, a different choice of personalization parameters (weights or duplications) in advance would have lead to better results than with vanishing.

## 7.3 Future Work

DP, RDP, and PDP are worst-case guarantees for the privacy expenditure of data-processing mechanisms with regard to that data. Since worst-case guarantees are required in the field of IT security, DP and its variants are popular in research. Unfortunately, the measurement of DP costs is more or less imprecise for different ML mechanisms. For example, the privacy accounting in PATE is based on the assumption that one data point completely determines the behavior of the teacher that was trained on it. This assumption is necessary due to the worst-case characteristic of DP.

In practice, each data point has much less influence on an ML model. This can solely be justified by the argument that there is a high number of different data points that the model learns. Thus, all training data share control over the model probably in almost equal parts. Therefore, the actual influence of sensitive data on the student model in the PATE algorithm is far lower than the DP guarantee suggests. Different mechanisms might have different gaps between DP guarantees and actual privacy losses s.t. they might not be comparable with each other.

One point of criticism on the privacy measurement of PATE is as follows. The scale of the induced noise is carefully chosen so that the noise changes a voted label very rarely. Consider a noise scale that leads to one changed label within 1 000 votings. Then consider a noise with an intensity s.t. one out of 10 000 labels is changed. The weaker second noise probably leads to much higher measured privacy costs although the votings are not affected significantly by any of both noises at all. This thought experiment hints on the precision problem again.

Besides the above described precision problem of DP, the practical meaning of epsilon values in DP is not understood well. The theoretical meaning is clear: For each possible result of a mechanism, the probabilities of the mechanism to hit that result once with and once without a particular data point included in the training set must not exceed  $e^\epsilon$  times each other plus  $\delta$ . Other DP variants are similarly defined. Ultimately, the amount of information that could be obtained by attackers is of users' interest. But DP can not be transformed into attacker advantages in general. This is due to two facts. At first, the precision problem of DP and the incomparability of different DP mechanisms inhibits such a transformation. Secondly, there could be invented stronger attacks in the future that are not anticipated. This is why IT security mostly relies on theoretical worst-case guarantees. The solving of these problems will be challenging and could highly benefit the whole privacy research field.

In this work, all data points were sampled randomly. Therefore, correlations between specific properties of data points and their corresponding privacy preferences were not examined. Nevertheless, it can be assumed that there are such correlations in many cases. Research on this matter would be beneficial, no matter if for personalized PATE or PDP research in general.





## 8 Conclusion

The goal of this thesis was to provide possibilities to enable donations of sensitive data for PPML in a personalized way. Therefore, a variety of privacy options should be provided in order for the data donors to decide how much privacy they require. The current state of research on PDP was described in Chapter 3 and a gap regarding ML applicability was pointed out. Since donors of sensitive data usually are laypersons, personalized PPML algorithms and their privacy protection should be easy to understand. Unfortunately, the only two known PDP mechanisms for ML are not. In contrast, the prominent DP mechanism PATE provides an intuitive understanding of its functionality as well as its privacy preservation since the sensitive data are never given to the ML model to publish (see Chapter 4). PATE transfers the knowledge of teacher models learned from partitions of sensitive data to a student model via produced labels for a public dataset.

To achieve this thesis' goal, three personalized extensions of PATE were developed (see Chapter 5). Two of them reduce the influence of particular teachers on produced labels. This is done by weighting teachers differently (see Section 5.4) or by avoiding the participation of some teachers at the production of some labels (see Section 5.3). The other PATE variant increases the privacy expenditure of particular data points by upsampling them so that one duplicate is given to several teachers for each upsampled data point (see Section 5.2). Simultaneously, the privacy costs of all data points are decreased due to a higher number of teachers to be trained and a stronger noise that is applied in the label production.

The advantage of the proposed techniques over the non-personalized PATE were shown in Chapter 6. Depending on the personalized privacy requirements and the underlying dataset, the proposed PATE extensions create 22% – 925% more labels (see Section 6.2.2). Moreover, it was pointed out that PATE and its proposed extensions are robust against data augmentation on teacher-level. By making use of this finding, significant privacy reductions were achieved due to more accurate teacher models (see Section 6.2.1). Another main contribution of this work is the observation that given a total privacy budget, the maximal efficiency of generating utility is achieved by a uniform distribution of budgets over all sensitive data. This would actually not require any personalization at all. Nevertheless, since data donors are independent of each other and have individual privacy needs, personalized privacy expenditure optimizes utility in most cases.



# Bibliography

- [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*. 2016, pp. 308–318.
- [2] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. “Heterogeneous differential privacy”. In: *arXiv preprint arXiv:1504.06998* (2015).
- [3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.
- [4] Jason Brownlee. *How to Develop a CNN for MNIST Handwritten Digit Classification*. 2019. URL: <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/> (visited on 05/20/2021).
- [5] Lei Cui, Youyang Qu, Mohammad Reza Nosouhi, Shui Yu, Jian-Wei Niu, and Gang Xie. “Improving data utility through game theory in personalized differential privacy”. In: *Journal of Computer Science and Technology* 34.2 (2019), pp. 272–286.
- [6] Rachel Cummings and David Durfee. “Individual sensitivity preprocessing for data privacy”. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 528–547.
- [7] TensorFlow Developers. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <http://tensorflow.org/>.
- [8] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- [9] Cynthia Dwork. “Differential privacy: A survey of results”. In: *International conference on theory and applications of models of computation*. Springer. 2008, pp. 1–19.
- [10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

## Bibliography

- [11] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy.” In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.
- [12] Hamid Ebadi, David Sands, and Gerardo Schneider. “Differential privacy: Now it’s getting personal”. In: *Acm Sigplan Notices* 50.1 (2015), pp. 69–81.
- [13] Vitaly Feldman and Tijana Zrnic. “Individual Privacy Accounting via a Renyi Filter”. In: *arXiv preprint arXiv:2008.11193* (2020).
- [14] 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*. 2015, pp. 1322–1333.
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 0262035618.
- [16] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [17] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [18] 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*. 2019, pp. 4323–4332.
- [19] Zach Jorgensen, Ting Yu, and Graham Cormode. “Conservative or liberal? Personalized differential privacy”. In: *2015 IEEE 31st international conference on data engineering*. IEEE. 2015, pp. 1023–1034.
- [20] Jong Wook Kim, Kennedy Edemacu, and Beakcheol Jang. “MPPDS: Multilevel Privacy-Preserving Data Sharing in a Collaborative eHealth System”. In: *IEEE Access* 7 (2019), pp. 109910–109923.
- [21] Nitin Kohli and Paul Laskowski. “Epsilon voting: Mechanism design for parameter selection in differential privacy”. In: *2018 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE. 2018, pp. 19–30.
- [22] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

- [23] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: <http://yann.lecun.com/exdb/mnist/>.
- [24] 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*. Springer. 2017, pp. 615–627.
- [25] Frank McSherry and Kunal Talwar. “Mechanism design via differential privacy”. In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*. IEEE. 2007, pp. 94–103.
- [26] Ilya Mironov. “Rényi differential privacy”. In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.
- [27] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8 (2019), pp. 1979–1993. DOI: [10.1109/TPAMI.2018.2858821](https://doi.org/10.1109/TPAMI.2018.2858821).
- [28] 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)*. IEEE. 2020, pp. 1–6.
- [29] Nicolas Papernot. *Implementation of an RDP privacy accountant and smooth sensitivity analysis for the PATE framework*. 2018. URL: [https://github.com/tensorflow/privacy/tree/master/research/pate\\_2018](https://github.com/tensorflow/privacy/tree/master/research/pate_2018) (visited on 05/20/2021).
- [30] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. “Semi-supervised knowledge transfer for deep learning from private training data”. In: *arXiv preprint arXiv:1610.05755* (2016).
- [31] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. “Scalable private learning with pate”. In: *arXiv preprint arXiv:1802.08908* (2018).
- [32] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.
- [33] Fernando Pérez and Brian E. Granger. “IPython: a System for Interactive Scientific Computing”. In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 21–29. ISSN: 1521-9615. DOI: [10.1109/MCSE.2007.53](https://doi.org/10.1109/MCSE.2007.53). URL: <https://ipython.org>.
- [34] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. “Stochastic gradient descent with differentially private updates”. In: *2013 IEEE Global Conference on Signal and Information Processing*. 2013, pp. 245–248. DOI: [10.1109/GlobalSIP.2013.6736861](https://doi.org/10.1109/GlobalSIP.2013.6736861).

## Bibliography

- [35] Stanley Smith Stevens et al. “On the theory of scales of measurement”. In: (1946).
- [36] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134). URL: <https://doi.org/10.5281/zenodo.3509134>.
- [37] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.
- [38] Isabel Wagner and David Eckhoff. “Technical privacy metrics: a systematic survey”. In: *ACM Computing Surveys (CSUR)* 51.3 (2018), pp. 1–38.
- [39] Shaowei Wang, Liusheng Huang, Miaomiao Tian, Wei Yang, Hongli Xu, and Hansong Guo. “Personalized privacy-preserving data aggregation for histogram estimation”. In: *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2015, pp. 1–6.
- [40] Yu-Xiang Wang. “Per-instance Differential Privacy”. In: *Journal of Privacy and Confidentiality* 9.1 (2019).
- [41] Michael L. Waskom. “seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021). URL: <https://doi.org/10.21105/joss.03021>.
- [42] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. “Local differential privacy and its applications: A comprehensive survey”. In: *arXiv preprint arXiv:2008.03686* (2020).
- [43] Shun Zhang, Laixiang Liu, Zhili Chen, and Hong Zhong. “Probabilistic matrix factorization with personalized differential privacy”. In: *Knowledge-Based Systems* 183 (2019), p. 104864.
- [44] Benjamin Zi Hao Zhao, Mohamed Ali Kaafar, and Nicolas Kourtellis. “Not one but many Tradeoffs: Privacy Vs. Utility in Differentially Private Machine Learning”. In: *arXiv preprint arXiv:2008.08807* (2020).